



UNIVERSIDAD NACIONAL DE COLOMBIA

Diseño Generativo Realimentado (DGR) como soporte en los procesos de creación de productos

Jhoan Sebastian Restrepo Mendoza

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Bogotá D.C., Colombia
Año 2023

Diseño Generativo Realimentado (DGR) como soporte en los procesos de creación de productos

Jhoan Sebastian Restrepo Mendoza

Tesis presentada como requisito parcial para optar al título de:
Magister en Automatización Industrial

Director(a):
Profesor Ernesto Córdoba Nieto

Línea de Investigación:
Diseño de productos y procesos industriales y preseries
Grupo de Investigación:
Grupo de trabajo en nuevas tecnologías de diseño y manufactura-automatización DIMA-UN

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Bogotá D.C., Colombia
Año 2023

Dedicatoria

Al apoyo constante de mi familia y seres queridos, que han propiciado el desarrollo exitoso de este trabajo.

Resumen

Diseño Generativo Realimentado (DGR) como soporte en los procesos de creación de productos

El presente trabajo de maestría se presenta como una metodología de diseño generativo realimentado para la creación de producto. Esta metodología se plantea considerando la integración de tecnología actual como lo es el diseño paramétrico en el software CAD y la propuesta de un ecosistema discreto de nube de puntos. Se usa conceptos de técnicas de optimización multiobjetivo para la exploración y explotación de un espacio, al generar un conjunto de soluciones que satisfacen los objetivos del nuevo producto.

Se propone un desarrollo de modelamiento basado en estructuras de datos, específicamente grafos direccionales. Los grafos direccionales contienen en sus nodos la información necesaria para operar sus entradas y generar valores de salida, que a su vez serán usados por otro nodo para operar y obtener otras salidas. Este proceso secuencial permite obtener el modelamiento de un componente, que será tratado posteriormente en procesos de optimización multiobjetivo para obtener soluciones (nuevos productos) al clasificar los frentes de Pareto.

La realimentación de esta propuesta se genera al desarrollar simulaciones de las etapas de manufactura, por tal motivo el ecosistema propuesto y la tecnología actual permite ingresar como objetivos estas etapas posteriores y reducir las iteraciones para obtener un producto funcional. Para la creación del ecosistema propuesto se implementa el uso de programación paralela en la obtención de soluciones que por métodos secuenciales no son viables.

Palabras clave: Diseño generativo realimentado, Optimización multiobjetivo, diseño paramétrico, grafos direccionales, frentes de Pareto, programación paralela.

Abstract

Feedback Generative Design (FGD) as Support in Product Creation Processes

The present master's thesis is presented as a feedback-based generative design methodology for product creation. This methodology is proposed considering the integration of current technology such as parametric design in CAD software and the proposal of a discrete point

cloud ecosystem. Concepts of multi-objective optimization techniques are used for the exploration and exploitation of a space, generating a set of solutions that satisfy the objectives of the new product.

A modeling development based on data structures is proposed, specifically directed graphs. Directed graphs contain the necessary information in their nodes to operate their inputs and generate output values, which will in turn be used by another node to operate and obtain further outputs. This sequential process allows obtaining the modeling of a component, which will be subsequently subjected to multi-objective optimization processes to obtain solutions (new products) by classifying Pareto fronts.

Feedback in this proposal is generated by simulating the manufacturing stages. For this reason, the proposed ecosystem and current technology allow incorporating these subsequent stages as objectives and reducing iterations to obtain a functional product. The creation of the proposed ecosystem involves the implementation of parallel programming to obtain solutions that are not viable through sequential methods.

Keywords: feedback-based generative design, multi-objective optimization, parametric design, directed graphs, Pareto fronts, parallel programming.

Lista de Figuras

| | |
|---|----|
| 2-1. Análisis - terminología sobre “Diseño Generativo”. Fuente: Creación Propia - VOSviewer | 8 |
| 2-2. Ejemplo de diseño paramétrico. Fuente: Congreso CIMMX [1] | 12 |
| 2-3. Modelo de arquitectura de referencia para la Industria 4.0 (RAMI4.0). Fuente: Martin Hankel, Digitising Manufacturing 2016 [2] | 21 |
| 2-4. Concepto lineal de ciclo de vida del producto. Fuente: Creación Propia | 25 |
| 2-5. Concepto concurrente aplicado en el LabFabEx-UN. Fuente: Creación Propia. | 25 |
| 2-6. Descripción del modelo RAMI4.0 en el tiempo. Fuente: OPC Day Europe 2016 [3] | 26 |
| 2-7. Esquema de la influencia del diseño generativo en el ciclo de vida del producto. Fuente: Creación Propia | 32 |
| 3-1. Interacción de las clases basado en el modelo. Fuente: Creación Propia | 36 |
| 3-2. Definición de MUCO basado en un arreglo de arboles. Fuente: Creación Propia | 37 |
| 3-3. Esquema de la estructura de un grafo direccional basado en el modelo. Fuente: Creación Propia | 38 |
| 3-4. Interacción de las clases basado en primitivas. Fuente: Creación Propia | 39 |
| 3-5. Definición de MUCO basado en primitivas. Fuente: Creación Propia | 40 |
| 3-6. Esquema de la estructura de un grafo direccional basado en primitivas. Fuente: Creación Propia | 41 |
| 3-7. Definición de sistema como la interacción de componentes. Fuente: Creación Propia | 42 |
| 3-8. Representación de conceptos fabricados en base al modelo y a primitivas. Fuente: Creación Propia | 44 |
| 3-9. Representación de conceptos fabricados en las estructuras de datos. Fuente: Creación Propia | 45 |

| | |
|--|----|
| 3-10. MUCO de un eslabón basado en el modelo. | |
| Fuente: Creación Propia | 46 |
| 3-11. Ampliación de un MUCO en la estructura de datos basado en primitivas. | |
| Fuente: Creación Propia | 47 |
| 3-12. Operaciones en el grafo direccional por primitivas. | |
| Fuente: Creación Propia | 54 |
| 4-1. Salidas de la etapa de diseño en el enfoque de diseño generativo. | |
| Fuente: Creación Propia | 57 |
| 4-2. Planteamiento de un eslabón bajo el diseño paramétrico. | |
| Fuente: Creación Propia | 58 |
| 4-3. Cadena cinemática generada por la interacción del concepto de eslabón. | |
| Fuente: Creación Propia | 59 |
| 4-4. Flujo planteado para la representación de un componente como una nube de puntos. | |
| Fuente: Creación Propia | 60 |
| 4-5. Representación del grafo inicial en la nube de puntos. | |
| Fuente: Creación Propia | 61 |
| 4-6. Representación generativa por nube de puntos. | |
| Fuente: Creación Propia | 62 |
| 4-7. Salidas de la etapa de manufactura en base al diseño generativo. | |
| Fuente: Creación Propia | 64 |
| 4-8. Despliegue de servidor de visión en el LabFabEx-UN. | |
| Fuente: Creación Propia | 66 |
| 4-9. Esquema de integración de servidores de visión en PRIA-UN. | |
| Fuente: Creación Propia | 67 |
| 4-10. Ejemplo aplicado del modelo de vision de maquina. | |
| Fuente: Congreso CIMMX [1] | 68 |
| 4-11. Puntos de ubicación de cámara y etiqueta de resultados. | |
| Fuente: Gonzalez [4] y Creación Propia | 69 |
| 4-12. Planteamiento de estaciones para despliegue y ajuste en fabrica. | |
| Fuente: Creación propia | 70 |
| 4-13. Prueba de ubicación de la estación en diferentes espacios de trabajo del LabFabEx-UN. | |
| Fuente: Creación propia | 71 |
| 4-14. Proceso de identificación de contornos en imágenes sintéticas y fotografías de componentes. | |
| Fuente: Creación propia | 72 |
| 4-15. Definición de una superficie de salida de parámetros de software CAM. | |
| Fuente: Creación propia | 74 |

| | |
|--|-----|
| 4-16.Arquitectura de uso de generación de código G para remoción de material. Fuente: Creación propia | 76 |
| 4-17.Funcionalidad presente en los actores del sistema de generación de código G. Fuente: Creación propia | 77 |
| 4-18.Modelo de clases para el sistema de generación de código G. Fuente: Creación propia | 78 |
| 4-19.Visualización colisión en nube de puntos de fresado en Y. Fuente: Creación propia | 80 |
| 4-20.Visualización colisión en nube de puntos de fresado en Z. Fuente: Creación propia | 80 |
| 4-21.Visualización colisión en nube de puntos de tronzado en X. Fuente: Creación propia | 81 |
| 4-22.Vista superior ejercicios prácticos. Fuente: Creación propia | 81 |
| 4-23.Vista lateral ejercicios prácticos. Fuente: Creación propia | 82 |
| 4-24.Ejercicios prácticos de altura mayor al diámetro de herramienta. Fuente: Creación propia | 83 |
| 5-1. Estructura planteada de diseño generativo para resolver el diseño del Gripper. Fuente: Creación propia | 86 |
| 5-2. Definición del objeto a manipular. Fuente: Creación propia | 88 |
| 5-3. Representación de distintos grafos para un eslabón. Fuente: Creación propia | 89 |
| 5-4. Representación paramétrica de un grafo en el modelo CAD. Fuente: Creación propia | 89 |
| 5-5. Grafos direccionales para el componente de contacto general. Fuente: Creación propia | 90 |
| 5-6. Interfaz rígida de componente de contacto general. Fuente: Creación propia | 91 |
| 5-7. Definición espacio de trabajo del objeto a manipular. Fuente: Creación propia | 100 |
| 5-8. Muestra de componente modulares para desarrollo de estructura y sujeción de actuadores. Fuente: Creación propia | 106 |
| 5-9. Generación molde en CAD para grafo de contacto general. Fuente: Creación propia | 107 |
| 5-10.Vista lateral y frontal de la cadena cinemática impresa. Fuente: Creación propia | 108 |

| | | |
|--------------|--|-----|
| 5-11. | Vista frontal y lateral de la estructura del Gripper. | |
| | Fuente: Creación propia | 109 |
| 5-12. | Manufactura de los moldes y elementos de contacto general. | |
| | Fuente: Creación propia | 109 |
| 5-13. | Arquitectura del Gripper para integración con software MES. | |
| | Fuente: Creación propia | 110 |
| 5-14. | Secuencia del Gripper realizando la operación de Pick and Place. | |
| | Fuente: Creación propia | 111 |
| 5-15. | Descripción de la secuencia de ejecución en NodeRED. | |
| | Fuente: Creación propia | 112 |
| 5-16. | Flujo de operación del Pick and Place. | |
| | Fuente: Creación propia | 113 |
| 5-17. | Visualización del proceso de Pick and Place bajo la configuración de servidor. | |
| | Fuente: Creación propia | 113 |
| 5-18. | Secuencia de sujeción del objeto Circular y rectangular. | |
| | Fuente: Creación propia | 114 |
| B-1. | Tabla de MUCOs adyacentes. | |
| | Fuente: Creación Propia | 125 |
| B-2. | Alojar memoria dinámica en la GPU. | |
| | Fuente: Creación Propia | 126 |
| B-3. | Posibilidad 1. | |
| | Fuente: Creación Propia | 127 |
| B-4. | Posibilidad 2. | |
| | Fuente: Creación Propia | 127 |
| B-5. | Posibilidad 3. | |
| | Fuente: Creación Propia | 128 |
| E-1. | Esquema de aplicación en el LabFabEx-UN. | |
| | Fuente: Creación propia | 135 |
| E-2. | Parámetros de configuración del algoritmo generativo secuencial. | |
| | Fuente: Creación propia | 136 |
| E-3. | Ejemplificación de las soluciones generadas por el algoritmo. | |
| | Fuente: Creación propia | 137 |
| E-4. | Modificación de parámetros en Fusion360. | |
| | Fuente: Creación propia | 138 |
| E-5. | Plancha de impresión 3D. | |
| | Fuente: Creación propia | 139 |
| E-6. | Eslabones impresos 3D. | |
| | Fuente: Creación propia | 139 |

| | |
|---|-----|
| E-7. Medición radios en dos probetas- Estudiante: Jhoan. Fuente: Creación propia | 140 |
| E-8. Medición radios en dos probetas- Estudiante: Fabian. Fuente: Creación propia | 140 |
| E-9. Medición radios en dos probetas- Estudiante: Camilo. Fuente: Creación propia | 141 |
| E-10 Medición radios en dos probetas- Estudiante: Camilo. Fuente: Creación propia | 141 |
| E-11 Probetas de tolerancia 0.2mm a 0.7 mm. Fuente: Creación propia | 142 |
| E-12 Ensamblaje de pines con tolerancia de 0.6 y 0.7 mm. Fuente: Creación propia | 143 |
| E-13 Ensamblaje de pines con tolerancia de 0.6 y 0.7 mm de todos los estudiantes. Fuente: Creación propia | 143 |
| E-14 Estudiantes participantes. Fuente: Creación propia | 144 |
| F-1. Gripper ensamblado en el espacio de trabajo. Fuente: Creación propia | 145 |
| F-2. Comando robot SCARA apertura del efector final. Fuente: Creación propia | 146 |
| F-3. Vista trasera sujeción del Gripper. Fuente: Creación propia | 146 |
| F-4. Vista lateral y frontal sujeción del Gripper. Fuente: Creación propia | 147 |
| F-5. Arquitectura de conexión con PRIA-UN en RPI4. Fuente: Creación propia | 148 |
| F-6. Conexión VNC. Fuente: Creación propia | 149 |
| F-7. Comenzar a ejecutar en RPI la imagen de docker. Fuente: Creación propia | 149 |
| F-8. Configurar el entorno de ROS en docker. Fuente: Creación propia | 150 |
| F-9. Ejecuta el maestro de ROS. Fuente: Creación propia | 150 |
| F-10 Ejecución Roslauch. Fuente: Creación propia | 151 |
| F-11 Conexión al sistema de PRIA-UN del robot. Fuente: Creación propia | 151 |

| | |
|---|-----|
| F-12. Verificación conectividad del Gripper con PRIA-UN. Fuente: Creación propia | 151 |
| F-13. Ejecutar tópico de ROS para transmitir imágenes. Fuente: Creación propia | 152 |
| F-14. Ejecutar el servidor de visión en la RPI4. Fuente: Creación propia | 152 |
| F-15. Visualización externa del servidor de visión. Fuente: Creación propia | 153 |
| H-1. Diagrama de caja gris del Gripper. | 161 |
| H-2. Descomposición funcional del Gripper. | 162 |
| H-3. Mapeo función 1-concepto. | 166 |
| H-4. Mapeo función 2-concepto. | 167 |
| H-5. Mapeo función 3-concepto. | 167 |
| H-6. Mapeo función 5-concepto. | 168 |
| H-7. Tabla de combinación de conceptos en la línea funcional principal. | 168 |
| H-8. Conceptos Globales resultantes. (a.) Eslabones rígidos y tensores. (b.) Pinzas por engranajes. (c.) Pinzas por correas dentadas. | 169 |
| H-9. Conceptos Globales dominante: Sistema de eslabones rígidos y actuación por tensores. | 170 |
| I-1. Interacciones fundamentales. | 172 |
| I-2. Interacciones Incidentales. | 173 |
| I-3. Interacción de los subsistemas. | 174 |
| I-4. Movimiento de eslabones por medio de tensores. a) Simulando el movimiento sin condiciones espaciales en el eje de rotación. b) Simulando condiciones de polea en el eje de rotación. | 177 |
| I-5. Condiciones de polea en el eje de rotación. | 178 |
| I-6. Condiciones de polea en la aplicación bidireccional de fuerzas tensoras. | 178 |
| I-7. Condición crítica al manipular objetos. | 179 |
| I-8. Fuerza de contacto con el objeto a manipular. a) La fuerza normal que ejerce el objeto sobre el último eslabón. b) Geometría que debe tener el último eslabón para que la fuerza que el contacto con el objeto se de en diferentes orientaciones. | 180 |
| I-9. Espacio de trabajo para una cadena cinemática. a) Tres posiciones del eslabón inferior. b) Diez posiciones del eslabón inferior. | 180 |
| I-10. Espacio de trabajo del Gripper. | 181 |
| I-11. Prototipos de cadenas cinemáticas accionada por tensores. a) Accionamiento por un tensor en tres eslabones. b) Accionamiento de por dos tensores en tres eslabones. | 182 |
| I-12. Simulación de tensores para la cadena cinemática inferior en lazo abierto. | 183 |

| | |
|--|-----|
| I-13. Momentos en los ejes de rotación de cada eslabón. a) Eslabón superior. b) Eslabón inferior. | 183 |
| I-14. Simulación de la interacción de contacto de las cadenas cinemáticas con un objeto de prueba. | 184 |
| I-15. Resultados de torques de la interacción de contacto de las cadenas cinemáticas con un objeto de prueba. a) Cadena inferior eslabón superior. b) Cadena inferior eslabón inferior. c) Cadena superior eslabón superior. d) Cadena superior eslabón inferior. | 184 |
| I-16. Simulación de motores en los ejes de rotación. | 185 |
| I-17. Resultados de torques de simulación de motores en los ejes de rotación. a) Cadena inferior eslabón superior. b) Cadena inferior eslabón inferior. c) Cadena superior eslabón superior. d) Cadena superior eslabón inferior. | 186 |
| I-18. Calculo de elementos finitos en el eslabón superior. a) Von Mises. b) Deformaciones. c) Factor de seguridad. | 186 |
| I-19. Calculo de elementos finitos en el eje de rotación. a) Von Mises. b) Deformaciones. c) Factor de seguridad. | 187 |
| I-20. Desarrollo del proceso de optimización topológica. | 187 |
| I-21. Desarrollo del proceso de optimización topológica de verificación. | 188 |
| J-1. Ensamblaje final del Gripper. | 189 |
| J-2. Conjunto de la cadena cinemática. | 190 |
| J-3. Conjunto de la estructura | 190 |
| J-4. Secciones del conjunto de la estructura. a) Electrónica, control y potencia. b) Motores lineales. | 191 |
| J-5. Interfaz de sujeción con el brazo robótico. | 191 |

Lista de Tablas

| | |
|--|-----|
| 5-1. Especificaciones CUDA tarjeta NVIDIA Geforce GTX 1050 Ti. | 87 |
| 5-2. Disponibilidad tecnológica en el LabFabEx-UN. | 94 |
| 5-3. Complejidad Tecnológica en el LabFabEx-UN. | 95 |
| 5-4. Tolerancias de los procesos de manufactura en el LabFabEx-UN. | 95 |
| 5-5. Tiempo de ciclo de fabricación en el LabFabEx-UN. | 96 |
| 5-6. Resumen de criterios de flexibilidad actual del LabFabEx-UN. | 96 |
| 5-7. Disponibilidad tecnológica con mejoras en flexibilidad en el LabFabEx-UN. | 97 |
| 5-8. Complejidad Tecnológica con mejoras en flexibilidad en el LabFabEx-UN. | 97 |
| 5-9. Tolerancias de los procesos de manufactura con mejoras en flexibilidad en el LabFabEx-UN. | 98 |
| 5-10. Resumen de criterios de flexibilidad con mejoras en flexibilidad en el LabFabEx-UN. | 98 |
| 5-11. Soluciones obtenidas en el entorno actual del LabFabEx. | 104 |
| 5-12. Soluciones obtenidas en el entorno con mejoras en flexibilidad del LabFabEx. | 105 |
| 5-13. Resultados de operación de Pick and Place con diferentes piezas. | 114 |
| | |
| E-1. Soluciones seleccionadas. | 137 |
| E-2. Diámetros efectivos para los ejes. | 142 |
| E-3. Mediciones en maquina Zoller. | 142 |
| | |
| G-1. Requerimientos del cliente. | 156 |
| G-2. Especificaciones técnicas, sus valores, tipo y valor de dificultad. | 157 |

Contenido

| | |
|---|-------------|
| Resumen | vii |
| Lista de figuras | ix |
| Lista de tablas | xvii |
| Lista de símbolos | xxii |
| 1. Introducción | 1 |
| 1.1. Justificación de la investigación | 2 |
| 1.2. Pregunta de investigación | 6 |
| 1.3. Objetivo General | 6 |
| 1.3.1. Objetivo Específicos | 6 |
| 2. Revisión de la literatura | 7 |
| 2.1. Diseño Generativo | 7 |
| 2.1.1. Experticia del diseñador | 8 |
| 2.1.2. Integración etapas de diseño | 9 |
| 2.1.3. Mejoramiento de producto | 13 |
| 2.2. Integración en esquemas de manufactura | 13 |
| 2.2.1. Software MES en la creación de productos | 15 |
| 2.2.2. Software PLM en la creación de productos | 16 |
| 2.2.3. Manufactura flexible | 18 |
| 2.3. Marco teórico | 19 |
| 2.3.1. Modelos contemporáneos de automatización industrial | 19 |
| 2.3.2. Influencia de procesos de creación de productos según su clasificación de manufactura | 20 |
| 2.3.3. Ciclo de vida del producto | 24 |
| 2.3.4. Programación secuencial y paralela | 27 |
| 2.3.5. Optimización aplicada en la creación de producto | 28 |
| 2.4. Hipótesis | 31 |
| 2.5. Metodología | 32 |

| | |
|--|-----------|
| 3. Modelación en el Diseño Generativo Realimentado | 34 |
| 3.1. Estructura de datos | 35 |
| 3.1.1. Basado en modelamiento matemático | 36 |
| 3.1.2. Basado en primitivas | 39 |
| 3.1.3. Desarrollo de las estructuras de datos | 42 |
| 3.2. Componentes rígidos y flexibles | 44 |
| 3.2.1. Componentes rígidos | 44 |
| 3.2.2. Componentes flexibles | 46 |
| 3.3. Planteamiento de objetivos | 47 |
| 3.3.1. Objetivos espaciales | 48 |
| 3.3.2. Objetivos de fuerzas | 49 |
| 3.3.3. Objetivos en diferentes etapas del ciclo de vida del producto | 51 |
| 3.4. Operaciones de mutación | 53 |
| 3.5. Conclusiones de las estructuras de datos | 54 |
| 4. Aplicación del Diseño Generativo Realimentado | 56 |
| 4.1. Concepciones del diseño generativo en el diseño | 57 |
| 4.1.1. Basado en diseño paramétrico | 58 |
| 4.1.2. Basado en la representación por nube de puntos | 60 |
| 4.2. Concepciones del diseño generativo en la manufactura y almacenamiento | 63 |
| 4.3. Concepciones del diseño generativo en el uso | 65 |
| 4.4. Generación de objetos inteligentes por visión de maquina | 66 |
| 4.4.1. Obtención Imágenes CAD | 69 |
| 4.4.2. Obtención registro fotográfico | 69 |
| 4.4.3. Identificación de características en componentes | 71 |
| 4.5. Transmisión de información a etapa de fabricación | 73 |
| 4.5.1. En modelo de software comercial | 74 |
| 4.5.2. En programación de macros | 76 |
| 4.5.3. En modelo de nubes de puntos | 76 |
| 5. Caso de estudio desarrollo del Gripper | 85 |
| 5.1. Desarrollo del proceso propuesto | 85 |
| 5.1.1. Parámetros de usuario | 87 |
| 5.1.2. Cargar los grafos direccionales | 87 |
| 5.1.3. Cargar parámetros de fabrica | 91 |
| 5.1.4. Definición de los objetivos | 99 |
| 5.1.5. Generar población | 101 |
| 5.1.6. Evaluación iterativa | 102 |
| 5.1.7. Selección de soluciones finales | 103 |
| 5.1.8. Aplicación al concepto CAD | 106 |

| | |
|---|------------|
| 5.1.9. Manufactura del producto | 107 |
| 5.1.10. Integración de control basado en ROS para el sistema PRIA-UN | 110 |
| 5.2. Discusión de resultados | 115 |
| 5.2.1. Estructuras de Datos Basadas en Grafos para Modelar Componentes en el Diseño Generativo | 115 |
| 5.2.2. Desarrollo del Gripper Utilizando Diseño Generativo de Producto | 116 |
| 5.2.3. Evaluación de la Flexibilidad en el Proceso de Diseño | 118 |
| 5.3. Conclusiones del proceso | 119 |
| 5.4. Trabajos futuros | 120 |
| A. Anexo: Librería para clasificación por frentes de Pareto | 122 |
| B. Anexo: Algoritmo resolución de caminos en la estructura basada en primitivas | 125 |
| C. Anexo: Código G generativo para fresado en Y diámetro 6 mm | 129 |
| D. Anexo: Código G generativo para fresado en Z diámetro 6mm | 132 |
| E. Aplicación en LabFabEx-UN del diseño generativo secuencial | 135 |
| F. Protocolo despliegue del Gripper en el LabFabEx-UN | 145 |
| G. Anexo: Desarrollo método tradicional Gripper: Problema de diseño | 154 |
| G.1. Planteamiento del problema | 154 |
| G.1.1. ¿Quién? | 154 |
| G.1.2. ¿Qué y para Qué? | 154 |
| G.1.3. ¿Por qué? | 154 |
| G.1.4. ¿Dónde y Cuándo? | 154 |
| G.1.5. ¿Costos? | 155 |
| G.2. Planteamiento de subproblemas | 155 |
| G.2.1. Defina algunas características esperadas de su producto y el entorno de operación: | 155 |
| G.2.2. ¿Cuál es la función positiva primaria de su producto? | 155 |
| G.2.3. ¿Anticipa efectos negativos para su producto? | 155 |
| G.3. Identificación de contradicciones | 156 |
| G.4. Requerimientos del cliente | 156 |
| G.5. Especificaciones de ingeniería | 157 |
| G.6. Desarrollo y análisis de resultados del QFD | 157 |
| H. Anexo: Desarrollo método tradicional Gripper: Diseño conceptual | 160 |
| H.1. Análisis funcional | 160 |
| H.1.1. Diagrama de caja gris | 160 |

| | |
|--|------------|
| H.1.2. Descomposición funcional | 161 |
| H.2. Técnica de inventiva Triz | 162 |
| H.2.1. Sub-problema 1 | 163 |
| H.2.2. Sub-problema 2 | 163 |
| H.2.3. Sub-problema 3 | 164 |
| H.3. Generación de conceptos | 165 |
| H.3.1. Técnicas de creatividad | 165 |
| H.3.2. Mapeo función-concepto | 166 |
| H.4. Integración de conceptos | 168 |
| H.4.1. Árbol de clasificación de conceptos | 168 |
| H.4.2. Tabla de combinación de conceptos | 169 |
| H.5. Evaluación y selección de Conceptos | 169 |
| H.6. Transmisión efectiva del Concepto Global Dominante (CGD) | 169 |
| I. Anexo: Desarrollo primera iteración Gripper: Diseño detallado | 171 |
| I.1. Generación del diseño detallado | 171 |
| I.1.1. Diseño a nivel del sistema | 171 |
| I.1.2. Definición de la arquitectura | 173 |
| I.1.3. Componentes disponibles y normalizados | 174 |
| I.1.4. Selección de materiales y técnicas de producción | 175 |
| I.1.5. Restricciones espaciales | 176 |
| I.2. Planteamiento y desarrollo de los modelos | 176 |
| I.2.1. Modelos Gráficos | 176 |
| I.2.2. Modelos Analíticos | 180 |
| I.2.3. Modelos físicos y prototipos | 181 |
| I.2.4. Prototipos | 181 |
| I.2.5. Modelos físicos | 182 |
| J. Anexo: Desarrollo método tradicional Gripper: Diseño detallado | 189 |
| J.1. Transmisión efectiva del producto | 189 |
| K. Anexo: gripper_driver.py | 192 |
| L. Anexo: gripper_driver_ros_wrapper.py | 194 |
| M. Anexo: Manual de uso del Gripper | 196 |
| M.1. En PRIA-UN | 196 |
| M.2. Manualmente | 197 |
| Bibliografía | 199 |

Lista de símbolos

| Termino | Término |
|-----------------------|---|
| <i>API</i> | Interfaz de programación de aplicaciones (Application Programming Interface) |
| <i>ASMG</i> | Generador de modelos de simulación automáticos (Automated Simulation Model Generator) |
| <i>CAD</i> | Diseño asistido por computadora (Computer-Aided Design) |
| <i>CAM</i> | Manufactura asistida por computadora (Computer-Aided Manufacturing) |
| <i>CNC</i> | Control Numérico Computarizado (Computerized Numerical Control) |
| <i>CPS</i> | Sistemas ciber-físicos (Cyber-Physical System) |
| <i>CPU</i> | Unidad central de procesamiento (Central Processing Unit) |
| <i>CUDA</i> | Arquitectura Unificada de Dispositivos de Cómputo (Compute Unified Device Architecture) |
| <i>Digraph</i> | Grafo direccional (Direct Graph) |
| <i>DIMA – UN</i> | Grupo de trabajo en nuevas tecnologías de diseño y manufactura-automatización |
| <i>DGR</i> | Diseño Generativo Realimentado |
| <i>DSM</i> | Matriz de estructura de diseño (Design Structure Matrix) |
| <i>FDM</i> | Modelado por deposición fundida (Fused Deposition Modeling) |
| <i>FMS</i> | Sistemas de manufactura flexible (Flexible Manufacturing System) |
| <i>GPU</i> | Unidad de procesamiento gráfico (Graphics Processing Unit) |
| <i>Gripper</i> | Sujetador |
| <i>IIoT</i> | Internet industrial de las cosas (Industrial Internet of Things) |
| <i>LabFabEx – UN</i> | Laboratorio Fabrica Experimental |
| <i>MES</i> | Sistema de ejecución de manufactura (Manufacturing Execution System) |
| <i>Mipymes</i> | Micro, pequeñas y medianas empresas |
| <i>MUCO</i> | Objeto complejo de unidad menor (Minor Unit Complex Object) |
| <i>PCT</i> | tratado de cooperación en materia de patentes |
| <i>Pick&Place</i> | Tomar y colocar |
| <i>PLM</i> | Gestión del ciclo de vida del producto (Product Lifecycle Management) |
| <i>PRIA – UN</i> | Plataforma Robótica Industrial Automatizada. |
| <i>RAMI4,0</i> | Modelo de Arquitectura de Referencia Industria 4.0 (Reference Architecture Model Industrie 4.0) |
| <i>ROS</i> | Sistema operativo robótico (Robot Operating System) |

| Termino | Término |
|----------------|---|
| <i>SCARA</i> | Brazo robótico articulado de cumplimiento selectivo (Selective Compliant Articulated Robot Arm) |
| <i>SDV</i> | Vehículos auto conducidos (Self Drive Vehicles) |
| <i>SQL</i> | Lenguaje de consulta estructurada (Structured Query Language) |
| <i>STL</i> | Estereolitografía (Stereolithograph) |

1. Introducción

El objetivo principal de esta tesis de maestría es brindar apoyo a los procesos de creación de productos orientados a la integración con entornos de manufactura modernos, con el fin de facilitar la toma de decisiones y la implementación de los procesos de manufactura en fábricas. Estos entornos modernos engloban el uso de diversas tecnologías para transmitir, operar, almacenar y compartir información a lo largo del ciclo de vida de un producto y proceso.

En esta tesis de maestría, se centra en la etapa de diseño de productos, con el objetivo de desarrollar procesos semi-automatizados que reduzcan la intervención humana y aumenten la capacidad productiva de países en vías de desarrollo, como es el caso de Colombia. El apoyo del grupo en nuevas tecnologías de diseño y manufactura-automatización DIMA-UN y el laboratorio de fábrica experimental LabFabEx-UN permiten plantear escenarios de desarrollo en el contexto de la industria 4.0, brindando orientación basada en trabajos previos.

Es importante destacar que los sistemas de manufactura automatizados actuales son viables cuando se producen grandes cantidades de un producto. Sin embargo, esto deja rezagados a los países en desarrollo en términos de generación de conocimiento, ya que su producción es menor. Según cifras presentadas por la Superintendencia de Industria y Comercio en Colombia [5], las fábricas en el país presentan bajos índices en la creación de productos. Para abordar este desafío, se exploran alternativas a la creación de productos en entornos de manufactura discreta como la remanufactura, la manufactura por imitación y la personalización masiva. Estas alternativas involucran modificaciones realizadas a productos ya fabricados para satisfacer requerimientos específicos de los usuarios y complementan la aplicación del diseño generativo.

El diseño generativo se refiere a un conjunto de algoritmos que proporcionan alternativas de diseño al explorar y explotar los espacios de diseño de los componentes, evaluándolos en función de los requerimientos funcionales planteados [6][7][8]. En esta tesis de maestría, se plantean mecanismos para integrar y evaluar los requerimientos funcionales del producto como objetivos a evaluar en un sistema multiobjetivo, al establecer funciones que describen un comportamiento basado en su modelado y con en el análisis de características cualitativas que necesitan la clasificación basado en la experticia del diseñador. Además, a través de la realimentación de las diferentes etapas del ciclo de vida del producto simulado, se generan conceptos de componentes que pueden aplicarse en la creación de nuevos productos.

Este trabajo está dividido en cinco capítulos. El primer capítulo es la Introducción donde se desarrolla la justificación de la investigación, se establece la pregunta de investigación y se mencionan los objetivos propuestos a desarrollar. En el segundo capítulo se realiza una revisión de la literatura, lo que permite plantear una hipótesis de solución al problema de investigación y se establece la metodología del trabajo de tesis. El tercer capítulo presenta el planteamiento de la estructura de datos que permite modelar los componentes de un producto y condensar la información de diseño que ha sido dependiente de la experticia del diseñador para la toma de decisiones. El cuarto capítulo se enfoca en el desarrollo del ecosistema de diseño generativo realimentado en cada etapa del ciclo de vida, complementado con propuestas de desarrollo para fortalecer dicho ecosistema, con el objetivo de reducir las iteraciones en la manufactura y obtener un producto de manera más eficiente. Finalmente, en el quinto capítulo se establece un esquema de desarrollo de gripper limitado por la capacidad tecnológica del LabFabEx-UN, aplicando y seleccionando las herramientas del ecosistema que se pueden utilizar en el proceso de diseño, manufactura y uso.

Aunque se menciona la posibilidad de semi-automatizar el proceso de diseño esta tesis de maestría se centra en el desarrollo de conceptos que, en la práctica, permiten alcanzar ese objetivo pero acotando el alcance del trabajo investigativo. Todos los desarrollos propuestos se enfocan en ayudar a la toma de decisiones del diseñador, debido a que se desarrolla un entorno con múltiples rutas de desarrollo que brindan herramientas adicionales para modelar y evaluar conceptos. Además, este trabajo busca complementar el marco del diseño generativo para hacerlo más general y aplicable en más etapas del ciclo de vida del producto, además del diseño, generando una interacción en esquemas de manufacturas estructurados como lo es el LabFabEx-UN y su etapa de uso.

1.1. Justificación de la investigación

Esta tesis de maestría se plantea con el propósito de aumentar la capacidad tecnológica para la creación de productos, al soportar la etapa de diseño con procesos semiautomáticos que proporcionen ayudas al diseñador mientras incluye información obtenida de la fabricas de manufactura. Para entender el entorno en el cual se encuentra Colombia en términos de creación de productos en esta sección se desarrolla en primera parte el análisis de patentes y se desarrolla la exploración de complemento bajo el diseño generativo mientras se guía el desarrollo en un enfoque integral conocido como Diseño Integrado de Producto y Proceso (DIPP).

La creación de productos es necesario para un país porque puede generar empleo con el enfoque de creación de nuevas empresas o nuevas líneas de negocio. En Colombia se observa un déficit en la creación de producto de nacionales en el país, como lo plantea el reporte generado por el Departamento Nacional de Planeación (DNP), la Superintendencia de Industria

y Comercio (SIC), la Dirección Nacional de Derecho de Autor (DNDA), el Instituto Colombiano Agropecuario (ICA), junto con la Organización Mundial de la Propiedad Intelectual (OMPI) y la Misión Permanente de Colombia ante las Naciones Unidas [5].

Este reporte analiza el rango de patentes entre 2000 y 2016 en el país y se enfoca en los registros de Propiedad Intelectual, debido a que representan un herramienta de medición para el sector de innovación del país, según y lo expresan “pues contienen información completa sobre el conocimiento generado y sobre la procedencia y destino del mismo, a su vez el contenido de estos instrumentos es público”. Aunque el reporte incluye tres categorías el enfoque de análisis para esta tesis se centra en la Propiedad Industrial y en específico a la clasificación de tipo de solicitud de nuevas creaciones.

El reporte aclara en primera instancia que en Colombia el registro de patentes por vías nacionales de nuevas creaciones es de aproximadamente el 80 % para no residentes y 20 % para residentes. Lo anterior muestra como a nivel nacional las empresas internacionales dominan el sector y esto se agrava al observar que por el tratado de cooperación en materia de patentes (PCT) indican que casi el 100 % de solicitudes son extranjeras, por lo cual interna y externamente el país no es competitivo en la creación de productos.

La segunda instancia que nos describe el panorama sobre las nuevas creaciones incluyen las patentes de innovación y de modelo de utilidad. La primera hace referencia a una creación totalmente innovadora y la segunda a modificaciones o mejoras en un producto o servicio. Estas resultan ser importantes debido a que reportan que el porcentaje de patentes de invención a nivel nacional es de 16 %, sin embargo, de ese porcentaje el 42 % son nacionales y el resto internacionales. Mientras que para las patentes de modelo de utilidad reportan el 8 % del cual de este casi el 92 % son nacionales. Al tomar la cantidad de patentes de innovación y de modelo de utilidad realizados por nacionales se identifican que representan una cantidad cercana, sin embargo, se observa que los extranjeros están mayormente interesados en la protección por medio de la figura de patentes de innovación, mientras a nivel nacional el interés se centra en mejoras o modificaciones a un producto. Al igual se recalca el hecho que bajo la figura de participación internacional PCT es casi nula la participación de nacionales, e igualmente por las figuras nacionales el porcentaje de participación es mayoritariamente extranjero dejando relegado al país en ambos mecanismos.

Con lo descrito en el informe queda identificado por qué es importante la exploración y el desarrollo herramientas que permitan mejorar la creación de productos de un país por la innovación, por lo cual se extiende la exploración de procesos de adopción de creación en otros países.

En la revisión sobre la innovación en países de desarrollo se resalta la revisión de la literatura realizada por Al-Shami [9], en esta desarrolla el caso de estudio de Corea del Sur el cual pasa de ser una economía en vía de desarrollo a ser una economía desarrollada. Al-Shami describe

el desarrollo en los últimos 60 años de Corea del sur que comienza por la asimilación del conocimiento y después se enfoca en la innovación. Expresa en detalle una división en cuatro etapas: la primera etapa cuando un país no es dueño de la tecnología que usa; la segunda etapa un país desarrolla industria para imitar productos importados y satisfacer el mercado local; la tercera etapa se imita productos, pero para satisfacer un mercado internacional; la cuarta etapa añade innovación e investigación a los productos para ofrecer un producto diferenciador.

Otra revisión menciona en China que un proceso similar se puede observar con el modelo político de la etiqueta de “Hecho en China” y planes gubernamentales como “Hecho en China 2025” lanzado en 2015 como lo menciona Bartikowski [10] y por Thoben et al. [11], para darse a conocer a nivel mundial como el principal país manufacturero. Sin embargo, en el proceso de manufactura de productos de alta calidad se ha presenciado la misma modalidad de asimilación de conocimiento descrito en Corea del sur, al punto de crear empresas multinacionales con alto índices de similitud en productos de alta gama. En la actualidad China refuerza sus esquemas para enfocar la producción en productos de lujo o de alto valor, pero basado en el artículo de Bartikowski, puede jugarle en contra la percepción global de “País de manufactura” para la conversión a ser referente en “País de diseño”, debido al enfoque de producción de masa genera una contradicción al establecerse en una manufactura de personalización donde la cantidad de producción es reducida y lo que destaca es la manufactura de características diferenciales.

Se destaca de estos modelos de dos países asiáticos que la creación de productos requiere el poder entender y manipular conocimiento generado de otros productos fabricados previamente. Una limitante radica en que las etapas de diseño para la creación de un producto dependen de la experticia de un diseñador, como lo presenta Ullman en libros guías de diseño de producto contemporáneas [12] o Farid et al. en el diseño axiomático [13]. Por lo tanto, los conceptos desarrollados no son fácilmente transmitidos por otro diseñador en la creación de un nuevo producto. Esta práctica genera que se excluyan intencionalmente conceptos que pueden solventar un problema, por razones como la falta de comprensión o la complejidad del modelamiento de fenómenos que se encuentran en otros dominios del área de conocimiento del diseñador. Aunque la transmisión de conocimiento entre diseñadores para la creación de producto no es clara, para la “remanufactura”, la “manufactura por imitación” o la “personalización masiva” se desarrollan algunos conceptos en torno al mejoramiento de productos como lo expresa Norton [14] en el diseño detallado.

El Diseño Integrado de Producto y Proceso (DIPP) busca la sinergia entre el diseño del producto y los procesos de fabricación que se emplearán para su producción. En lugar de abordar estas dos áreas de manera independiente, se busca una colaboración estrecha entre los equipos de diseño y fabricación desde las primeras etapas del proceso creativo. Esto permite considerar simultáneamente aspectos como la funcionalidad, la manufacturabilidad,

la eficiencia en la producción y la sostenibilidad del producto final.

Se detalla la necesidad de revisar la metodología en la cual interactúa el diseñador en la etapa de diseño y conectado con los procesos de manufactura para la creación de nuevos productos. Las metodologías de diseño actuales incluyen la interacción con recursos computacionales como lo son softwares “asistidos por computadora”. Existen dos representantes principales de dichos softwares que son el diseño asistido por computadora CAD y el de manufactura asistido por computadora CAM. Estos se encuentran en la actualidad arraigados a los procesos de diseño, debido a un entorno de softwares que han dado pautas y desarrollos para establecerse en el mercado como lo son Autodesk, Catia, Solidworks, MasterCam, etc.

Las limitantes que presentan estos softwares se centran en la estandarización y en la interacción con otro software. El desarrollo de estos softwares busca abarcar una gran cantidad de etapas del ciclo de diseño y manufactura dentro de su mismo entorno, por lo que establecen diferentes estructuras de datos internas que describen los componentes y operaciones, y dada la falta de estandarización no es posible abarcar toda esta información para transmitir a otros software y seguir su procesamiento.

Adicionalmente la necesidad del uso de herramientas computacionales se ha representado en la adquisición de software privado y el desarrollo de herramientas de código abierto, lo que permite ampliar el espectro de aplicaciones en las cuales se investiga y se despliega en las fabricas de manufactura. El diseño generativo es una vertiente actual que se enfoca en la automatización de procesos en la etapa de diseño para la generación de conceptos funcionales, estos evalúan diferentes requerimientos en un producto como lo describen Mountstephens, J et. al [6] o BuHamdan, S et al. [15] en sus revisiones de la literatura. Es de desatacar que su significado no ha llegado a una unificación por la diversidad de temas de enfoque observados en los artículos revisados, pero que van conservando características de desarrollo que dan guía al objetivo principal del diseño generativo, el de explorar y explotar espacios de diseño para dar posibles soluciones en la creación de un componente.

Este enfoque de diseño generativo proporciona una visión de procesos computacionales que se pueden interconectar y realimentar con información que se tiene del proceso como de productos. en este sentido brinda un espectro de búsqueda de algoritmos de diseño generativo que permitan proporcionar sistemas semi-automáticos que permitan integrarse con diferentes software al proveer la información necesaria y que permita su manipulación sin necesidad de la interacción constante del diseñador, lo cual cumple con la intencionalidad de explorar y explotara espacios de diseño para la generación de nuevos productos.

La necesidad de explorar esta técnica computacional es que permite la exploración de múltiples espacios de diseño para la generación de soluciones en ambientes virtuales. Lo que diferencia a lo descrito por los países asiáticos como Corea del Sur y China previamente,

donde las soluciones generadas se dieron por la concepción de imitación basado en la fabricación por masa y procesos de mejoramiento posteriores. Estos avances tecnológicos guían a que se brinden soportes adicionales a los diseñadores al momento de generar un nuevo producto, y que se permita realizar en un ambiente virtual.

1.2. Pregunta de investigación

¿Cómo se puede desarrollar una metodología que integre algoritmos modernos y enfoques de automatización y manufactura en las diferentes etapas del ciclo de vida de un producto, permitiendo así la semi-automatización de la etapa de diseño y prescindiendo de la necesidad de depender de la experiencia adquirida en métodos de manufactura en masa o la experticia de un diseñador?

1.3. Objetivo General

Desarrollar una metodología de diseño generativo de producto como soporte a la etapa de diseño, con enfoque al ciclo de vida de producto en sistemas de manufactura flexibles.

1.3.1. Objetivo Específicos

1. Formular una estructura de datos que permita el flujo de información de parámetros de los componentes de un producto previamente diseñado, para la realimentación en la creación de nuevos productos con base en el modelamiento del diseño generativo.
2. Desarrollar un “gripper” que combine el modelamiento de componentes desarrollados en el objetivo específico 1, para operaciones “Pick & Place” como concepto de creación de nuevos productos en el Laboratorio de Fabrica Experimental - UN (LabFabEX-UN)
3. Referenciar la metodología de diseño del “gripper” bajo métrica específica de flexibilidad, para identificar sus contribuciones en la cantidad y complejidad de las piezas generadas en sistema de manufactura flexible.

2. Revisión de la literatura

La revisión de la literatura se centra en explorar el diseño generativo como una técnica que permite automatizar partes del proceso de diseño de producto basado en el desarrollo en entornos virtuales, lo que solventa la dependencia de explorar diseños basados en la manufactura en masa. Adicionalmente se centra en la exploración de desarrollo de los esquemas de manufacturas actuales enfocados en la creación de productos, para identificar la integración completa del diseño generativo con los esquemas de producción.

Este capítulo empieza por describir el diseño generativo bajo un enfoque de creación de producto, continua desglosando la integración con los esquemas de manufactura, para complementar la búsqueda con un marco teórico que complementa los conceptos usados en el desarrollo de esta tesis de maestría, mientras en el desarrollo del capítulo se desarrolla la hipótesis general de este trabajo.

2.1. Diseño Generativo

El diseño generativo se basa en una serie de algoritmos para proveer alternativas de diseño, basado en un proceso parcialmente automatizado de exploración de los espacios de diseño bajo unos requerimientos funcionales establecidos [6][7][8]. Es de notar que la variabilidad se da en la definición de los requerimientos funcionales debido a lo observado por Mountstephens, J et. Al [6] y BuHamdan, S [15] en sus revisiones de literatura. Los artículos analizados presentan diferentes objetivos a resolver y no hay un método unificado para evaluar el éxito de los modelos. Adicionalmente añaden que la mayoría de los artículos se pueden considerar pruebas de concepto, debido a que no tienen un enfoque para producción.

En la figura **2-1** se aprecia una revisión del léxico usado en el diseño generativo por medio de la herramienta VOSviewer. En esta revisión terminológica se usaron los títulos y resúmenes de 300 artículos organizados por relevancia por el gestor bibliográfico SCOPUS. La revisión bibliográfica nos brinda una idea de lo descrito anteriormente sobre la variabilidad de los conceptos en el cual no se resaltan agrupaciones dominantes, sin embargo, se pueden extraer palabras claves que en conjunto describen lo que representa el diseño generativo.

La visualización de la figura **2-1** nos muestra en color amarillo la tendencia actual de investigación, el cual incluye el desarrollo de la manufactura aditiva y conceptos ligados al desarrollo en ingeniería. Lo anterior concuerda nuevamente con la revisión de la literatura de

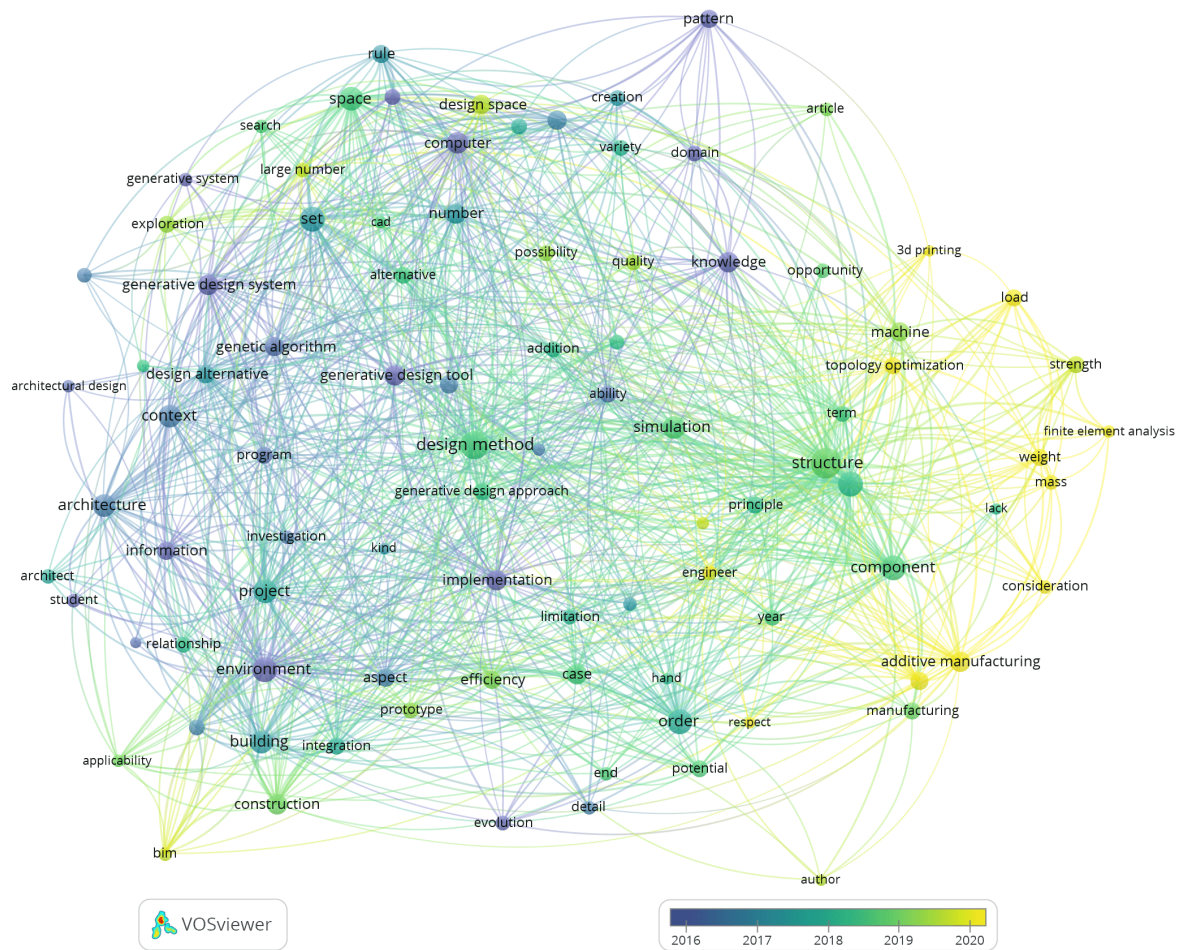


Figura 2-1.: Análisis - terminología sobre “Diseño Generativo”.

Fuente: Creación Propia - VOSviewer

BuHamdan, S et al. [15] y Mountstephens, J et al. [6], en el cual mencionan que inicialmente la mayor cantidad de investigaciones se centraban en el área de arquitectura, que se aprecia en color morado en la figura 2-1, pero que en la actualidad abarca todo tipo de producto manufacturable lo que hace referencia a la ingeniería.

2.1.1. Experticia del diseñador

Mountstephens, J et. Al [6] expresan que recientemente las maquinas están participando directamente del proceso de creación, refiriéndose a la capacidad que tiene estos sistemas de funcionar cada vez con una interacción menor de un diseñador. Esta disminución de la interacción del diseñador responde a un problema generalizado de las limitaciones de exploración que impone él mismo al diseño, clasificado como la experticia del diseñador [16][15][17]. Sin embargo, la disminución de la interacción del diseñador en la exploración no

lo reemplaza, sigue siendo necesario para las etapas posteriores, pero partiendo de un conjunto de soluciones [8][18][19].

La problemática se expresa en que no hay un estándar para representar un componente y como operarlo, por tal motivo, los diseñadores no pueden adicionar su experticia en los archivos que manipulan virtualmente [6]. También la variabilidad de los objetivos a operar en los diferentes sistemas de diseño generativo implica que se describan de diferente manera las características que definen a los componentes y su transmisión sigue siendo complicada.

Por esta razón los softwares comerciales han optado por limitar los objetivos al aplicar estas técnicas en varios casos a la resolución bi-objetivo de la disminución de la masa mientras se garantiza una rigidez. Esta tendencia se puede observar en la figura **2-1** en el cual se observa que en color amarillo los términos de peso y masa están cercanos a la rigidez, lo que indica una correlación. Autodesk es un ejemplo claro de esta propuesta en su software Fusion 360, en el cual integra su funcionalidad de diseño generativo. Wu, J et al. [20] y Pilagatti, A et al. [18] muestran las implicaciones del diseñador en el ajuste de parámetros de frontera para su resolución usando este software y como se genera una serie de alternativas que quedan a decisión del diseñador seleccionar.

Adicionalmente la interfaz de estos softwares comerciales es intuitiva y gráfica, lo que permite una interacción mayor con el diseñador. Mientras como lo menciona Li, H et al. [21] y Mountstephens, J et al. [6] la complejidad requerida a los diseñadores para aplicar estas técnicas de diseño generativo es mayor y en algunos casos se plantea el entrenamiento para el uso, con el fin de definir las reglas, características, descripciones, interacciones y demás valores que intervienen estos sistemas.

Se observa actualmente un incremento de complejidad en diseños mecánicos como lo mencionan Ullman [12] y Farid et al. [13], donde un solo diseñador es insuficiente y se necesita establecer una estructura jerárquica basada en distintos niveles de abstracción. Bajo este esquema el nivel de abstracción más bajo lo limita la capacidad del diseñador y su experiencia, por lo cual la multidisciplinariedad entra a ser otro factor que el diseño generativo puede satisfacer al condensar la experticia en una estructura de datos [15]. Por lo tanto, una hipótesis se centra en poder almacenar en una estructura de datos la información que es necesaria para describir y operar estos sistemas. Para permitir una interacción multidisciplinar y de estandarización para la industria.

2.1.2. Integración etapas de diseño

Adicionalmente se hace un énfasis de dos fases en la etapa de diseño, el diseño conceptual y el diseño en detalle. Esta separación se da principalmente para identificar en que fase se enfoca los desarrollos en el diseño generativo, debido a que Nagaraj, D et al. menciona que en pocos

artículos el diseño generativo abarca ambas etapas [16][6]. Esto representa una limitación para que las soluciones sean manufacturadas y se queden como pruebas de concepto. En este sentido también se abre la brecha entre los softwares comerciales y los desarrollos de los artículos, donde en los softwares comerciales el enfoque es la manufactura de un producto, mientras que estos trabajos quedan limitados al planteamiento del diseño o la generación de una cantidad de conceptos que den las bases al diseñador de entrar a la etapa de diseño como lo muestra Khan, S et al. [22][23].

Por lo tanto, un puente para vincular estas dos fases es propuesto por Nagaraj, D et al. [16], se basa en trabajar un descriptor que después sea una representación en 3d mientras en cada iteración se obtienen valores de evaluación que optimicen los requerimientos establecidos. En este caso cita funciones que pueden ser entrenadas por medio de inteligencia artificial o aprendizaje de maquina para obtener los descriptores de un componente, estos modelos por lo general requieren una gran cantidad de datos etiquetados para poder proveer un modelo cercano, lo cual puede ser una limitante adicional para plantear esta integración de ambas fases.

Es notable que, para el enfoque de diseño conceptual se plantean algoritmos con diversas estructuras para operar y explorar los conceptos en el diseño generativo, que tradicionalmente se realiza por medio de dibujos o bocetos hechos a mano, sin embargo, para el diseño en detalle está establecido el uso de software CAD [16][6]. Por lo cual dos de las principales representaciones de software CAD se expresan a continuación debido a la influencia que tienen en como almacenan la información de forma de un objeto [17].

Diseño CAD por superficies discretas

Es un tipo de representación que permite almacenar la información del modelo como una superficie que no es continua, expresando una discretización de la superficie por medio de representaciones geométricas como lo son puntos, rectas, planos, entre otros [17][24]. Debido a su condición de no continuidad existen diversos modelos matemáticos para realizar una interpolación entre puntos vecinos y generar una resolución mayor para simular la representación de una superficie continua. Las mallas, nube de puntos, pixeles y voxels representan la clasificación usada mayormente para la representación gráfica de los modelos. La resolución se expresa por cantidad de unidades discretas y por la separación entre estas.

Nagaraj, D et al. en su revisión expresa que los formatos más comunes encontrados para la representación 3d son las nubes de puntos y los voxels [16]. Es común encontrar que se estudie en planos 2d donde la exploración se realiza por medio de pixeles como lo muestran Oh, S et al. [25] y Ghannad, P et al. [26], mientras Kallioras, N et al. tienen una propuesta similar donde lo interesante es que los pixeles toman valores entre 0 y 1 que representan la densidad [27][28]. Adicionalmente aclara que en trabajos futuros se establece su representación por voxels.

Es utilizado en modelado artístico donde se pueden expresar superficies complejas e irregulares por el desplazamiento, adición o substracción de las unidades discretas que lo definen. Lo anterior no excluye que se pueda usar en ingeniería, en donde para el diseño de productos bajo técnicas como el diseño generativo también se hace uso de estas herramientas CAD, como lo muestra Garcia en su trabajo de trayectorias de mecanizado [29] o Sarmiento en un mecanizado CNC para un proceso de ingeniería inversa [30].

También se usan este tipo de descripciones en los softwares CAD en procesos de análisis de elementos finitos para resolver el sistema, por medio de programas de código abierto como OpenFoam o NVIDIA Modulus y planteamientos específicos en diseño generativo como el expuesto por Kallioras, N et al. [28]. Adicionalmente en maquinaria CNC como las impresoras FDM usan este tipo de archivos para calcular sus trayectorias.

En el diseño generativo otro concepto utilizado es el concepto de forma de gramática, que incluye una serie de reglas para definir la forma de un objeto usando las representaciones geométricas [31]. Es notable el uso de estas reglas en los primeros trabajos ligados a la arquitectura y recientemente en el enfoque de obtención de formas novedosas por la generación automática de geometrías complejas [17]. En varios artículos se referencia la necesidad de usar esta técnica para la obtención de formas novedosas [25], mientras se ejecuta en tiempos más prolongados y con más detalle que al compararlo con un diseñador.

Diseño CAD por representación de contornos

Es la representación más usada en ingeniería debido a que se basa en representación de contornos por medio de funciones. Esta representación es más exacta comparada con los modelos CAD por superficie y debido a que la representación de la superficie es una función continua, no presenta problemas de resolución. Sin embargo, presenta limitantes de forma debido a que sus operaciones se basan en acotar funciones, estas generalmente se definen en planos de dos dimensiones y para conseguir superficies complejas en tres dimensiones, donde su acotación se realiza por la intersección de múltiples planos 2d.

Otra ventaja que presenta computacionalmente es el peso de la representación del objeto, es mucho más liviano almacenar una función, que almacenar N cantidad de puntos basado en la resolución. Aplicar transformaciones a todo el objeto es igualmente más rápido al aplicarla a una función que aplicarlo una por una a todos los puntos. Vale aclarar que una de las ventajas que cuenta estos modelos por contornos es la posibilidad de modificar los parámetros de las funciones fácilmente, lo que permite automatizar procesos. Una gran parte de los artículos de diseño generativo hacen uso del diseño paramétrico para generar la variación en los componentes [15][16][8].

El diseño paramétrico permite acceder y manipular los valores de las variables de las funciones (parámetros) y hacer una actualización del modelo. Un ejemplo de esta aplicación se aprecia

en la figura 2-2, en el cual por medio del software Fusion 360 de Autodesk se modifican algunos parámetros. En la figura de la izquierda se observa cómo se puede variar el ancho mientras la altura permanece constante y al mismo tiempo se modifica el grosor de las extrusiones interiores. A la derecha de la figura observamos como se manipula el componente para obtener óvalos con diferentes diámetro interiores y exteriores u obtener una círculo perfecto. Al igual se aclara que la modificación se realiza en algunas variables descriptivas y no en un escalamiento de todo el objeto.

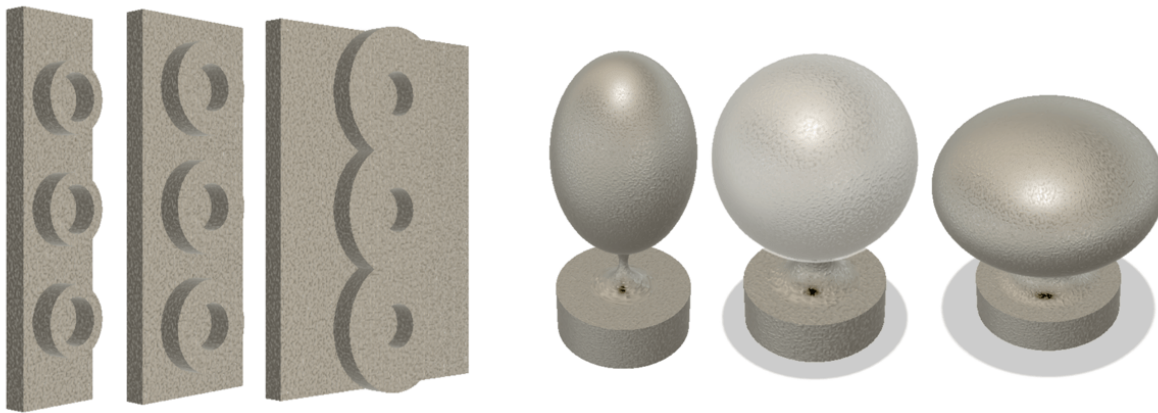


Figura 2-2.: Ejemplo de diseño paramétrico.

Fuente: Congreso CIMMX [1]

Ambas representaciones de mallas y de contornos son usadas en la mayoría de artículos revisados en esta tesis [16], por lo cual se observa que hay dos caminos generales para la representación de diseños y la selección las toma el diseñador basado en las ventajas descritas, sin embargo, la decisión del tipo de representación en la mayoría de artículos no incluye la consideración de manufactura, como sería la integración en un sistema de producción.

La principal desventaja que presenta representaciones discretas como lo son las mallas es la falta de interacción con los procesos de manufactura. Mientras que la principal desventaja que muestra el diseño paramétrico es que la diversidad está limitada por las variables que establece el diseñador, lo cual implica también una representación de la experticia del diseñador y que no se genere una exploración total de los espacios de diseño.

Es de tener en cuenta lo que mencionan Oh, S et al. [25] acerca de que la parametrización en un enfoque matemático es complicada cuando se quiere describir superficies complejas, debido a la dificultad de definir todas las variables y que entre ellas muestran una correlación entre variables demasiado fuerte. Sin embargo, no implica que la metodología de superficies como mallas sea más sencilla, dado que para este tipo de descripciones es común utilizar técnicas de inteligencia artificial o aprendizaje de máquina para generar el modelado del

componente a modificar. Hyunjin, C y Kallioras, N et al. expresan que el futuro del diseño generativo para ambos casos va a estar soportada por sistemas de inteligencia artificial para el mejoramiento continuo en cada etapa [19][28][27].

2.1.3. Mejoramiento de producto

Nagaraaj, D et al. indica que algunos artículos mencionan el diseño generativo con “base al modelo” con un foco único al mejoramiento de un diseño ya existente con técnicas como la optimización topológica que remueve el exceso de material [16]. Al igual otras muestras de mejoramiento de un producto parten de un componente ya existente e identifican las características que debe mantener, luego el programa genera una forma que vincula las características, como lo propuesto por Li et al., 2018 [32], Jiang, Z et al. [7] y AUTODESK, en la implementación de un módulo de diseño generativo en el software Fusion 360.

Wolniak, P et al. plantea un sistema buscando la optimización donde identifica un análisis funcional del producto respecto a los distintos valores que puede tomar el modelado CAD para satisfacer las necesidades [8]. Lo establece como un sistema multiobjetivo que debido a la cantidad de parámetros e iteraciones es necesario una gran cantidad de tiempo de resolución.

2.2. Integración en esquemas de manufactura

Pibal, S et al. [33] y Jiang, Z et al. [7] expresan que el diseño generativo puede ser una herramienta útil para responder a la manufactura de personalización en masa. Ambos expresan la necesidad de tener realimentación del ciclo de vida del producto, para que en una etapa de diseño temprano se puede generar soluciones que se adapten a las necesidades específicas de cada usuario. Cabe resaltar que en el caso de Pibal, S et al. plantean una integración de softwares comerciales en arquitectura para su aplicación, lo que da muestras de estandarización en esta área, mientras en la ingeniería como se mencionó en la sección anterior tiene estas limitaciones.

El proceso de manufactura de un producto abarca diversas etapas entre ellas la etapa de diseño y fabricación, las cuales son críticas y deben ser ejecutadas de manera precisa para garantizar la calidad del producto final. Sin embargo, debido a la complejidad del proceso, algunas empresas pueden optar por tercerizar uno o ambos servicios. En algunos casos, empresas especializadas en diseño pueden encargarse de la creación del producto, mientras que otras empresas se enfocan en la fabricación.

Aunque la tercerización de servicios puede ser beneficiosa para algunas empresas, es importante tener en cuenta que esto puede llevar a una posible falta de integración entre las etapas de diseño y fabricación. Por tal motivo los sistemas PLM (Product Lifecycle Management) y

MES (Manufacturing Execution System) son piezas clave en la etapa de diseño y manufactura de un producto, ya que permiten la gestión y control de toda la información y procesos que intervienen en la fabricación del producto.

El sistema PLM se enfoca en la gestión del ciclo de vida del producto y está diseñado para ayudar a las empresas a administrar y controlar la información relacionada con el diseño, la fabricación, la documentación y la gestión de cambios del producto. De esta forma, se asegura que la información esté disponible para todas las áreas involucradas en la producción del producto.

Por otro lado, el MES se centra en la ejecución de la producción, monitorizando y controlando los procesos de fabricación en tiempo real, asegurando la calidad del producto y optimizando la eficiencia de la producción. También proporciona información relevante a los trabajadores de la planta, como instrucciones de trabajo, hojas de ruta, registros de producción y seguimiento de los materiales utilizados [34].

Como lo aprecia Jauhar, T et al. [35] la mayoría de los sistemas PLM están enfocados al área de diseño, mientras que la mayoría de los sistemas MES responden al área de producción. Esta se expresa como una problemática al momento de responder a procesos de creación de productos, debido a que la integración entre estos sistemas es reducida [36][37][35]. Dada la separación de estas dos áreas y factores como la tercerización se establecen soluciones que provean un sistema que conecte la información de ambas etapas para que este disponible en todo momento.

Según Fox, S la implementación de sistemas de diseño generativo no se da en pequeñas empresas debido a que no tienen la totalidad de recursos para desarrollarlo [17]. Sin embargo, establece que apoyos gubernamentales o la experiencia de centros de investigación pueden disminuir esta brecha al proveer soluciones aplicables en estos entornos.

Nagaraj, D et al. aclara que no solo es llegar a una representación del CAD para asegurar que sea manufacturable y que modelamientos y simulaciones adicionales pueden ser necesarias [16]. BuHamdan, S et al. establece en su revisión literaria del diseño generativo que en la actualidad es un problema aún vigente la integración entre herramientas analíticas con herramientas de modelado y que deberían vincularse a plataformas de manejo del ciclo de vida del producto (PLM)[15] y/o adicionalmente plataformas de producción, aunque en 2009 ya se había identificado esta problemática.

Dada la clara separación que se presenta entre estos dos softwares MES y PLM se desarrolla una revisión de como se establecen soluciones para el desarrollo de productos en cada tipo.

2.2.1. Software MES en la creación de productos

Los softwares MES son muy eficientes para transmitir la información obtenida del proceso entre las áreas de producción e integrar los sistemas de nivel de planta con el nivel empresarial comparados con los procesos manuales y por medio de papel físico [38][34]. Tienen la finalidad de mejorar los procesos de producción con el uso de métricas para soportar la toma de decisiones para distribuir las actividades.

Los softwares MES usan la norma ISA 95 como un marco de referencia de la integración o interoperabilidad entre el control de procesos y la planificación y gestión, por lo cual se observa la estandarización en estos temas. Al ser un marco de referencia permite que las fábricas adapten las características que se creen necesario en cada entorno, aunque por la complejidad que requiere una implementación completa se identifica que la cantidad de softwares que incluyen todas las características de integración son muy escasos. Pero indica que la creación de un sistema MES no requiere el uso de software específicos comerciales y que manejarlo en un entorno de desarrollo personalizado para cada fabrica es una alternativa comúnmente usada [39][40][38][41][42][43].

El desarrollo de un entorno en la nube de sistemas MES ha sido desarrollada por varios autores, partiendo de la disponibilidad en cualquier locación de la información y de la conectividad inalámbrica que se ha establecido en la industria [44][45][46][47][39]. Teniendo en cuenta el desarrollo abierto de estos softwares MES un entorno de desarrollo web con conexión en la nube es viable para las pequeñas y medianas empresas, por la disponibilidad de diferentes frameworks de diseño de pagina web y por la infraestructura en la nube por terceros que se encargan del mantenimiento y seguridad de estos sistemas como lo son Azure de Microsoft, Firebase de Google, AWS de Amazon, entre otros.

Una limitante se centra en acceder a los datos de maquinaria porque sus softwares son cerrados y se usan técnicas como exportar la información en un tipo de archivo estructurado para ser procesado por otro programa y convertirlo a la estructura de datos planteada en el sistema MES, como lo muestra Mahmoud, M et al. [38] donde la maquinaria usada exportaba PDF pero tuvieron que convertir esa información en un archivo Excel para incluirla en el sistema MES y que se hiciera automáticamente y en tiempo real.

La flexibilidad en los sistemas MES es un vacío que falta mejorar como lo menciona Ko, M et al. [46] al decir que ante cambios en el sistema como puede ser la integración de un nuevo producto los softwares MES necesitan un desarrollo en las diferentes etapas de producción para integrarlo. En este enfoque el LabFabEx-UN por medio del desarrollo de PRIA-UN ofrece flexibilidad en la integración de nuevos productos en su sistema MES por medio la estandarización y el uso de una interfaz de usuario creada por NodeRED que permite configurar nuevos productos en bloques de operación secuenciales [48]. Al igual menciona que una de las direcciones en las cuales se debe desarrollar en los sistemas MES para responder

a los planteamientos modernos de industrias automatizadas es el internet industrial de las cosas IIoT.

La implementación de plataformas IIoT permiten mejorar esta integración y adaptarse a nuevos productos disminuyendo la necesidad de interacción humana directa [45][49][46]. Mantravadi et al. [44] detallan que muchos de las implementaciones IoT se hacen preferiblemente en las instalaciones y no en la nube por los problemas en latencia y disponibilidad de red. En la actualidad ya no es suficiente tener la información sino poder procesarla en tiempo real, por tal modo los periféricos IIoT están proporcionando esta información a sistemas MES en tiempo real y de manera inalámbrica para permitir la flexibilidad de proceso [38][50][46].

Un desarrollo de trazabilidad es mostrado por Barata, J et al. [45] donde usan tecnologías como códigos de rápida respuesta QR (Quick Response code) e identificadores de radio frecuencia RFID para transmitir a los softwares MES información del ciclo de vida del producto en la fábrica [49]. Otro sensor que permite la realimentación en un software MES son las cámaras, que con el transcurrir del tiempo siguen mejorando características de resolución y sus costos han disminuido por la producción masiva para distintas actividades.

D'Antonio, G et al. [51] en el desarrollo de un sistema que integra el diseño de manufactura aditiva con los softwares MES, da un ejemplo del uso de estos sensores para el análisis, sin embargo, el sensor usado necesita mayor resolución por lo cual muestra las limitantes que aun pueden presentar estos sistemas de visión en tareas específicas. El área de visión de maquina ha tenido avances recientemente por la disminución de costos de tarjetas gráficas, donde el procesamiento gráfico se ha enfocado en programación paralela para realizar estas tareas de cómputo pesadas, mostrando que las cámaras pueden ser una gran fuente fiable de información para cumplir con tareas de trazabilidad.

La simulación en entornos MES es limitada como lo menciona Naedele [52] por lo cual procesos de simulación y proyecciones futuras son limitadas. Masood, T et al. [53] también muestra la necesidad de realimentar el proceso con información simulada para analizar el comportamiento en la fabrica y ayudar a la toma de decisiones.

2.2.2. Software PLM en la creación de productos

Los sistemas PLM presentan similares problemas que los mencionados en los sistemas MES, donde los mas claros son la integración, la estandarización, la flexibilidad y el análisis de gran cantidad de datos como lo identifican Habib, H et al. y Hayat, M et al. en las respectivas revisiones de la literatura del manejo de cambios de ingeniería con un enfoque en el ciclo de vida del producto y la identificación de retos de los sistemas PLM existentes [54][55]. Lo anterior se explica en que los sistemas MES y PLM se basan en el desarrollo de software para transmitir y procesar información, la diferencia se presenta en que mientras los problemas en

los sistemas MES se basan en adquirir información de maquinaria, en los sistemas PLM se representan en obtener información de los softwares de diseño asistidos como los son CAD.

Una condición adicional que implica los softwares PLM es la diversidad de dominios que debe interconectar para obtener la información de diferentes etapas, por lo tanto, el soporte de sistemas es complejo por lo cual la integración y estandarización también juega un rol importante para solventar esta falta [56]. Habib et al. [54] también añade el uso de frameworks que integra el seguimiento de ciclo de vida de producto como servicios que se coordinan en diferentes dominios del PLM, pero que depende de la estandarización de la información del producto como alternativas para generar un soporte de estos sistemas.

La estandarización de la información es estructurada principalmente por esquemas XML, que provee niveles jerárquicos de información [54]. Esta estructura en modelos en la nube es similar a las llamadas por el protocolo HTML para transmisión de información entre servidores por llamados REST. Entonces modelos de bases de datos o información estructurada se pueden concebir para que se realice pensado en esquemas como JSON y comunicación con bases de datos estructuras como sería SQL. Es de notar que los datos heterogéneos como los archivo CAD presentan una falta de integración con estándar web, lo que genera una falta en como manipular esta información en la nube [57].

Rädler, S. et al. [56] parten de la automatización del diseño por medio de algoritmos y combinado con el conocimiento del producto permite explorar espacios de diseño mientras se minimiza costos y errores. Por lo tanto, con sistemas de integración de áreas se pueden emplear para el entendimiento de la información y usarla para el desarrollo de producto como lo son la ciencia de datos, la minería de datos, la inteligencia artificial, redes neuronales, algoritmos genéticos y el aprendizaje de máquina. Sin embargo, Rädler, S et al. muestran en la encuesta generada como la falta de conocimiento de como hacerlo y la calidad y disponibilidad de la información son de los principales retos a solventar, por lo que las estrategias se basan en muchos casos en la dependencia de la experticia del diseñador.

La pérdida del conocimiento del diseñador por no estar incluida en el marco de flujo digitalizado de la empresa es uno de los principales objetivos por solventar en un futuro según lo expresado por Rädler, S. et al. [56]. Al igual Habib et al. [54] complementa con que no solo es tener extensa información, sino que esta debe estar estructurada, organizada, comprensible, presentada de una manera intuitiva y que no presente pérdidas de resolución. Por lo cual el planteamiento de estructuras de datos que realizan el almacenamiento de información y permiten acceder y realizar operaciones entre los datos es una alternativa que se plantea en el estudio de esta tesis de maestría.

Respecto a la información no estructurada que se utiliza en estos sistemas se encuentran los modelados CAD. Nzetchou, S et al. [57] en su revisión de la literatura expresan como la estandarización en formatos estándar básicos no es suficiente, debido a que ninguno de estos

archivos es nativo para cada software y por lo general internamente estos modelos estándar debe ser complementada con otro tipo de información que no es compartida en los procesos de exportación por la falta de estandarización, por lo cual también se observa la experticia del diseñador juega un rol importante en la capacidad de manipulación de estos modelos, ya que la experticia del diseñador permite extraer esta información.

Esta problemática también se refiere a la capacidad de reutilizar los modelamientos o de realizar modificaciones, en el cual detallan los autores que hay dos maneras para expresar estos cambios cuando se parte de modelos estándar: 1) En primer lugar que el diseñador genere nuevamente un modelo nativo partiendo del estándar. 2) Con una metodología sistemática que permita al diseñador semánticamente traducir las modificaciones al objeto de modelamiento estándar [57].

2.2.3. Manufactura flexible

En los escenarios de manufactura discreta o de producción en masa personalizada, las líneas de producción flexible pueden responder a la variabilidad de productos [36][44], sin embargo, la flexibilidad no se debe considerar únicamente en cuestión de proceso sino también en los diseños como lo menciona Wolniak, P et al. [8], por lo cual varios artículos resaltan la creación de plataformas como un medio eficaz de representación de diseño flexible, destacando la plataforma de familias de productos y en el cual la modularidad es la principal herramienta para la personalización [58].

Mientras los desarrollos en ingeniería se centran en un componente, en la arquitectura criterios de evaluación pueden abarcar la interacción entre ellos, donde se destaca temas de flexibilidad, costos, energía, entre otros [33].

El uso de manufactura aditiva en el diseño generativo es generalizado debido a la capacidad de manufacturar geometrías complejas, que por medio de otras técnicas como la de remoción de material no es posible [25][18]. D'Antonio, G et al. [51] identifican que la interacción del diseño para la manufactura aditiva con sistemas de producción como el software MES falta desarrollo y por eso proponen una metodología para ir validando la manufactura en un software MES. Identifican dos factores clave en el desarrollo de esta integración la primera la necesidad de contar con periféricos especializados como lo son cámaras inteligentes y una alta capacidad computacional para procesar estas imágenes en tiempo real.

Naedele, M et al. [52] hace énfasis que uno de los problemas en los sistemas MES es la falta de unificación de la información transmitida, donde toda la información se presenta como diversos tipos de información heterogénea. Pero esta afirmación como se aprecia en la revisión bibliográfica es también un problema de sistemas PLM y del diseño generativo, por lo cual una integración general de todas las etapas puede guiar a procesos de estandarización.

La integración entre los softwares se puede realizar por medio de API [52][59], Mantravadi, S et al. [44] y Hayat, M et al. [55] concluyen que la interoperabilidad es esencial cuando se habla de manufactura flexible y que puede ser guiada por el uso de APIs y en específico de las de código abierto. Kopei, V et al. [59] reafirma la necesidad de integrar diferentes softwares de código abierto para cerrar el seguimiento de ciclo de vida de un producto desde la conceptualización hasta la evaluación.

La integración propuesta en los sistemas PLM y MES se da por actualizar los métodos de transmisión de información del papel físico a un entorno virtual estructurado. En estos entornos se puede procesar los datos obtenidos basados en la exportación en formatos de manipulación de datos como lo es el software Excel, pero en entornos modernos se desarrolla el uso de bases de datos estructuradas como lo sería SQL que puede ser desplegada en un entorno en la nube [38]. La integración de los sistemas PLM y MES permite una colaboración eficaz entre los equipos de diseño y producción, mejorando la eficiencia y la calidad del proceso de fabricación.

2.3. Marco teórico

Los conceptos para el planteamiento del diseño generativo se desglosan bajo una estructura de detalle superior hasta abarcar en aspectos puntuales o propios del diseño, por tal motivo: (1) se comienza analizando los entornos de la industria manufacturera y como se plantean sus esquemas o modelos actuales, (2) seguido de una clasificación interna de cada industria, a la cual puede evidenciarse una aplicación de soporte a la etapa de diseño, (3) continuando con un desarrollo del ciclo de vida de producto, destacando como en la actualidad provee información que puede modificar la etapa de diseño y (4) finalizando con la aplicación de herramientas tecnológicas que han simplificado los procesos de creación de productos y procesos de mejora basados en optimización.

2.3.1. Modelos contemporáneos de automatización industrial

Algunos paradigmas actuales de manufactura se adaptan en respuesta a los planes de gobierno como los menciona Liao et al. [60], en este estudio se observa como los gobiernos buscan mejorar su productividad al desarrollar avances focalizados en diferentes tecnologías o paradigmas. En la actualidad conceptos como la “Industria 4.0”, “Smart Factories”, “Hecho en China 2025” y el “Internet Industrial de las Cosas” (IIoT) son un marco de referencia del desarrollo de Fabricas de Manufactura Flexible (FMS) que incluyen los avances en distintas tecnologías de la información y automatización.

Los conceptos de FMS utiliza modelos que proveen sistemas robustos ante variaciones en oferta-demanda, calidad, cantidad y complejidad en los productos. Aunque en los primeros

modelos de FMS solo se analizaba el proceso en la fábrica, en los nuevos paradigmas se integran otros sistemas como centros de investigación y universidades, tal y como recopila Veile et al. en la aplicación de la industria 4.0 en la actualidad [61], por tal motivo desarrollos hechos por Tosse et al. [48] en centros de investigación, como el “Laboratorio de fabrica experimental de la Universidad Nacional de Colombia (LabFabEX UN)”, son un referente en el país.

Referencia base en la creación de producto

La automatización industrial continua en el desarrollo de arquitecturas que permita englobar los nuevos paradigmas para el desarrollo y uso de productos, incluyendo conceptos de ciclo de vida de producto, esquemas de transición de información, productos como servicios, seguridad, integración de fábricas por objetos inteligentes, entre otros. Este trabajo resalta el modelo de arquitectura de referencia para la Industria 4.0 (RAMI4.0), creado con el esfuerzo del sector industrial alemán y que es de consulta pública por la empresa ZVEI [62], debido a la aplicación directa en el laboratorio LabFabEx-UN como muestra del paradigma de la industria 4.0.

El modelo RAMI4.0 tiene como principio la integración de distintas normas, generando una inclusión complementaria entre diferentes modelos y acoplándolos en un modelo tridimensional que responde a las tecnologías y visiones de las industrias modernas automatizadas. Este tipo de modelo lo ejemplifica en su tesis de maestría Cifuentes, A [63] y Cifuentes, P [64], donde la creación y seguimiento del ciclo de vida de producto es el enfoque último de aplicación como una muestra aplicada del LabFabEx-UN.

El modelo RAMI incluye el seguimiento del ciclo de vida del producto y del nivel jerárquico del ISA 95, por lo cual se observa que la integración de ambos es lo que representa la visión a futuro [44].

2.3.2. Influencia de procesos de creación de productos según su clasificación de manufactura

Debido a la naturaleza del trabajo de apoyar el soporte de creación de productos y que las políticas gubernamentales influyen en el tipo de industria que se quiere desarrollar en un país, a niveles locales se despliega el desarrollo de diversas clasificaciones de industria manufacturera y en las cuales se identifican una mayor influencia de la interacción con dicha etapa.

Cabe aclarar que según el desarrollo tecnológico de los países e industrias, estas barreras diferenciadoras de clasificación dejan de ser estrictas, y se utilizan con la finalidad de generalizar un concepto. Lo anterior se puede ejemplificar que mientras la producción de un

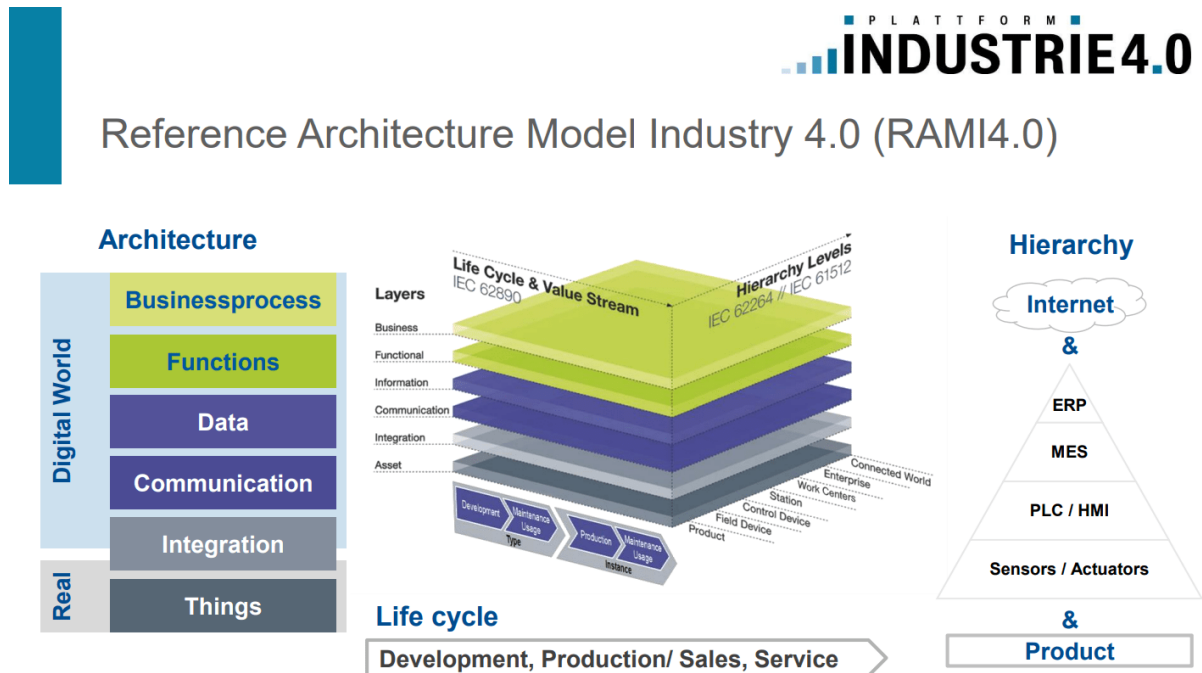


Figura 2-3.: Modelo de arquitectura de referencia para la Industria 4.0 (RAMI4.0).

Fuente: Martin Hankel, Digitising Manufacturing 2016 [2]

producto en un país se puede considerar de gran escala por los volúmenes, en otro país dichos volúmenes de producción dan una representación de pequeña escala.

Clasificación de la manufactura según sus operaciones

La finalidad de esta clasificación es definir las operaciones que pueden modificar la materia prima para convertirla en un producto. En esta clasificación identificamos la manufactura discreta y continua, destacando que la manufactura discreta se acopla mas al ambiente del laboratorio LabFabEx-UN.

- Manufactura continua:** Esta clasificación hace referencia a que la materia prima sufre operaciones que no son reversibles y que no se pueden detener en el proceso de fabricación, porque afectaría el resultado final del producto. Lo anterior también hace referencia a que sea llamado "Manufactura por procesos".
- Manufactura discreta:** Hace referencia a que utiliza diferentes productos para generar un nuevo producto, por lo cual se entiende la discretización del nombre. La operación principal de este tipo de manufactura es el ensamblaje, por lo cual este tipo de manufactura también es llamado "Manufactura por ensamblajes".

Como se puede apreciar un proceso de manufactura discreta generalmente trae la implicación de procesos de manufactura continua para generar cada uno de los componentes a utilizar

en el ensamblaje, por lo cual los requerimientos técnicos en la manufactura discreta son más detallados como lo menciona Mohanty et al.[65], en un estudio a la industria manufacturera en la India. Por las características cambiantes en el producto y en los procesos de este tipo de manufactura requiere un nivel elevado de conocimiento del producto y del proceso.

En el contexto de esta tesis de maestría se cuentan con operaciones de remoción de material y adición de material (manufactura continua) para fabricar las piezas discretas a utilizar en procesos de ensamblaje. El proceso de creación de producto sera discreto, pero también afectara los procesos de manufactura continua por su directa implicación.

Familia de productos

Heredada de la manufactura continua y aplicada junto al diseño paramétrico, busca simplificar los procesos de seteo de herramienta y dinamizar el ambiente de fabricación, al poder manufactura productos "discretos" con las mismas características pero en magnitudes diferentes. Una ejemplificación es que mientras un producto necesita tener una perforación, el valor de la magnitud de esta característica puede cambiar entre diferentes elementos pero en todos los elementos se encuentra presente, en este caso siendo el diámetro o la profundidad de la perforación las dos magnitudes principales de cambio.

Clasificación de la manufactura según su filosofía

Esta clasificación brinda una idea de como se adapta una fabrica con todos sus activos para responder al mercado. Esto tiene implicaciones directa en el producto final en parámetros como la cantidad, tiempos, calidad, entre otros. Esta clasificación esta ligada a la imagen corporativa que tienen, pero a diferencia de la misión o visión de una empresa, esta puede cambiar sin alterar el producto final para responder a distintos mercados.

- **Manufactura celular:** Diseño de planta que consta de lineas de producción o talleres (células) en las cuales en cada linea se manufactura una familia de productos. Cada célula da como resultado un producto y el comportamiento de cada célula implica la realización de distintos procesos, representado en un nivel de automatización mayor.
- **Manufactura flexible:** Su principal característica es que consta de sistema de transporte y control como interfaz entre maquinaria, esta interacción permite producir una gran cantidad de productos haciendo uso de cualquiera de los recursos disponibles, de ahí a que se considere flexible.
- **Manufactura esbelta:** Se basa en una idea de mejora continua en el cual buscan eliminar operaciones que no generan valor al producto, disminuir desperdicios y reducir costos de manufactura.

Desde el planteamiento de sistemas de manufactura flexible (FMS) hasta la actualidad se ha buscado obtener el mejor desempeño, por tal motivo todos estos modelos se basan en recopilar información y aplicar un proceso de optimización. La principal diferencia de los nuevos paradigmas respecto a los antiguos es la cantidad de información y el tiempo en el cual se obtiene. Dada la importancia de estos procesos de optimización Yang et al., 2019 [66], muestra una recopilación de distintos algoritmos basados en tiempo discreto y tiempo continuo, el cual se complementa en la sección 2.3.5.

Debido a su filosofía los FMS permiten ajustar una fabrica para responder a la creación de diferentes productos, por tal motivo el ambiente de diseño es mas critico que en los otros dos modelos. Estos sistemas tienen registro de estudio desde el año 1970 según el estado de arte realizado por Sethi et al., 1990 [67], en el cual aseguran que se originó debido a los avances de los microprocesadores en dicho periodo. En la actualidad estudios realizados por Saren et al., 2016 [68], muestra la vigencia de este modelo que busca la flexibilidad en los procesos de fabricación, pero ahora soportado sobre nuevas tecnologías y paradigmas.

Esta tesis mantiene la mentalidad de desarrollo bajo esquemas de manufactura flexible, pero puede ser extrapolado a los diferentes modelos planteados con responder a las métricas que definen a cada uno. Por lo tanto la identificación de las métricas es lo mas importante y para efectos propios del trabajo solo se analiza la manufactura flexible.

Métricas de flexibilidad

La evaluación de diseño de producto en métodos tradicionales, como los expuestos por Ullman y Norton, es realizada respecto a los requerimientos del producto proporcionados por el usuario. Esta evaluación puede ser generalizada al incluir la capacidad de fabricación del producto, pero no detalla como se comporta el producto frente al modelo de producción de la fábrica. Enfocado en la etapa de diseño en FMS se han propuesto métricas de flexibilidad de producto donde la complejidad, la interacción y la cantidad de piezas de un producto son las características principales. Métricas como las descritas por Kim et al., 2016 [69], Asikoglu et al., 2012 [70] y Lafou et al., 2016 [71] utilizan matrices para expresar las correlaciones y complejidad de todas las piezas de un diseño. Estas métricas permiten estimar las capacidades de respuesta de las fábricas y tomar decisiones. Kasie et al., 2017 [72], muestra cómo trabajan estos sistemas y se detalla cómo el uso de la información recopilada en la fábrica alimenta todas las etapas de manufactura.

Dada la acotación del trabajo de investigación en sistemas de manufactura flexible, se pretende evaluar la flexibilidad del diseño, en términos de producto y proceso. En el enfoque de producto se han desarrollado trabajos que expresan la interacción de todos los componentes de un producto entre ellos, esta interacción de componentes también se referencia bajo la temática de “arquitectura de producto”. Trabajos como los expuestos Kim et al. y Asikoglu

et al., describen una matriz cuadrada de tamaño n , donde n es el número de componentes del producto, en el cual describen estas interacciones bajo diferentes medios, por ejemplo, Kim et al. define cuatro medios “conexiones físicas”, “flujo de masa”, “flujo de energía” y “flujo de información”. Asikoglu et al. definen 6 medios en su trabajo.

La flexibilidad de producto se puede calcular según la interacción de componentes usando las matrices de “arquitectura de producto” y midiendo la influencia al agregar, reemplazar o eliminar componentes de un producto. La flexibilidad de proceso nos indica la flexibilidad que tiene una fábrica de manufacturar el producto, Lafou et al., desarrolla una metodología para identificar entre dos o más fábricas cual es la que ofrece mejor flexibilidad respecto al producto.

Estas métricas descritas proporcionan indicios de agrupación de componentes y comunicación, lo que permite plantear capacidad de realimentación por medio de la estructura de datos propuesta, donde la conectividad debe estar descrita y debe ser medible.

2.3.3. Ciclo de vida del producto

Como eje principal al diseño de producto en las metodologías tradicionales se usan conceptos como factor de seguridad, tiempo de uso, entre otros. Dichos conceptos se establecen como requerimientos funcionales para garantizar un tiempo de vida útil del producto, pero la evaluación al producto se establece en periodos más cortos y hay pocos protocolos establecidos para realizar el seguimiento cuando llega al usuario.

En la actualidad se puede hacer una clasificación muy simplificada de como ofertar el producto como un bien o un servicio. Cuando son ofertados como bien se considera un sistema lineal que solo se analiza en el entorno de producción. En el caso específico de productos que se ofrecen bajo el concepto de servicios, Igba et al., 2015 [73] asegura que es posible generar procesos de retroalimentación que conlleven a mejorarlo al obtener información durante todo su ciclo de vida. Lo anterior debido a que cuando se oferta como bien, una vez es adquirido por el individuo el producto deja de ser de la compañía, mientras en el caso de los servicios, el producto es usado mientras sigue siendo de la compañía.

Manufactura Lineal o Secuencial

Este concepto se basa en la creación de un producto mediante la limitación de sus características de uso para garantizar su cumplimiento con límites de tiempo establecidos, mediante estimaciones de seguridad. Estos sistemas realizan inspecciones del producto durante las etapas de manufactura en una fábrica, evaluando su conformidad con los requerimientos de diseño. La figura 2-4 presenta un esquema simplificado de este concepto de ciclo de vida lineal.

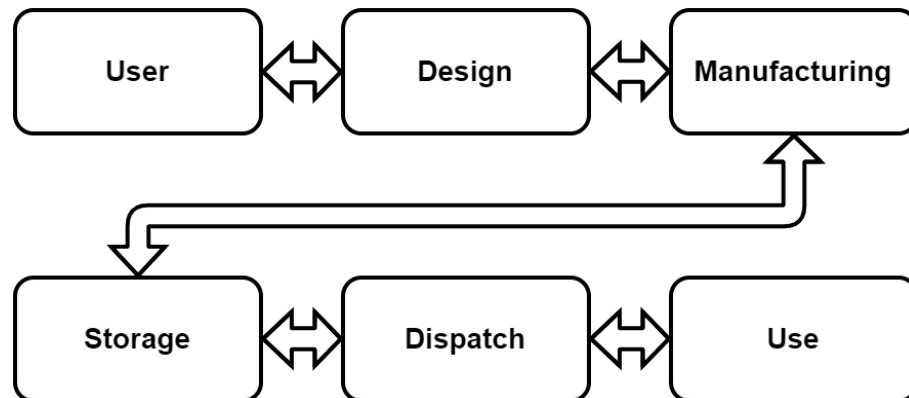


Figura 2-4.: Concepto lineal de ciclo de vida del producto.

Fuente: Creación Propia

Sin embargo, estas aproximaciones tienen ciertas limitaciones, ya que se centran en registros de manufactura y no de uso. En las modernas industrias de producción, los tiempos de manufactura son notablemente reducidos en comparación con el tiempo de vida estimado del producto. En la actualidad, los sistemas digitales o electrónicos son capaces de generar auto-diagnósticos, transmitir información por la red y solventar procesos de mejoramiento del producto a partir de la retroalimentación obtenida. No obstante, los sistemas mecánicos se enfrentan a la dificultad de que su diagnóstico se realiza únicamente cuando un elemento falla, lo cual no es una situación deseada para un diseño integral y de alta calidad.

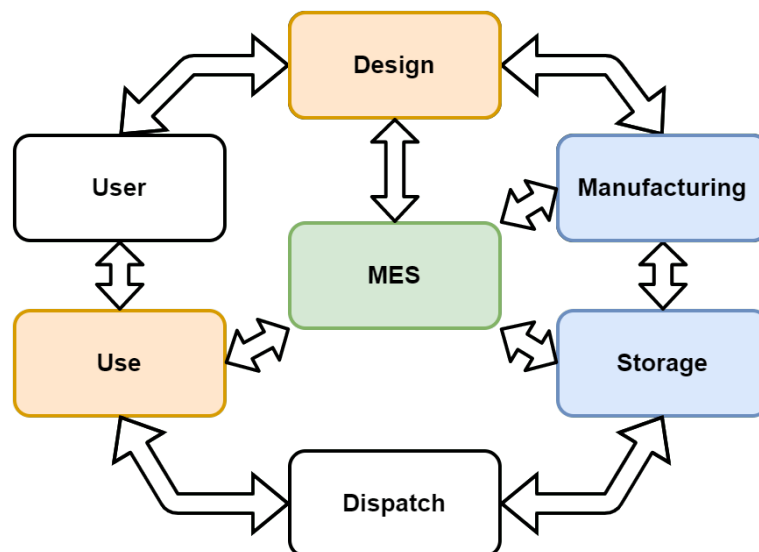


Figura 2-5.: Concepto concurrente aplicado en el LabFabEx-UN.

Fuente: Creación Propia.

Manufactura Concurrente

Los sistemas concurrentes dan la idea de cerrar el ciclo de vida de producto, en el cual se puede tener una realimentación de todas las etapas en cualquier momento. En este tipo de sistemas se destacan los software de ejecución de manufactura (MES) que permiten centralizar y transmitir la información del producto. Von Stietencron et al, [74] transmiten la idea de un manejo de ciclo de vida de producto en lazo cerrado (close loop -PLM) que muestra un sistema más detallado de las etapas que conlleva este ciclo de vida concurrente.

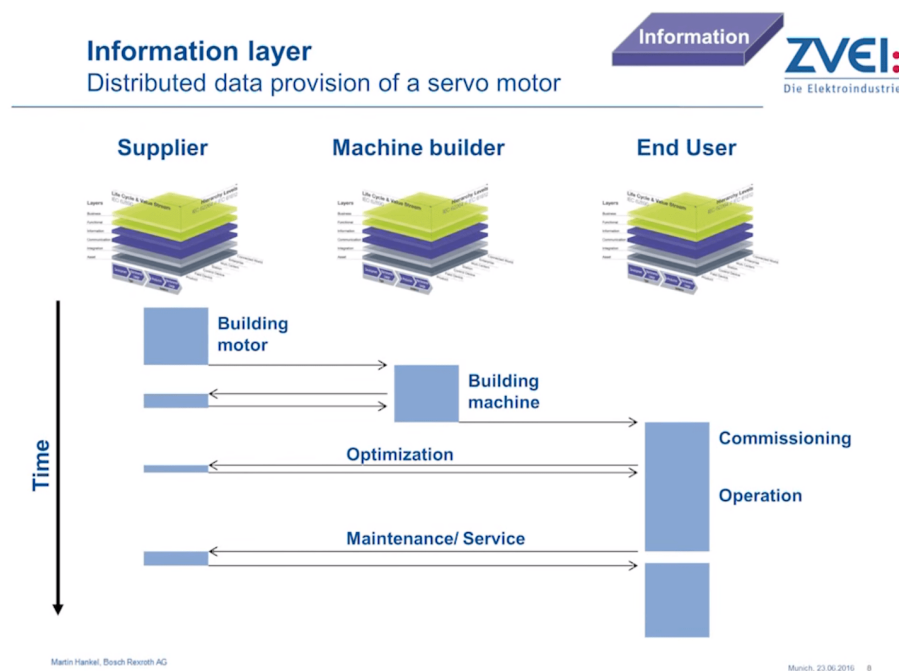


Figura 2-6.: Descripción del modelo RAMI4.0 en el tiempo.

Fuente: OPC Day Europe 2016 [3]

En la figura 2-5 se observa un esquema base para un ciclo de vida concurrente planteado en el LabFabEx-UN, el cual las etapas que actualmente interactúan con PRIA-UN son la etapa de manufactura y almacenamiento, mientras este trabajo busca aumentar la funcionalidad de los sistemas MES complementado las etapas de diseño y de uso, donde generalmente el alcance de estos software son en la etapa de manufactura y almacenamiento. Bajo la idea de poder aplicarlo en esquemas como el RAMI4.0 en la figura 2-6 se observa como se puede escalar el sistema simplificado al incluir la variable del tiempo en las capas de interacción. Esta imagen es tomada en una divulgación oficial del modelo RAMI4.0 realizado por Heidelberg en el evento OPC Day Europe 2016 [3], y aunque en este trabajo no se llega a desarrollar el modelo RAMI4.0, la proyección del laboratorio hace necesario considerar el escalamiento del sistema PRIA-UN para lograr este objetivo en el futuro.

2.3.4. Programación secuencial y paralela

En el marco de la ejecución de las estructuras de datos y la exploración basada en técnicas de optimización multiobjetivo, es crucial comprender el desarrollo de algoritmos mediante programación secuencial y programación paralela, ya que esto tiene un impacto significativo en el tiempo de resolución de un sistema.

- **Secuencial:** Este enfoque implica que una serie de tareas se ejecutan de manera individual, y cada tarea debe completarse antes de que pueda comenzar la siguiente. La sincronización en este método establece un orden definido para las actividades, y los resultados de una tarea pueden influir en la ejecución de la próxima tarea.
- **Paralela:** En cambio, en la programación paralela se define claramente cómo realizar una tarea y se aplica simultáneamente N veces para obtener N resultados en el mismo tiempo que tomaría realizar la tarea una vez en programación secuencial. Por lo tanto, el paralelismo se aplica a tareas pequeñas y bien definidas que son repetitivas.

En la actualidad, existen dos enfoques principales para aplicar el paralelismo basado en hardware. (1) Paralelismo basado en CPU: hace uso de la capacidad de cómputo de las CPU modernas, que cuentan con múltiples núcleos y la capacidad de manejar múltiples hilos de procesos por núcleo. Este enfoque es adecuado para tareas que requieren altos recursos de memoria y tiempos prolongados de ejecución. (2) Paralelismo basado en GPU: en este enfoque, la arquitectura de la GPU permite tener una mayor cantidad de núcleos y hilos de programación, pero a expensas de limitaciones en la memoria. Esto implica que, aunque puede ejecutar un gran número de procesos, estos deben ser relativamente pequeños y cortos en tiempo de ejecución.

Es importante destacar que, si bien son conceptos distintos, el paralelismo se emplea generalmente en combinación con la programación secuencial, aprovechando su capacidad para reducir el tiempo empleado en tareas repetitivas presentes en la programación secuencial.

En esta tesis de maestría se emplea el paralelismo mediante el uso de GPU con la API de CUDA (Compute Unified Device Architecture) de NVIDIA, lo que permite hacer uso de la memoria estática y dinámica en la GPU.

En la alternativa de memoria estática, se deben establecer valores límite para evitar desbordamientos de información, lo que implica reservar una gran cantidad de espacio en memoria para cada copia de la estructura que no esté en uso. En un sistema de desarrollo de optimización, como se presenta en la sección 2.3.5, es necesario generar una población, lo que resulta en una duplicación de información de la estructura de datos. Esto presenta limitaciones de memoria y, como consecuencia, se deben establecer límites para la población a evaluar.

En la alternativa de memoria dinámica, las estructuras comienzan con una asignación mínima de espacio y se ajustan a las variaciones a medida que se desarrolla el sistema generativo, para responder a la exploración y explotación de la evaluación multiobjetivo. Sin embargo, surge un problema debido a que la asignación de memoria en la GPU no se realiza directamente, sino que debe realizarse en la CPU. Desafortunadamente, la API de CUDA en su versión 11.8.0 aún no ha implementado esta característica. Como resultado, al realizar la asignación de memoria directamente en la GPU, no es posible copiarla a la CPU sin haberla definido previamente.

Se han realizado pruebas donde la asignación de memoria dinámica se hace en la GPU y se desarrolla un sistema. Para extraer las soluciones, estas se traen individualmente en una estructura de memoria fija. Se observa que el desarrollo en GPU es posible, pero surge la necesidad de crear estructuras de memoria fija para copiar los resultados de la GPU a la CPU. Esto lleva a establecer el desarrollo de estructuras de manera híbrida, combinando las ventajas de ambas alternativas de memoria.

2.3.5. Optimización aplicada en la creación de producto

En el campo del diseño de productos, la optimización desempeña un papel crucial para mejorar diversas características y cumplir con los requisitos del producto. La optimización puede ser aplicada en cualquier etapa del ciclo de vida del producto y se puede clasificar en dos categorías: mono-objetivo y multiobjetivo.

- **Optimización mono-objetivo:** Es un enfoque de optimización que se centra en resolver o mejorar una única propiedad o objetivo del producto. En este caso, se busca encontrar una solución óptima que optimice esa única propiedad. El resultado de la optimización mono-objetivo es un único punto en el espacio de soluciones.
- **Optimización multiobjetivo:** Es un enfoque de optimización que busca encontrar un equilibrio entre múltiples objetivos o características a mejorar del producto. En este caso, se consideran múltiples criterios de optimización que pueden entrar en conflicto entre sí. El resultado de la optimización multiobjetivo es un conjunto de soluciones no dominadas, conocido como el frente de Pareto. Estas soluciones representan diferentes compromisos entre los objetivos, donde ninguna solución del conjunto puede mejorar un objetivo sin empeorar al menos uno de los otros objetivos.

Un objetivo de optimización se define como resolver una función para encontrar el valor mínimo (o máximo). Ejemplos comunes de objetivos en la optimización del diseño de productos pueden ser minimizar el peso, aumentar la rigidez, reducir los tiempos o disminuir los costos, entre otros. Estos ejemplos ilustran los conceptos a los que se enfrentan los diseñadores al generar conceptos y tratar de satisfacer los requisitos funcionales del producto.

Un gran número de técnicas multiobjetivo son generacionales o por iteraciones como se observan en las revisiones de la literatura de Coello et al. [75] y Ishibuchi et al. [76]. Además se identifican en un gran número de estas que abarcan dos etapas una de exploración y otra de explotación. Tomando como guía la recopilación de algoritmos para optimización multiobjetivo realizada por Chiandussi et al. [77] se desarrolla en detalle el concepto de Frente de Pareto que permite clasificar las soluciones obtenidas.

Optimización Multiobjetivo y el Frente de Pareto

En la optimización multiobjetivo, se busca encontrar soluciones que representen un equilibrio entre diferentes objetivos, lo cual puede implicar compromisos y trade-offs. Una herramienta clave en este contexto es el frente de Pareto, que proporciona un conjunto de soluciones no dominadas en el espacio de los objetivos.

El frente de Pareto se define como el conjunto de soluciones para las cuales no es posible mejorar un objetivo sin empeorar al menos otro objetivo. En otras palabras, ninguna solución en el frente de Pareto puede ser considerada estrictamente mejor que otra en todos los objetivos. Esto se conoce como la relación de no dominancia.

Formalmente, sea un problema de optimización con m objetivos a ser minimizados, y sea x una solución factible con m valores objetivo denotados como $f_1(x), f_2(x), \dots, f_m(x)$. Se dice que una solución x domina a otra solución y si y solo si $f_i(x) \leq f_i(y)$ para todo $i = 1, 2, \dots, m$ y existe al menos un j tal que $f_j(x) < f_j(y)$. El frente de Pareto está compuesto por todas las soluciones que no son dominadas por ninguna otra solución.

En la optimización multiobjetivo, el objetivo es encontrar soluciones que se encuentren lo más cerca posible del frente de Pareto, para así ofrecer un conjunto de opciones óptimas que representen diferentes compensaciones entre los objetivos. Este enfoque permite al diseñador explorar diferentes soluciones y tomar decisiones informadas en función de sus preferencias y restricciones.

Los frentes de Pareto pueden desarrollarse de manera óptima por programación secuencial en problemas de dos o tres objetivos, cuando el número de objetivos incrementa se pueden obtener bajo tres conceptos de programación paralela como lo desarrolla Ortega et al. [78] en la comparación en rendimiento de plantear el ordenamiento de frentes de Pareto: 1) usando programación paralela según los hilos de trabajo de la CPU. 2) planteando la GPU por medio de CUDA. 3) un método híbrido entre GPU y los hilos de la CPU.

Los resultados de Ortega indican que usar la GPU resulta en una mejora sustancial en el desarrollo de dicha clasificación y basado en el planteamiento de esta tesis basado en esquemas multiobjetivos es la opción seleccionada para dicho ordenamiento. Aclarando que bajo el esquema de implementación de este trabajo todos los objetivos tendrán el mismo peso

al momento de obtener los frentes de Pareto, la implementación realizada para este trabajo se puede observar en el anexo A. Adicionalmente, el desarrollo de algoritmos como el NSGA-II que hacen uso de esta clasificación también se ve reflejada las mejoras sustanciales como lo menciona De Oliveira et al. [79] en el desarrollo de NSGA-II sobre GPU para responder a un problema multiobjetivo de una planta de poder hidroeléctrica.

Técnicas metaheurísticas

Las técnicas metaheurísticas son una opción adecuada para abordar problemas de optimización en el diseño de productos en un entorno multiobjetivo, ya que se basan en planteamientos abstractos y eficientes para resolver problemas complejos. Estas técnicas permiten la transformación de valores en el espacio de trabajo en valores reales, lo que las hace generalizables y aplicables a una amplia gama de problemas.

El objetivo de esta investigación es desarrollar una metodología basada en técnicas metaheurísticas para la optimización en el diseño generativo de productos. Se hará especial énfasis en el uso de técnicas evolutivas, siendo el algoritmo NSGA-II (Nondominated Sorting Genetic Algorithm II) la técnica principal que se empleará en los algoritmos desarrollados en esta tesis de maestría.[80]

El algoritmo NSGA-II se destaca como una de las técnicas más populares en la optimización multiobjetivo. Se basa en una estructura generacional que se divide en dos etapas: exploración y explotación. Utilizando el concepto de dominancia de Pareto, NSGA-II es capaz de generar un conjunto de soluciones no dominadas que representan diferentes compromisos entre los objetivos del diseño. Esto proporciona al diseñador una variedad de opciones óptimas en función de las necesidades y preferencias específicas.

La elección de NSGA-II como la técnica principal en esta tesis de maestría se justifica por su capacidad para abordar problemas de optimización multiobjetivo y generar un conjunto diverso de soluciones no dominadas. La utilización de esta técnica permitirá explorar y evaluar diferentes opciones de diseño, lo que facilitará la toma de decisiones informadas durante el proceso de diseño generativo.

En el proceso de selección también se destacan los siguientes técnicas con enfoques populares y ampliamente utilizados en la comunidad científica para resolver problemas de optimización multiobjetivo. Cada una de ellas tiene características distintivas:

- **SPEA2:** Es una mejora del algoritmo Strength Pareto Evolutionary Algorithm (SPEA). Se basa en la idea de evaluar y clasificar las soluciones en función de su dominancia de Pareto y su densidad en el espacio objetivo. SPEA2 ofrece un equilibrio entre exploración y explotación al seleccionar las mejores soluciones y mantener una diversidad de soluciones no dominadas.[81]

- TV-MOPSO: Es una variante del algoritmo Particle Swarm Optimization (PSO) que se adapta a problemas multiobjetivo con cambios en el tiempo. Utiliza un enfoque basado en partículas que se mueven en el espacio de búsqueda y se actualizan mediante una combinación de su mejor posición personal y la mejor posición global. TV-MOPSO permite explorar el espacio de soluciones y encontrar soluciones óptimas en diferentes momentos del tiempo.[82]
- NSPSO: Es otra variante del algoritmo Particle Swarm Optimization (PSO) diseñada específicamente para problemas de optimización multiobjetivo. Incorpora el concepto de dominancia de Pareto y utiliza una clasificación de no dominancia para seleccionar y actualizar las partículas. NSPSO ofrece una búsqueda eficiente de soluciones no dominadas al utilizar una estrategia basada en enjambres de partículas.[83]

Estas técnicas de optimización multiobjetivo, como el SPEA2, TV-MOPSO y NSPSO, comparten conceptos similares como el frente de Pareto, generaciones y poblaciones. Sin embargo, es importante destacar que en esta tesis de maestría no se busca obtener soluciones específicas utilizando estas técnicas, sino que se utiliza la literatura existente sobre ellas para seleccionar la más adecuada. La selección se basa en encontrar una técnica que cumpla con los requisitos de exploración y explotación en entornos multiobjetivos, aprovechando las propiedades del frente de Pareto. De esta manera, se busca aplicar una técnica metaheurística eficiente y efectiva en la exploración de soluciones óptimas en el diseño generativo de productos.

2.4. Hipótesis

La implementación de una estructura de datos basada en grafos que utilice un modelado matemático para transmitir valores de propiedades físicas y se integre en la etapa de diseño conceptual y en detalle, permitirá mejorar la eficiencia y efectividad del proceso de diseño de componentes, reducir el tiempo y costo de producción, y aumentar la calidad del producto final. Además, el uso de cámaras web para seguimiento de producto y la transmisión estructurada de información mediante protocolos HTTP permitirán una disminución de la sensorica invasiva en los componentes, facilitando el proceso de producción.

La implementación de algoritmos genéticos, redes neuronales e inteligencia artificial como soporte para manejar la información estructurada en grandes cantidades y generar conocimiento para la creación de nuevos productos, en combinación con la estructura de datos basada en grafos, permitirá almacenar y operar información estructurada y conectarla de manera flexible. Se espera que la implementación de esta propuesta tenga un impacto significativo en la industria manufacturera, al ofrecer una alternativa eficiente y efectiva para el diseño de componentes y la producción de nuevos productos.

2.5. Metodología

En esta tesis de maestría, se llevó a cabo un diseño de investigación en el entorno de una fábrica automatizada llamada LabFabEx-UN, que utiliza el sistema de software PRIA-UN para controlar la maquinaria del laboratorio. Este entorno incluye maquinaria de manufactura, como una impresora 3D FDM para la adición de material y un torno suizo para la remoción de material. Además, se integran sistemas robóticos, como un brazo robótico antropomórfico y un robot tipo SCARA, y se utilizan vehículos autónomos de conducción propia (SDV) para el transporte de carga en el laboratorio. Todos estos componentes están gestionados por la arquitectura de software de ejecución de tareas PRIA-UN y pueden interactuar entre sí en procesos de manufactura.

Para aplicar los algoritmos de diseño generativo seleccionados, de tipo evolutivo y complementados con información simulada para obtener retroalimentación, se desarrolló una estructura de datos que permite el modelado automático de los componentes y la generación de variaciones de parámetros para crear un sistema. Esta estructura de datos se detalla en el siguiente capítulo, enfocándose en el modelado de un componente y su posterior aplicación en ensamblajes.

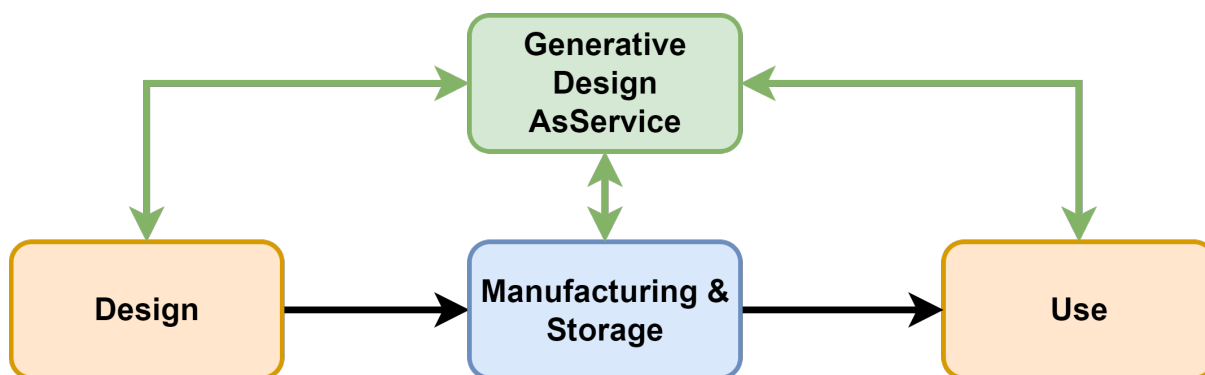


Figura 2-7.: Esquema de la influencia del diseño generativo en el ciclo de vida del producto.
Fuente: Creación Propia

La metodología utilizada abarca tres etapas del ciclo de vida del producto: diseño, manufactura y uso, como se muestra en la figura 2-7. Se propone el enfoque del diseño generativo como un servicio independiente que puede complementar los softwares MES existentes. Esta aproximación permite generar información estructurada según las necesidades de cada etapa, almacenando diferentes tipos de datos en cada estructura y, en algunos casos, delegando al diseñador el proceso de transmitir la información obtenida para las etapas siguientes del ciclo de vida. En este sentido, el diseñador selecciona y modifica las soluciones desarrolladas, otorgándole un papel activo en la iteración del proceso. Además, el enfoque de diseño

generativo como servicio ofrece la ventaja de utilizar recursos de computación secuencial y paralela, lo que permite aprovechar recursos locales con capacidades instaladas.

En la etapa de diseño, se desarrollaron dos enfoques de aplicación: uno basado en archivos CAD para la representación de contornos y otro que utiliza representaciones CAD por nubes de puntos. Posteriormente, en la etapa de manufactura, se empleó la información obtenida durante el diseño, incluyendo los modelos CAD, para evaluar diversos procesos de manufactura disponibles en el entorno automatizado. Se proporcionaron alternativas de manufactura que permitieron la ejecución semiautomática de dichos procesos, enriqueciendo así el entorno de diseño generativo realimentado.

En la etapa de uso, se diseñó una arquitectura que facilita la implementación eficiente de nueva maquinaria utilizando tecnología Docker en servidores de bajo costo, como los Raspberry Pi 4. Estos servidores se integraron en la arquitectura existente y se utilizaron para aplicar técnicas de visión artificial, convirtiendo los objetos del entorno de fabricación en objetos inteligentes. Esta arquitectura complementa la etapa de uso y permite un análisis más detallado de los procesos.

Además de la aplicación generalizada del diseño generativo realimentado en cualquier entorno de manufactura, se llevó a cabo un caso de estudio específico en el desarrollo de un Gripper para operaciones de tomar y colocar (Pick and Place) en el laboratorio. En el capítulo correspondiente al caso de estudio, se presenta el análisis y desarrollo práctico del Gripper, junto con la evaluación de diferentes soluciones.

La metodología utilizada en esta tesis de maestría se basó en un diseño de investigación en un entorno de fábrica automatizada, aplicando algoritmos de diseño generativo realimentado. Se seleccionó una muestra de participantes y se utilizaron instrumentos de medición apropiados para llevar a cabo procedimientos específicos. Los datos recopilados fueron analizados minuciosamente, proporcionando valiosos conocimientos sobre la aplicabilidad del diseño generativo realimentado en el entorno de fábrica automatizada, así como mejoras significativas en los procesos de manufactura.

3. Modelación en el Diseño Generativo Realimentado

El modelamiento de componentes en mecánica desempeña un papel fundamental en la representación y descripción de los componentes físicos a través de modelos matemáticos y/o geométricos. Es a través de este proceso que se logra capturar y representar de manera virtual la geometría, propiedades físicas y comportamiento mecánico de un componente.

Para lograr un adecuado modelamiento de un componente, es necesario seleccionar el enfoque más apropiado, ya sea mediante el uso de modelos analíticos, modelos numéricos o modelos basados en simulación por computadora. Esto implica definir los parámetros y variables relevantes, establecer las ecuaciones que describen el comportamiento del componente, y aplicar métodos de análisis y simulación para obtener resultados y conclusiones significativas.

En el presente capítulo, se aborda el modelamiento de componentes bajo el concepto de grafos direccionales, los cuales permiten el flujo de información y la interacción entre los nodos enlazados. La estructura de datos propuesta busca proveer información descriptiva del producto, la cual será manipulada por el propio sistema para llevar a cabo una exploración del espacio de diseño y responder a las necesidades del problema planteado.

Es importante destacar que los procesos de optimización no se limitan únicamente a las etapas finales del mejoramiento del producto, sino que se establecen procesos de medición y evaluación desde la concepción del problema. Sin embargo, esto solo es posible si el modelamiento describe de manera precisa las características del componente en cuestión.

Bajo la amplia definición del diseño generativo, se busca modelar un componente de manera que su solución represente propiedades físicas específicas, como su ubicación en el espacio o las fuerzas que actúan sobre él. En este sentido, el diseño de un componente, basado en las estructuras propuestas, puede describirse computacionalmente por el diseñador o incluso originarse a través de procesos de mutación automáticos, los cuales se detallarán en este capítulo.

La resolución de los requerimientos de un sistema implica el mapeo de dichos requerimientos a objetivos que deben ser alcanzados. Aunque este enfoque está fundamentado en procesos de optimización, es importante destacar que el objetivo principal no es obtener el producto

óptimo a nivel global, sino utilizar las características de optimización para explorar diferentes espacios de trabajo y seleccionar algunos mínimos locales que representen posibles soluciones al problema planteado. El proceso de selección y aplicación del diseño generativo en las etapas de manufactura se abordará en detalle en el capítulo 4.

Este planteamiento nos brinda la perspectiva de alcanzar la automatización del proceso de diseño, al destacar la posibilidad de realizar operaciones que generen cambios en el producto sin la intervención directa del ser humano ni la necesidad de una experticia humana para configurar y establecer las características a optimizar. Para lograr este objetivo, se establecerán diferentes reglas de diseño en las estructuras de datos, permitiendo generar una población y establecer procesos de mutación que generen variabilidad en los individuos.

Además, el modelamiento se plantea como una forma rápida de imitación y generación a gran escala de conceptos, sin depender de un flujo de manufactura masiva característico de países como China o Corea del Sur. Esto abre las puertas para que países en desarrollo puedan generar conocimiento en esta etapa de desarrollo y continuar con los procesos de innovación.

El capítulo se estructura de la siguiente manera: en primer lugar, se definen dos tipos de estructuras, una basada en el modelo matemático y otra basada en primitivas. Luego, se explora la realimentación de componentes ya fabricados bajo estas estructuras y se detalla la representación que conllevan. A continuación, se define cómo los objetivos pueden ser operados en estas estructuras, seguido por una explicación de las operaciones de mutación que se llevan a cabo en las estructuras de datos. Finalmente, se presentarán las conclusiones derivadas del desarrollo de estas estructuras.

3.1. Estructura de datos

La estructura de datos base en este trabajo consiste en el planteamiento de un grafo direccional basado en el trabajo de Sedgewick et al. [84] en su libro “Algorithms”. En este enfoque, se define la interacción de los componentes o nodos del grafo que describen la descomposición de un producto. Este concepto de grafo direccional permite establecer conexiones no cíclicas para evitar entrar en bucles infinitos en la interacción de los nodos. El direccionamiento del grafo se utiliza para transmitir la información heredada de un componente previo como una interacción entre componentes. En el desarrollo de este trabajo, se nombrará a estos nodos que describen una entidad con varias propiedades como Objeto Complejo de Unidad Menor (Minor Unit Complex Object - MUCO).

Con el objetivo de poder transmitir la información que describe un producto, se plantean dos estructuras de datos basadas en grafos: el modelamiento matemático y las primitivas. En la primera estructura, se desarrolla un modelamiento matemático del componente, el cual se obtiene a partir de un proceso de diseño previo. En este enfoque, se identifican

las características que describen al componente y las variables necesarias para obtener una respuesta del modelo. Por otro lado, la segunda estructura se basa en principios de primitivas, como se define en la Sección 3.1.2. Esta aproximación está diseñada para sistemas discretos, en los cuales el modelamiento se desarrolla de manera secuencial, sin embargo, este enfoque requiere capacidades computacionales más avanzadas en términos de operaciones.

Se ha realizado un análisis de estas estructuras considerando el caso de 2 dimensiones, con el propósito de demostrar su aplicabilidad. No obstante, se plantea como trabajo futuro la ampliación de estas estructuras a un espacio tridimensional.

3.1.1. Basado en modelamiento matemático

La estructura de datos se concibe con la finalidad de heredar información de conceptos previamente diseñados en el modelamiento matemático del componente. Esta estructura se basa en un grafo direccional en el cual los nodos representan los componentes o unidades menores del producto y las aristas representan la interacción entre estos componentes.

En esta estructura, cada nodo representa un Objeto Complejo de Unidad Menor (Minor Unit Complex Object, MUCO) y contiene información sobre el modelamiento matemático del componente correspondiente. Cada MUCO se describe mediante un conjunto de ecuaciones matemáticas que representan el comportamiento y las relaciones del componente. Estas ecuaciones pueden ser obtenidas a través de métodos como la linealización o ecuaciones diferenciales, dependiendo de la naturaleza del componente y del conocimiento del diseñador.

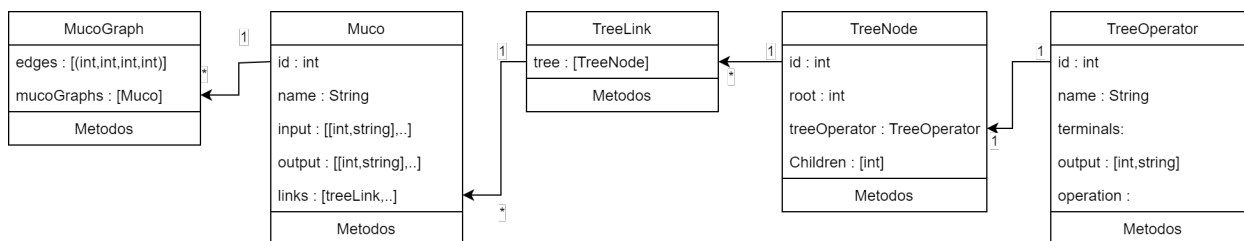


Figura 3-1.: Interacción de las clases basado en el modelo.

Fuente: Creación Propia

En términos de estructura, se establece una relación de cascada y de muchos a uno entre los componentes al cual se observa en la figura 3-1. Esto significa que la descripción de un componente representado como MucoGraph implica varios nodos de tipo MUCO, que al igual interactúa con otros componentes o MucoGraph por medio de las salidas de sus nodos MUCO. Esta estructura en cascada permite la descomposición del producto en componentes y la transmisión de información entre ellos para generar el modelamiento general. La concepción del modelo matemático que define las salidas de un nodo de tipo MUCO se produce por medio de arboles que es otra estructura de datos.

Objeto Complejo de Unidad Menor [MUCO]

En esta estructura se define el nodo del grafo como la unidad menor (MUCO), en el cual se aborda como un diagrama de caja negra con entradas y salidas, donde para cada salida corresponde una ecuación que toma valores de las entradas para resolverla. Para representar las ecuaciones en cada nodo MUCO se define un árbol como estructura de datos que permite almacenar y operar ecuaciones como se aprecia en la figura 3-2. Se destaca en la estructura por arboles la capacidad de establecer una única salida al tener ciertas entradas en la definición del modelo.

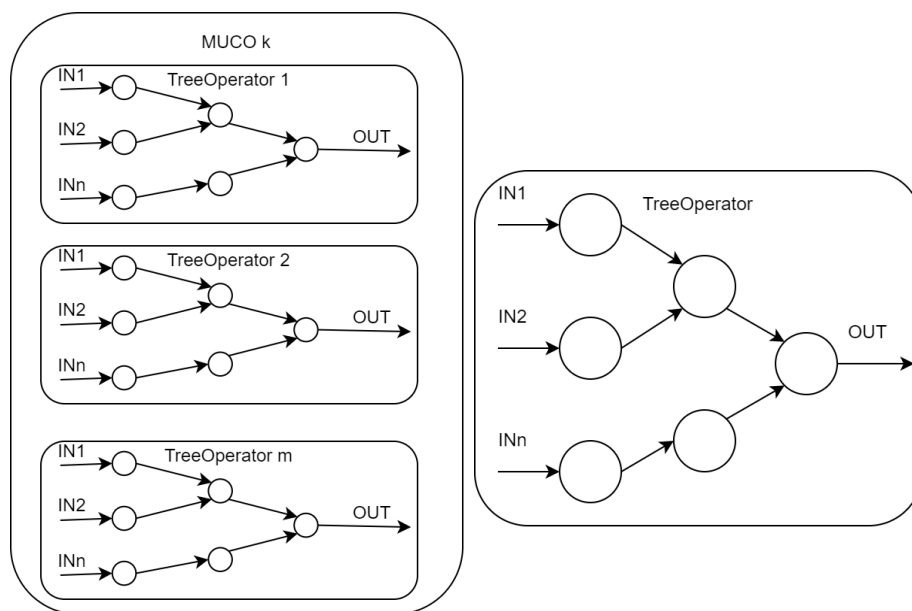


Figura 3-2.: Definición de MUCO basado en un arreglo de árboles.

Fuente: Creación Propia

La razón para definirlo mediante árboles es que estos se utilizan para definir, almacenar y resolver ecuaciones, al identificar completamente los valores numéricos y separarlos de los operandos. Por lo tanto, las entradas del árbol corresponden a valores numéricos que se operan según la operación definida en cada nodo, y dan como resultado un único valor final. Esta flexibilidad permite describir un componente modelado matemáticamente bajo esta estructura, a la cual se pueden aplicar constantes como una operación de suma, para ajustar modelos que no son precisos. Sin embargo, debido a su complejidad y baja efectividad, en esta tesis de maestría no se explora la manipulación de los árboles para ajustar automáticamente el modelado por árboles.

Los nodos MUCO se retroalimentan considerando un modelo geométrico y de fuerzas previamente conocido, lo que permite desarrollar el sistema. Por esta razón, se establece que

cada MUCO puede tener una cantidad de entradas y salidas diferentes, y tendrá M árboles que definirán las M salidas del MUCO.

Se observa que la inclusión de una estructura de datos dentro de otra, como es el caso de los árboles que permiten ajustarse a diferentes ecuaciones, requiere definir tres clases complementarias para el sistema, como se muestra en la figura 3-1. Sin embargo, dado que el proceso de resolución de árboles es secuencial, cada operación definida en el árbol genera un ciclo de ejecución en comparación con la incorporación directa de la ecuación en el código que lo ejecuta siempre en un ciclo de ejecución. El integrar la ecuación directamente en el código implica mayores operaciones manuales en el código fuente para la ejecución y que el almacenamiento de ecuaciones en bases de datos no sea escalable, a diferencia del uso de árboles que permite almacenar en bases de datos y generar interfaces de desarrollo de ecuaciones para los diseñadores.

En conclusión, aunque los árboles ayudan a generar una interfaz con el diseñador para expresar su conocimiento de modelamiento matemático mediante ecuaciones, esta estructura puede omitirse para lograr un desarrollo computacional más eficiente al incluir directamente la ecuación en el programa, a costa de un mayor esfuerzo en el desarrollo del código y una falta de escalabilidad en el almacenamiento de ecuaciones en bases de datos.

Grafo direccional

El modelo planteado para la interacción y resolución de los grafos direccionales se muestra en la figura 3-3. En este modelo se definen tres tipos de nodos MUCO: 1) terminales, 2) grafos y 3) salidas. Los grafos en color verde implican que todas sus entradas deben tener al menos una conexión a otro MUCO. Si alguna entrada queda desconectada, el sistema no se puede resolver. Esta es la razón principal de la definición de los nodos MUCO terminales y salidas, que complementan como valores de ajuste en entrada y salida del modelo.

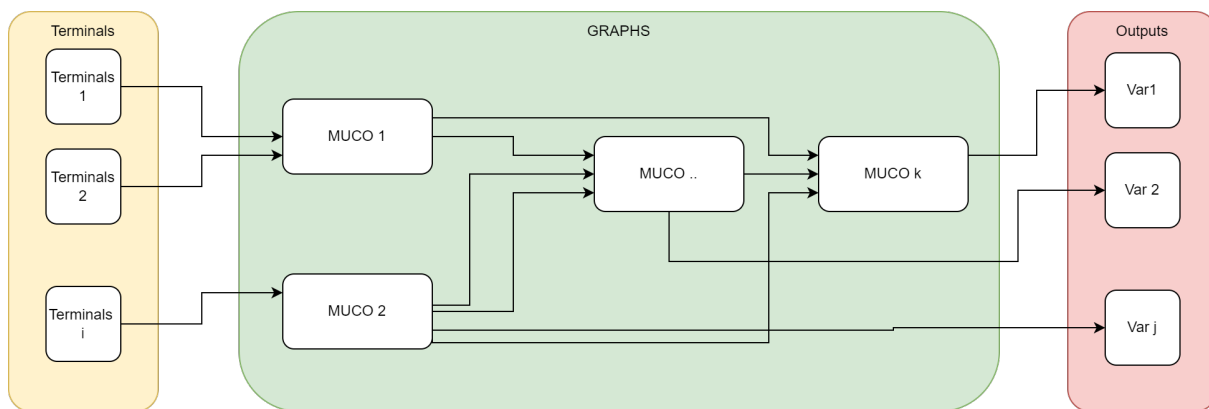


Figura 3-3.: Esquema de la estructura de un grafo direccional basado en el modelo.
Fuente: Creación Propia

Los terminales son un tipo de MUCO que no contienen operaciones y representan únicamente valores numéricos. Estos valores corresponden a valores iniciales del sistema y valores controlados por el usuario, que causan modificaciones en el sistema. Las salidas son otro tipo de MUCO que corresponde a las variables de seguimiento del sistema. Por lo tanto, el MUCO de salida no contiene conexiones de salida y solo tiene una conexión de entrada única.

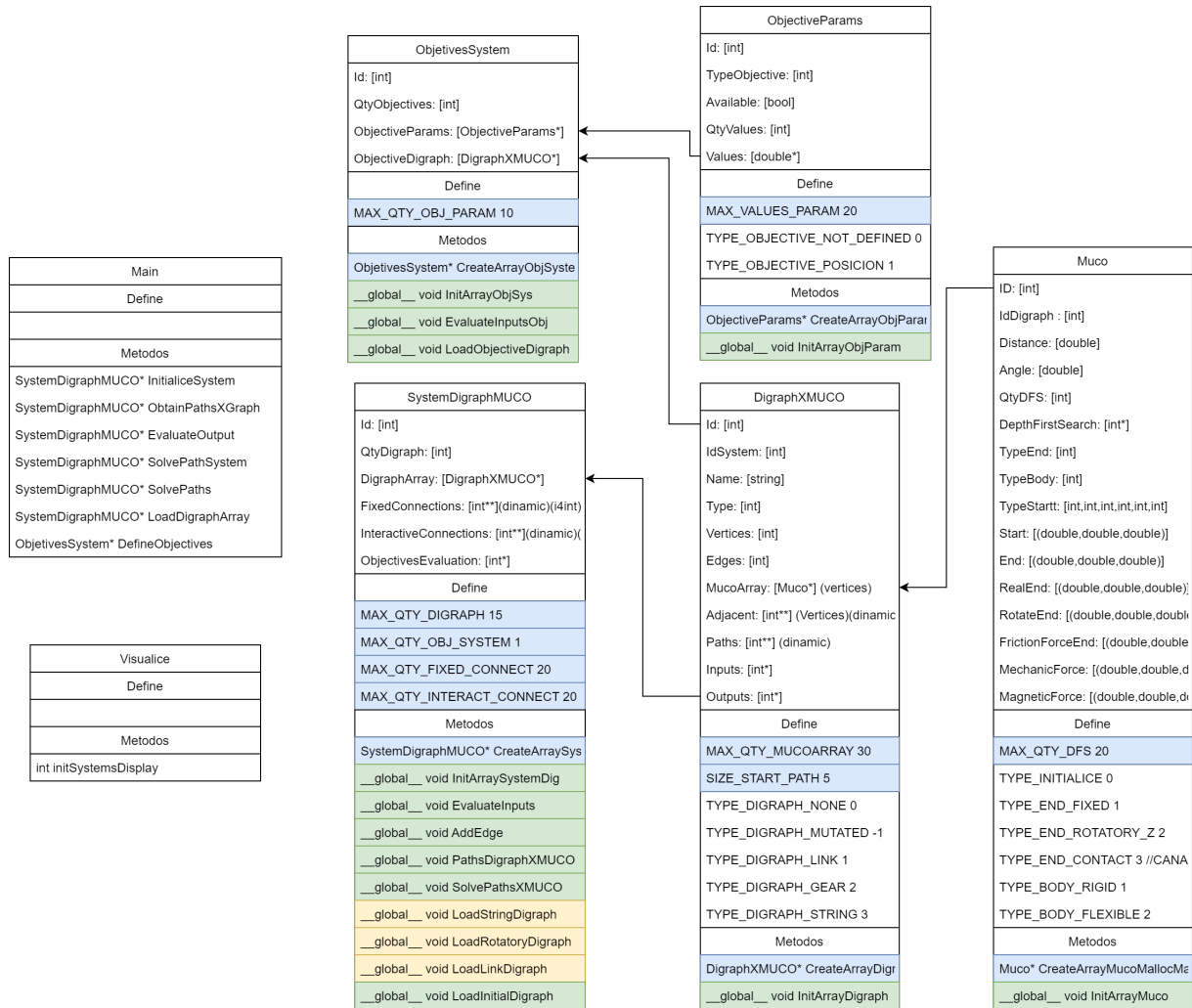


Figura 3-4.: Interacción de las clases basado en primitivas.

Fuente: Creación Propia

3.1.2. Basado en primitivas

La estructura de datos basada en primitivas toma su significado en base a la definición obtenida de la real academia española RAE: “Primero en su línea, o que no tiene ni toma origen de otra cosa”. Se expresa al definir una estructura mínima que será los nodos MUCO,

en la cual las combinaciones que se generen por la interacción de varias de estas con ellas mismas generan el modelo de un componente. En esta sección, se presenta la visualización de la interacción de las clases en la figura 3-4.

En la Figura 3-4 se observa una estructura en cascada, donde el MUCO representa la clase del último nivel de izquierda a derecha. Además, se indica que el grafo direccional (DigraphXMUCO) contiene varios nodos MUCO y tiene un nivel jerárquico mayor basado en sistemas (SystemDigraphMUCO). La relación con los objetivos se detallará en la sección 3.3, mientras que los conceptos de computación secuencial y paralela se observan aplicados en los métodos de cada entidad, en donde el color azul se refiere a métodos de programación secuenciales y el color verde a métodos de programación paralela.

Objeto Complejo de Unidad Menor [MUCO]

El MUCO es la entidad encargada de almacenar la interacción entre dos primitivas de descripción espacial, específicamente la interacción entre dos puntos. Esta clase almacena información como la distancia entre los puntos, el ángulo respecto a los ejes de referencia y la dirección y magnitud de las fuerzas expresadas en el punto de salida. La figura 3-5 ilustra el punto inicial y el punto final como los elementos que definen las primitivas en dos dimensiones para esta estructura.

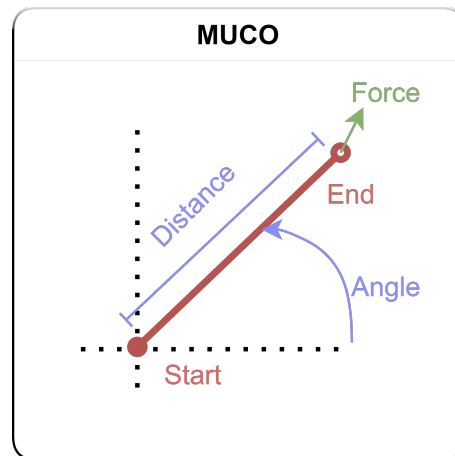


Figura 3-5.: Definición de MUCO basado en primitivas.

Fuente: Creación Propia

Resolver este tipo de MUCO implica obtener las propiedades en el punto final. Basándose en los parámetros almacenados en la estructura, se aplican las operaciones fijas descritas en la sección 3.3. Es importante destacar que, aunque no es el objetivo de este trabajo, el planteamiento del MUCO busca resolver la interacción de fuerzas mecánicas o magnéticas en el punto final con respecto al punto inicial. Esto establece una conexión entre las propiedades

de MUCO observables en la estructura de la Figura 3-4 y la representación espacial de la figura 3-5.

Grafo direccional

En esta estructura de datos basada en primitivas, se aplica el planteamiento de grafos direccionales propuesto, como se observa en la figura 3-6. Este grafo presenta conexiones simples que actualizan los valores del punto inicial de las características que lo definen descritas previamente en el MUCO basándose en los valores del punto final de otro MUCO. Esto logra una estandarización en la interacción de los nodos MUCO al operar bajo las mismas propiedades.

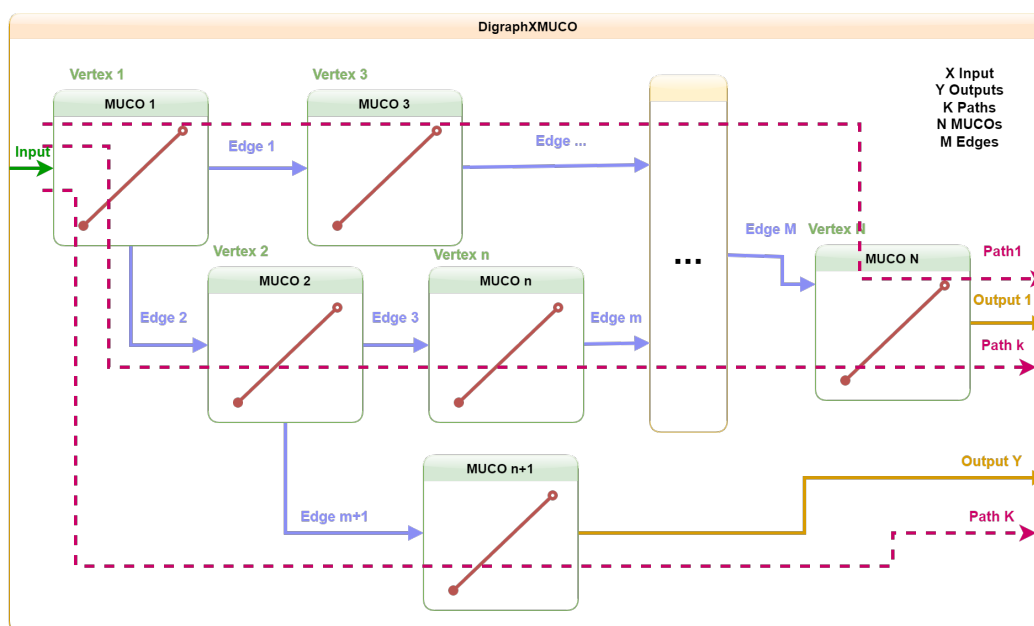


Figura 3-6.: Esquema de la estructura de un grafo direccional basado en primitivas.

Fuente: Creación Propia

Una particularidad de esta estructura es que las propiedades de distancia y ángulo en cada MUCO son únicas y no se modifican por el MUCO anterior. Sin embargo, estas propiedades pueden modificarse bajo dos condiciones:

- En procesos de exploración del espacio utilizando técnicas de optimización.
- En respuesta al MUCO del grafo direccional que recibe como entrada la salida de otro grafo direccional.

Esto indica que un grafo direccional representa un componente, y cuando se busca resolver un producto, se dice que es la interacción de varios grafos direccionales (es decir, varios componentes). Esta es la principal diferencia con la estructura basada en el modelo matemático,

en la cual el grafo representaba todo el sistema. En esta estructura, surge una clase superior llamada “sistema”.

Sistema

Como se mencionó anteriormente, un sistema representa un producto y, por lo tanto, implica conexiones entre grafos direccionales. La interacción de las clases del sistema se ejemplifica en la figura 3-7.

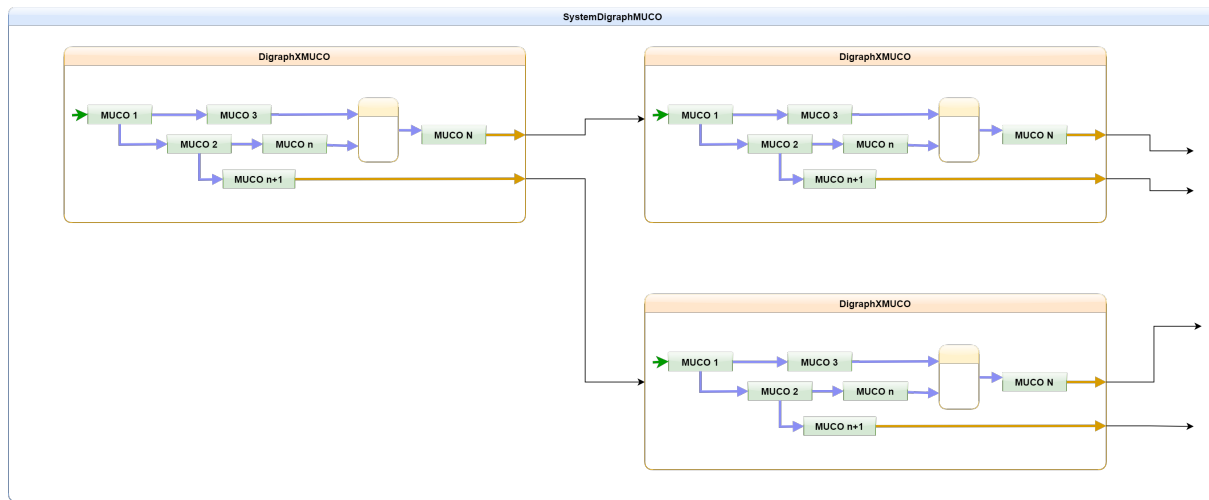


Figura 3-7.: Definición de sistema como la interacción de componentes.

Fuente: Creación Propia

Las clases del sistema contienen información general del producto, como los requerimientos a solventar, la evaluación de los requerimientos y la definición de interacciones fijas o móviles entre los diferentes componentes o grafos. Esta clase utiliza métodos de desarrollo de productos, que pueden hacer uso de computación paralela para su desarrollo, como se observó en el diagrama de clases de la Figura 3-5.

3.1.3. Desarrollo de las estructuras de datos

En el proceso de desarrollo de las estructuras de datos, se observa una dependencia secuencial entre los grafos, como se puede apreciar en la figura 3-3 y en la figura 3-7. Esto implica que la actualización de las salidas de un nodo debe ocurrir antes de que se pueda operar el siguiente.

Basado en modelamiento matemático

Dado que los árboles en cada nodo del grafo direccional permiten almacenar una ecuación de modelado para obtener una salida única, consideramos que cada MUCO puede tener

varios árboles o funciones que describen su modelado. Para evitar operaciones secuenciales en la resolución de los árboles, se incluyen las funciones directamente en el código, lo que representa una operación de complejidad $O(1)$.

Este modelo corresponde a la primera aproximación de la estructura de datos y fue desarrollado utilizando programación secuencial. Por lo tanto, el tiempo de operación está relacionado con la cantidad de grafos presentes en cada individuo de la población.

Basado en Primitivas

Este enfoque se basa en la secuencia de MUCOs para describir el componente, lo que implica que los MUCOs de niveles superiores no pueden resolverse hasta obtener el valor del MUCO anterior. En este enfoque, las operaciones en cada MUCO son fijas y se definen como ecuaciones que representan una ejecución computacional de complejidad $O(1)$.

Las ecuaciones que determinan las posiciones finales en dos dimensiones, basadas en los valores de distancia (*Dist*) y ángulo (*Angle*) definidos en cada MUCO apreciables en la figura **3-5**, se expresan de la siguiente manera:

$$X_{end} = X_{start} + \cos(Angle) * Dist \quad (3-1)$$

$$Y_{end} = Y_{start} + \sin(Angle) * Dist \quad (3-2)$$

Las ecuaciones mencionadas son fijas para cada MUCO y presentan una complejidad computacional baja, lo que cumple con los criterios establecidos para la programación paralela descrita en la sección 2.3.4. La cantidad de ejecuciones secuenciales necesarias se determina a partir de los caminos de conectividad de los MUCOs, representados por las líneas punteadas de color rojo en la figura **3-6**.

Para cada componente, es necesario desarrollar los MUCOs según el camino con la mayor cantidad de MUCOs conectados. Esto significa que una secuencia larga de MUCOs puede generar demoras en la resolución del componente. Los caminos se calculan para cada componente utilizando el algoritmo descrito en el Anexo B aprovechando la programación paralela en CUDA.

La obtención de caminos para cada componente se realiza en dos situaciones:

- Cuando se define por primera vez el componente como un grafo direccional.
- Cuando se presenta una mutación que varía los MUCOs del grafo direccional.

El uso de la programación paralela permite desarrollar múltiples componentes al mismo tiempo, lo que implica que el tiempo de ejecución será $O(N)$, donde N representa la cantidad de MUCOs conectados en el camino más largo entre todos los componentes.

3.2. Componentes rígidos y flexibles

En esta sección, se presentan los componentes rígidos y flexibles que se utilizan en el desarrollo de los componentes mediante estructuras para su realimentación. La clasificación de los elementos como rígidos se basa en la premisa de que el objeto no experimentará modificaciones de forma debido a fuerzas o agentes externos. Esta clasificación es especialmente útil en las etapas iniciales del proceso, ya que simplifica las operaciones. Por otro lado, los elementos flexibles consideran modificaciones adicionales de forma para lograr resultados más cercanos a la realidad.

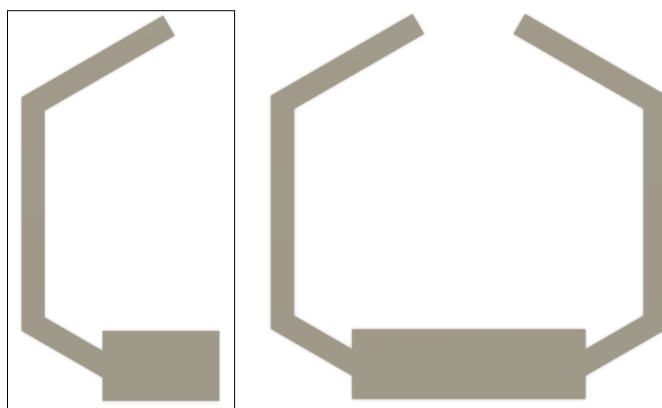


Figura 3-8.: Representación de conceptos fabricados en base al modelo y a primitivas.

Fuente: Creación Propia

3.2.1. Componentes rígidos

Con el objetivo de lograr la realimentación de componentes previamente fabricados y establecer este proceso para la creación de un Gripper, se presenta en la figura 3-8 un ejemplo de concepto de un Gripper con dos tenazas. La elección de este ejemplo se basa en la complejidad de la forma de cada tenaza, asumiendo que no se deforman al entrar en contacto con el objeto a sujetar.

En la figura 3-9.a se muestra el análisis de una tenaza del Gripper utilizando una estructura basada en modelo. En esta estructura, un MUCO se representa como una caja negra que contiene el modelo para cada salida. Si solo se busca representar la posición al final de la

línea roja, se utiliza una caja negra de color verde; mientras que si se desean representar diferentes formas de la geometría y también incluir la posición de salida de las tres líneas, se utiliza una caja negra de color morado. Por lo tanto, el planteamiento de los MUCOs depende de la precisión que se desee lograr en la consolidación del modelo, y esto queda a criterio del diseñador.

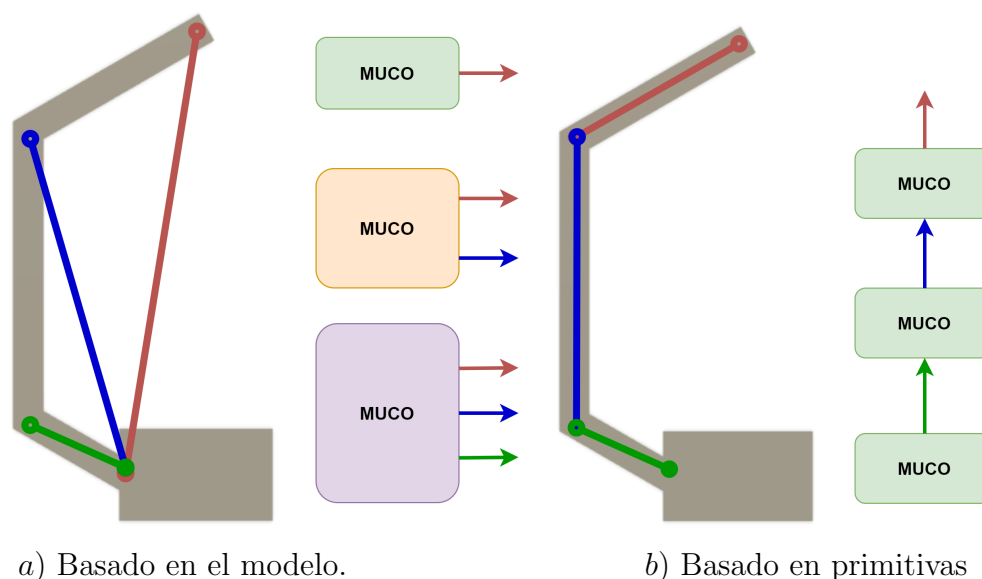


Figura 3-9.: Representación de conceptos fabricados en las estructuras de datos.

Fuente: Creación Propia

En cuanto a la estructura basada en primitivas, en la figura 3-9.b se observa que cada unidad mínima MUCO interactúa de forma secuencial para conformar la forma del Gripper. En este esquema, agregar más MUCOs mejora el modelado de un componente y permite establecer interfaces para responder a la interacción del Gripper con otro objeto, y en todos los casos se obtiene una solución. En el caso de la estructura basada en el modelo, agregar otra salida al MUCO implica agregar otra función al modelado, lo que aumenta la complejidad.

En el desarrollo de los conceptos a realimentar en las estructuras para el desarrollo de un Gripper, se propone, para ambas estructuras, un eslabón recto sin configuraciones especiales de forma. Esto permite modelar el MUCO de un eslabón como se muestra en la figura 3-10.

Al detallar la estructura del MUCO de un eslabón, se observa que las entradas de color azul son del mismo tipo que las salidas de color verde, lo que implica que las salidas de un MUCO pueden alimentar secuencialmente a otro MUCO. Sin embargo, las entradas de color rojo indican variables que son únicas para cada MUCO y deben ser definidas individualmente para cada MUCO creado.

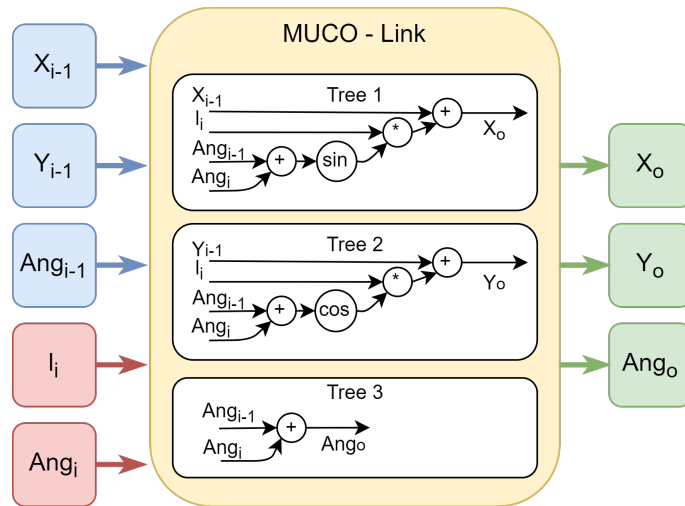


Figura 3-10.: MUCO de un eslabón basado en el modelo.

Fuente: Creación Propia

El estudio del eslabón más simple sienta las bases para la estructura de datos por primitivas, ya que es la descripción más simplificada del elemento rígido que puede interactuar con ella misma en las clases que lo componen. En esta estructura se destaca que las operaciones que resuelve cada MUCO basado en primitivas (descritas en las ecuaciones 3-1 y 3-2) son las mismas que las funciones descritas en la figura 3-10. Los valores específicos de distancia y ángulo se derivan de la necesidad de tener valores propios que no se heredan ni se transmiten de otro MUCO.

3.2.2. Componentes flexibles

El planteamiento de componentes flexibles solo se aplica en este trabajo a través de la estructura basada en primitivas. En esta estructura, se permiten modificaciones en el modelado mediante la adición o modificación automática de MUCOs, como se detalla en la sección 3.4. Por otro lado, en la estructura basada en el modelo matemático, no se han desarrollado herramientas para realizar modificaciones automáticas en los árboles debido a la complejidad analítica que requiere este tipo de soluciones.

Para abordar la necesidad de modelar la flexibilidad en los componentes, se pueden emplear dos metodologías:

- Flexibilidad en la estructura de datos: En esta metodología, se permite que la estructura de datos almacene otro valor final que muestre las variaciones de flexibilidad presentes en el componente. En la figura 3-11, se puede apreciar cómo se define un MUCO que incorpora otro punto final real como parte de la estructura. Este punto adicional está incluido en el esquema de clases mostrado en la figura 3-4 al comienzo del capítulo.

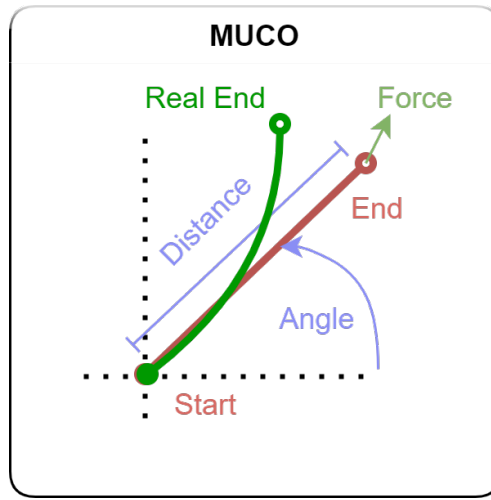


Figura 3-11.: Ampliación de un MUCO en la estructura de datos basado en primitivas.

Fuente: Creación Propia

En un esquema de elemento rígido, los dos puntos finales almacenados en la estructura tendrán el mismo valor.

- Planteamiento secuencial de MUCOs: En esta metodología, se busca ajustar la forma del objeto utilizando una secuencia de MUCOs, como se muestra en la figura 3-9.b. Esta secuencia de MUCOs permite la variación de los ángulos de todos los MUCOs del grafo, manteniendo la distancia de cada MUCO fija. Es importante destacar que a medida que la secuencia de MUCOs se vuelve más larga, se apreciará un mayor tiempo de operabilidad computacional, ya que el número de operaciones $O(N)$ que se deben realizar es igual a la secuencia más larga de MUCOs conectados.

3.3. Planteamiento de objetivos

El diseño generativo realimentado se basa en el establecimiento y evaluación de parámetros u objetivos a cumplir. En este enfoque, los objetivos se definen en términos de optimización, donde se consideran restricciones y se explora un espacio de trabajo definido para evaluar la minimización de dichas características.

Los objetivos representan la cuantificación de los requerimientos funcionales y pueden ser expresados a través de: (1) objetivos espaciales, (2) objetivos de fuerzas y (3) objetivos adicionales.

En la estructura de datos basada en el modelo matemático, la evaluación se realiza en función de los valores almacenados en los nodos MUCO de tipo salida al operar los valores de los

nodos terminales (ver figura **3-3**). Por otro lado, en la estructura basada en primitivas, se evalúan los nodos MUCO establecidas como salidas de la clase grafo direccional (ver figura **3-6**).

En cuanto al almacenamiento de los objetivos a resolver, se emplean dos enfoques distintos: (1) En el caso basado en el modelo matemático, se almacenan en un arreglo de escalares, ya que solo se necesita obtener el valor final. (2) En el caso basado en primitivas, como se muestra en la figura **3-4**, se define una clase "sistema de objetivos" que puede almacenar parámetros específicos u objetivos, como un grafo direccional, para incluir la interacción con componentes del mundo real.

Estos planteamientos de objetivos y su respectivo almacenamiento permiten orientar el proceso de diseño generativo realimentado hacia la obtención de resultados óptimos y la satisfacción de los requerimientos establecidos. En la siguiente sección se detallarán los objetivos de esta tesis de maestría.

3.3.1. Objetivos espaciales

Los objetivos espaciales se establecen para determinar puntos en el espacio en los cuales la distancia entre ellos y las salidas de las estructuras tiendan a cero. En el contexto de este trabajo, que se desarrolla en un análisis de dos dimensiones, es necesario identificar dos salidas en las coordenadas X e Y. Sin embargo, la evaluación se realiza mediante un único escalar que representa la magnitud de la diferencia entre los dos puntos. Este sistema es aplicado de igual manera para la estructura de datos basado en el modelado matemático y basado en primitivas, dado que en ambos casos se puede obtener el valor de las coordenadas finales de interés X y Y.

Cada punto definido se considera un objetivo en un sistema de optimización, lo que implica que la estructura será evaluada en términos de optimización multiobjetivo para verificar si el componente logra interactuar con todos los puntos definidos. La función de evaluación se define de la siguiente manera:

$$f(x) = \sqrt{(P_{objX} - X_{end})^2 + (P_{objY} - Y_{end})^2} \quad (3-3)$$

Donde P_{objX} y P_{objY} representan las coordenadas del objetivo a alcanzar y X_{end} y Y_{end} representan las salidas en las coordenadas X e Y de la estructura. Esta función calcula la distancia euclidiana entre el objetivo y las salidas de la estructura.

Además, se puede definir un grafo direccional en la estructura basada en primitivas, que represente otro componente. En este caso, aunque el nuevo componente pueda tener múltiples salidas, la evaluación retornará el valor mínimo encontrado entre todas las distancias entre

las salidas de los grafos direccionales (tanto del componente diseñado como del componente objetivo). Un ejemplo de la aplicación se define en el desarrollo del Gripper en la figura 5-2.

Además, es importante destacar que se desarrolló un proceso de optimización mono-objetivo para generar la variabilidad en la posición de los nodos MUCO de salida de un componente, con el fin de abarcar todos los puntos objetivo definidos, aplicable de igual manera a ambas estructuras de datos. En este proceso, se trabaja con una población de grafos en los que se modifican las posiciones angulares de las conexiones de los eslabones. Para cada punto objetivo se selecciona la configuración angular del eslabón que brinde la menor evaluación para la función objetivo.

En el caso de una configuración de cadena de eslabones y para abordar M puntos objetivo, se establecen M configuraciones angulares para cada eslabón. Este proceso generativo de obtención angular, dentro del marco del proceso generativo realimentado global, implica un aumento en la cantidad de memoria y operaciones requeridas. Por lo tanto, su viabilidad se observa principalmente en un entorno desarrollado en computación paralela.

La implementación de este proceso de optimización mono-objetivo permite generar la diversidad necesaria en las posiciones de los nodos MUCO de salida, lo que a su vez contribuye a abordar de manera efectiva los diferentes puntos objetivo definidos.

Adoptar objetivos espaciales permite asegurar que el componente diseñado sea capaz de alcanzar y mantener una posición específica en el espacio, lo cual es fundamental para su funcionamiento en diversas aplicaciones. En la siguiente subsección se abordarán los objetivos relacionados con las fuerzas.

3.3.2. Objetivos de fuerzas

En este trabajo, se plantea un enfoque estático que se basa en lograr el equilibrio del sistema. Para mantener el sistema en equilibrio, se utilizan técnicas de optimización en un sistema realimentado para determinar las fuerzas. En consecuencia, las fuerzas no se obtienen a partir de un modelo matemático, ya que esto sería computacionalmente ineficiente. Sin embargo, esta aproximación sienta las bases para realizar simulaciones dinámicas en investigaciones futuras.

El cálculo de la transmisión de fuerzas se realiza mediante la generación de poblaciones con diferentes valores en el nodo MUCO base de un grafo direccional. Se opera sobre estas poblaciones para obtener una salida en el nodo MUCO de salida, explorando así el espacio de diseño. Al igual que con los objetivos espaciales, se emplea un proceso generacional en cada generación para seleccionar la solución que cumpla con los requerimientos de fuerza en los nodos MUCO de salida.

La representación de las fuerzas se expresan en cadena considerando F_{ix} y F_{iy} la fuerza que se expresa en el inicio del eslabón en los ejes X y Y respectivamente y dependiendo de F_{jx} y F_{jy} las fuerzas que se expresan al final del eslabón. En el caso de la conectividad por grafos direccionales las fuerzas que se expresan al final del eslabón son correspondientes a las fuerzas del inicio del siguiente eslabón conectado. Se define que las fuerzas externas solo se pueden dar en los puntos finales de los eslabones, por lo cual la formulación se encuentra planteado en el siguientes formulas.

$$F_{ix} = F_{jx} * \cos(\theta) - F_{jy} * \sen(\theta) \quad (3-4)$$

$$F_{iy} = F_{jx} * \sen(\theta) + F_{jy} * \cos(\theta) \quad (3-5)$$

La descripción de esta transición de fuerzas derivan de la matriz de rotación y de la aplicación de matrices homogéneas en robótica expresadas por Craig, J et. al [85] en el análisis de fuerzas estáticas en manipuladores. En la cual hace uso de las matrices homogéneas para representar las matrices de rotación y las fuerzas estáticas. Como se detalla este proceso implica que se resuelva desde el ultimo MUCO al primero, por lo cual el desarrollo de los caminos se resuelven al inverso de como se describe las conexiones de los MUCOS en los grafos direccionales de ambas estructuras.

Dado que se requiere transmitir la fuerza desde el punto final del MUCO hasta el punto inicial, es necesario calcular el momento resultante en el punto inicial del MUCO. Esto se logra mediante la aplicación de la ecuación 3-6, donde \vec{M} representa el momento, \vec{R} es el vector de distancia entre los puntos y \vec{F} es la fuerza aplicada.

$$\vec{M} = \vec{R} \times \vec{F} \quad (3-6)$$

Además, en situaciones donde el punto final del MUCO hereda un momento de otro MUCO (\vec{M}_O), se utiliza la ecuación 3-7 para calcular el momento en el punto inicial ($\vec{M}_{O'}$). Esta ecuación permite trasladar el momento heredado al punto inicial considerando la distancia s y la fuerza aplicada \vec{F} en el punto O. Se utiliza el producto vectorial \times para realizar el cálculo adecuado del momento.

$$\vec{M}_{O'} = \vec{M}_O + \vec{R} \times \vec{F} \quad (3-7)$$

El enfoque de optimización utilizado en este trabajo permite determinar las fuerzas necesarias para mantener el equilibrio del sistema y cumplir con los requerimientos específicos de cada componente. En la siguiente subsección se abordarán los objetivos adicionales contemplados en el diseño generativo realimentado.

3.3.3. Objetivos en diferentes etapas del ciclo de vida del producto

La metodología propuesta en este trabajo ofrece la flexibilidad necesaria para obtener valores simuladas de etapas del ciclo de vida del producto, lo cual brinda un respaldo invaluable durante el proceso de diseño. Esta capacidad permite reducir la cantidad de iteraciones necesarias para obtener el producto final deseado. Además, en algunos casos, el proceso de diseño puede requerir ajustes por parte del diseñador para adaptar el producto al proceso de manufactura, sin necesidad de modificar la concepción del concepto original.

Dado que la filosofía de esta tesis de maestría se basa en entornos de manufactura flexibles, se detallarán las métricas que nos permiten evaluar la influencia de esta filosofía en el proceso de diseño, ajustadas a la estructura de datos propuesta. Adicionalmente, el desarrollo de estas métricas en las diferentes etapas del ciclo de vida de un producto se abordará en el siguiente capítulo 4, a través de mecanismos que nos permiten realizar aproximaciones en términos de tiempo, material, forma, disponibilidad, entre otros aspectos relevantes.

Objetivos de flexibilidad

Al analizar el estado del arte que ya se mencionó en la sección 2.3.2, se observa la implementación de matrices de correlación de componentes, como una matriz DSM (Design Structure Matrix) o una matriz basada en interfaz DSM, y ajustes de pesos de complejidad por cada componente. Según la Real Academia Española (RAE), la complejidad se refiere a "la composición de elementos diversos", lo que implica que la medición de la complejidad se basa en la interacción de los elementos que componen el producto. En este contexto, se considera que la complejidad de un componente surge de su interacción con otros componentes en el sistema como lo detalla Kim et al., 2016 [69].

Las propiedades son las características identificables y medibles, como el tiempo de ciclo de manufactura, la tolerancia, la disponibilidad tecnológica, la complejidad tecnológica, entre otros. Estas propiedades se utilizan para asignar un valor de complejidad a cada componente.

En el caso de la estructura propuesta en este trabajo, la conexión de los MUCOs en los grafos direccionales se considera como la interacción entre los componentes, sin la necesidad de generar una matriz DSM. Además, se aplican variables de ajuste establecidas por el diseñador, quienes evalúan subjetivamente cada propiedad en una escala del 1 al 5. Esto permite evaluar la complejidad de manufactura del objeto en etapas posteriores del ciclo de vida del producto.

$$Flex = \sum_{p=1}^N C_p + \sum_{i=1}^N \sum_{j=1}^N W_{ij} * I_{ij} \quad (3-8)$$

La ecuación 3-8 muestra la formulación de flexibilidad *Flex* propuesta en este trabajo, que combina la complejidad de los N componentes (C_p) y las interacciones entre componentes (I_{ij}) ponderadas por los pesos (W_{ij}). La suma discreta de estas métricas da como resultado la flexibilidad del producto. Un valor de Flexibilidad menor es el objetivo a satisfacer, dado que entre mas componentes interactúen o mayor sea la complejidad de algún componente se refleja en un mayor valor de flexibilidad.

Asikoglu et al., desarrolla el concepto de interacción entre componentes de un producto considerando diversos tipos de conectividad que lo expresa asignando pesos a las conexiones [70]. La conexión entre nodos MUCO que componen un grafo direccional representa las interacciones entre los componentes del producto en este tesis de maestría. En la ecuación 3-9 se muestra como la interacción entre componentes se da por la cantidad de grafos direccionales componen el sistema *nodosGrafo*, al suponer que la interconexión de los grafos siempre transmiten la misma información y ningún nodo MUCO tiene información adicional, lo que permite plantear con el mismo peso.

$$Flex = \sum_{p=1}^N C_p + \text{nodosGrafo} \quad (3-9)$$

Las propiedades del componente, como el tiempo de ciclo de manufactura, la tolerancia, la disponibilidad tecnológica y la complejidad tecnológica, entre otros, son características identificables y medibles que se utilizan para asignar un valor de complejidad. Estas propiedades se evalúan en el proceso de normalización, donde los valores obtenidos para cada propiedad se escalan entre 1 y 5 en función del valor máximo y mínimo. Asimismo, la evaluación de la complejidad incluye la experiencia y experticia del diseñador en propiedades cualitativas.

En la ecuación 3-10, se expresa la complejidad de cada componente (C_p) como la suma de las propiedades del componente (P_x) normalizada por el número total de propiedades (X). Además, se considera en la estructura basada en primitivas, donde la cantidad de nodos MUCO en un grafo direccional para cada componente ($MUCOs_p$) indica una propiedad de forma para el cálculo de la complejidad (C_p). Para la estructura basada en primitivas, se normaliza la cantidad de MUCOs en relación con el componente con la mayor cantidad de MUCOs. En el caso de la estructura basada en modelos matemáticos la complejidad de forma (C_{forma}) se describe directamente por el diseñador utilizando una escala de 1 a 5. Esta escala se basa en las métricas de complejidad usadas por Kim et al., 2016 [69], en el cual en el trabajo específico el establece tres propiedades que considera con igual peso y lo representa por la suma de estas.

$$C_p = \frac{\sum_{x=1}^X P_x}{X} \quad (3-10)$$

En la función de evaluación de las propiedades (ecuación 3-11), se asigna un valor de complejidad a cada propiedad. Los términos "mínima complejidad" y "poca complejidad" se refieren a las evaluaciones de los componentes basadas en las propiedades, las cuales están normalizadas entre 1 y 5. También se considera la evaluación o experticia del diseñador en propiedades cualitativas.

$$P(x) = \begin{cases} 1 & \text{mínima complejidad} \\ 2 & \text{poca complejidad} \\ 3 & \text{media complejidad} \\ 4 & \text{alta complejidad} \\ 5 & \text{máxima complejidad} \end{cases} \quad (3-11)$$

En cuanto a la normalización, esta se realiza durante el proceso de evaluación de las propiedades. Se toman los valores obtenidos para cada propiedad y se asigna un valor normalizado entre 1 y 5, considerando el valor máximo y mínimo obtenido. De esta manera, se establece una escala para comparar la complejidad de los componentes.

Los ejemplos de aplicación de esta métrica se pueden encontrar en el capítulo 5 de la tesis de maestría, donde se detallan casos específicos que ilustran cómo se utilizan las métricas de flexibilidad en el proceso de diseño y manufactura.

3.4. Operaciones de mutación

Con el objetivo de generar conceptos basados en componentes previamente fabricados, se analiza el proceso mediante el cual las estructuras de datos pueden generar modificaciones en los componentes. En las estructuras de datos basadas en modelos matemáticos, se plantean los árboles de operaciones para modificar el modelado del componente. La complejidad radica en que este proceso implica realizar modificaciones del modelo basadas en conceptos matemáticos. Además, la definición de un modelado puede depender de varios árboles, lo que implica encontrar un sistema de ecuaciones para resolver las dependencias entre estos árboles. En este contexto, resulta inviable generar modificaciones en un componente bajo este tipo de estructura.

En contraste, la estructura de datos basada en primitivas ofrece ventajas significativas en esta característica. Debido a que son una descripción simplificada y detallada de un componente por medio de nodos MUCO, se pueden realizar cuatro operaciones como se muestra en la figura 3-12. En la figura, el MUCO en color rojo representa el componente al que se le aplicarán las operaciones, mientras que el MUCO en color verde indica una interacción que debe mantenerse en el tiempo para no afectar el modelado previo del componente.

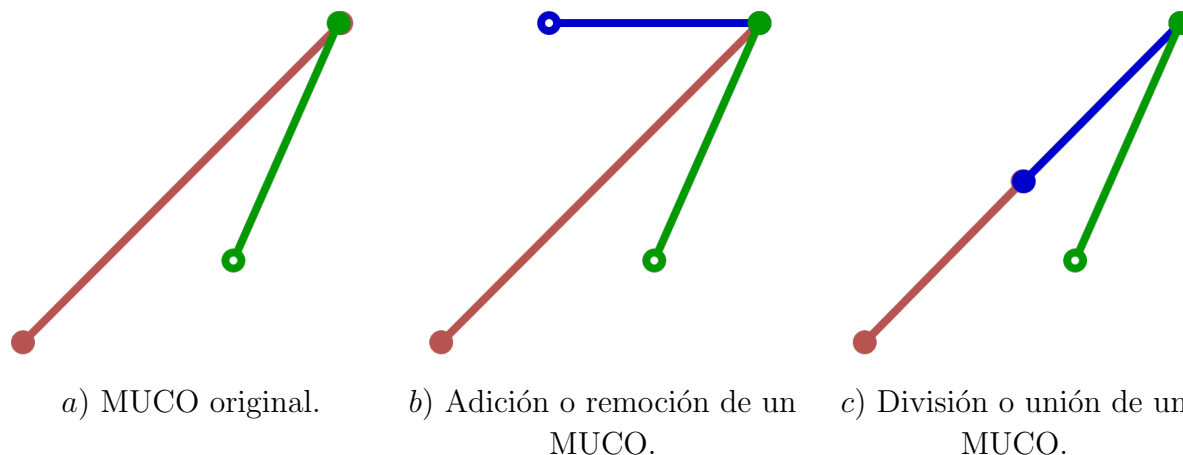


Figura 3-12.: Operaciones en el grafo direccional por primitivas.

Fuente: Creación Propia

La primera operación de la figura **3-12.**(b) consiste en agregar otro MUCO (azul) al final de un MUCO base (rojo). Este nuevo MUCO puede tener cualquier longitud y ángulo, lo que brinda mayor versatilidad al modelado. Es importante destacar que si el MUCO tiene conexiones con otros MUCOs, como el verde, estas conexiones no se verán alteradas. Sin embargo, es importante mencionar que la operación inversa, es decir, la eliminación de un MUCO, solo permite eliminar el MUCO en color azul o en color verde, siempre y cuando no tenga una conexión con otro MUCO.

En cuanto a la operación de división de la figura **3-12.**(c), se observa cómo el MUCO inicial en rojo del inciso (a) se reduce a la mitad de su longitud, y las conexiones o valores que se proyectaban en la salida pasan a formar parte del nuevo MUCO en azul. La operación inversa, la unión, solo tiene en cuenta la posición del punto final del MUCO azul y la posición del punto inicial del MUCO en rojo. Por lo tanto, el ángulo y la longitud de estos dos MUCOs no se tienen en cuenta y se calculan en base a los dos puntos definidos.

3.5. Conclusiones de las estructuras de datos

En esta tesis de maestría se ha explorado la estructura de datos basada en el modelo matemático con un enfoque de árboles para la modificación de componentes. Al analizar y describir en detalle los componentes de baja complejidad, en específico un eslabón, se ha introducido el concepto de primitivas como entidades fundamentales en la estructura de datos por primitivas. Esta estructura permite resolver elementos complejos y alimentar los procesos de evaluación en el diseño generativo, al establecer una conexión de estas primitivas llamadas nodos MUCO entre ellas.

Se ha observado que aunque la estructura de árbol ayuda a establecer una interfaz con los

diseñadores para expresar su conocimiento de modelado matemático mediante ecuaciones, es posible omitir esta estructura y lograr un mayor desarrollo computacional al incluir directamente las ecuaciones en el programa. Sin embargo, esto implica un desarrollo manual más extenso en el código para la ejecución y puede afectar la escalabilidad del almacenamiento de las ecuaciones en bases de datos.

El concepto de primitivas abarca tanto productos existentes como conceptos no explorados, al establecer el desarrollo de las unidades menores MUCO por primitivas y sus operaciones de mutación que se pueden generar automáticamente, lo que le confiere una gran flexibilidad como mecanismo en el diseño generativo. Esto establece las bases para la automatización de los procesos de creación de productos.

A diferencia de la estructura basada en el modelo matemático, se destaca que en el uso de primitivas como estructura nos ha permitido aprovechar la capacidad computacional paralela, ya que cada componente tiene un mecanismo de resolución detallado y un tiempo de ejecución reducido. Además, esta estructura puede ser modificada y explorada en modelos metaheurísticos, lo que amplía aún más las posibilidades de generación de nuevos productos.

La definición de objetivos mediante la definición de un grafo direccional con varios nodos MUCO ha demostrado ser efectiva para superar las limitaciones de los softwares CAD tradicionales, que se enfocan únicamente en la optimización de la forma o topología. Esto ha permitido aplicar la estructura de datos por primitivas en diferentes tipos de modelos de fábricas, como en el entorno de manufactura flexible que ha sido objeto de estudio en esta tesis de maestría.

El modelado de componentes flexibles ha mostrado ser adaptable a la secuenciación de objetos rígidos, lo que brinda flexibilidad en la resolución de los objetivos deseados. Sin embargo, se han identificado problemas en los tiempos de ejecución debido a las operaciones secuenciales de los nodos MUCO en cada grafo direccional, que representan una mayor de tiempo de cómputo. Estos desafíos se plantean como áreas de investigación futuras.

4. Aplicación del Diseño Generativo Realimentado

En el ámbito de la creación de productos, es comprensible que los procesos no sean lineales y requieran varias iteraciones a lo largo de diferentes etapas del ciclo de vida. Estimar y resolver todas las implicaciones físicas y funcionales de un producto resulta altamente complejo y, en muchos casos, requiere la experiencia de múltiples diseñadores con enfoques especializados en diversas áreas. En este contexto, el diseño generativo surge como una propuesta que estructura y unifica la información del producto en forma de un modelo que puede operar en diferentes entornos, con el objetivo de obtener información simulada de estas etapas y reducir la necesidad de iteraciones en la manufactura del producto.

Este capítulo se enfoca en el desarrollo de entornos para cada etapa del ciclo de vida del producto bajo la premisa del diseño generativo. El objetivo es proyectar simulaciones que representen un entorno cercano a la realidad, de manera que en un futuro el apoyo del diseño generativo no se limite únicamente a la etapa de diseño, sino que también aporte en las demás etapas, generando así una automatización general del proceso de creación de productos.

El desarrollo en las otras etapas del ciclo de vida del producto implica comprender la estructura del mismo (en base a primitivas, como se planteó anteriormente) para poder representar la información del producto en diferentes momentos del ciclo de vida. La interacción con nuevas estructuras específicas según la etapa de diseño se vuelve evidente, ya que la estructuración presentada en el capítulo anterior se enfoca únicamente en el modelado físico y espacial del producto.

Además, es importante destacar el concepto de realimentación, el cual es relevante tanto en la etapa de diseño a través de las simulaciones del ciclo de vida del producto, como en el proceso de manufactura. El primer producto manufacturado puede generar iteraciones adicionales con el objetivo de mejorar las características finales del producto, buscando reducir la cantidad de iteraciones en la manufactura al apoyarse en un mayor número de iteraciones en los procesos de simulación.

Cabe mencionar que la realimentación en la manufactura puede generar valores o condiciones que permitan identificar las diferencias obtenidas con respecto a la etapa de simulación. Estos valores pueden ser utilizados en métodos de aprendizaje automático para mejorar la

simulación al aplicar las operaciones en los nodos MUCO y obtener valores más cercanos a la realidad.

En resumen, este capítulo se centra en la aplicación del diseño generativo realimentado, explorando la implementación de entornos de simulación en las etapas del ciclo de vida del producto de diseño, manufactura y uso. Se busca ampliar el alcance del diseño generativo más allá de la etapa de diseño, automatizando el proceso de creación de productos y aprovechando la realimentación tanto en la etapa de manufactura como en las simulaciones del ciclo de vida del producto.

4.1. Concepciones del diseño generativo en el diseño

En el capítulo anterior, se desarrollaron todos los conceptos relacionados con el modelado de componentes mediante una estructura de datos. En el contexto de la etapa de diseño, se identifica que la representación conceptual descrita con esta estructura de datos debe traducirse a una representación mediante software CAD. En este sentido, se presenta la distinción entre el diseño CAD basado en superficies y el diseño CAD basado en contornos.

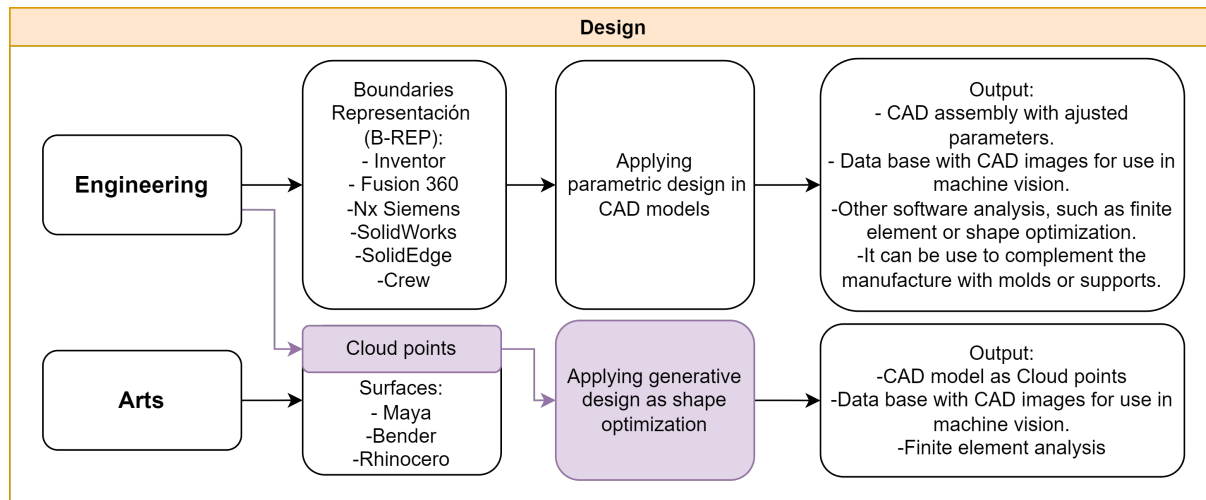


Figura 4-1.: Salidas de la etapa de diseño en el enfoque de diseño generativo.

Fuente: Creación Propia

La Figura 4-1 ilustra la propuesta de representación de la simulación en el diseño de manera detallada, así como las salidas que pueden alimentar el servicio del diseño generativo en las etapas posteriores del ciclo de vida del producto. En dicha figura, se observa que se trabajan dos enfoques: (1) basado en el diseño paramétrico, guiado por el campo de la ingeniería y utilizando representaciones basadas en contornos en CAD, y (2) basado en la representación mediante nube de puntos, representada en color morado.

Estas dos enfoques generan salidas del sistema principalmente basadas en la representación CAD. Sin embargo, al incorporar el flujo de información en diferentes etapas y sistemas de tipo MES, se pueden generar información adicional para complementar etapas posteriores, como la visión de máquina en manufactura, que se desarrolla en detalle en la sección 4.4, así como los procesos de análisis de elementos finitos, optimización de forma, entre otros.

En resumen, en esta sección se exploran las concepciones del diseño generativo en el contexto del diseño. Se destaca la importancia de la representación mediante software CAD, considerando tanto el enfoque paramétrico como el enfoque basado en nube de puntos. Además, se señala que estas representaciones pueden generar salidas que, al integrarse con otros sistemas y etapas del ciclo de vida del producto, enriquecen el proceso de diseño generativo y brindan información relevante para etapas posteriores.

4.1.1. Basado en diseño paramétrico

En el caso del planteamiento basado en diseño paramétrico, se pueden aplicar ambas estructuras de datos. La Figura 4-2 muestra el modelado de un componente, donde se detallan las interfaces que presenta para interactuar con otros componentes, representadas como entradas y salidas en la Figura 4-2.(a). En la Figura 4-2.(b), se ilustra cómo los modelos basados en diseño paramétrico pueden anular características de las interfaces al establecer los valores base de las geometrías como cero.

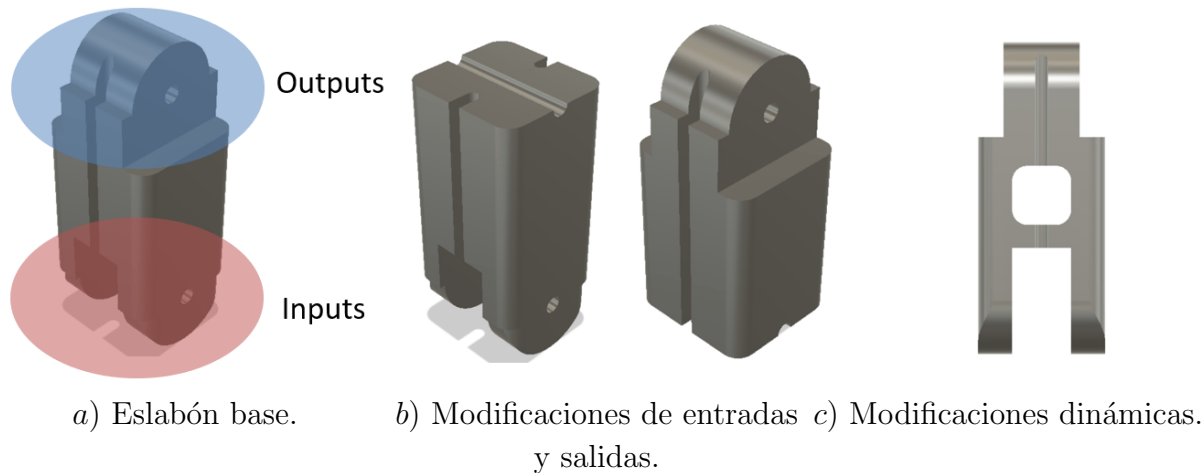


Figura 4-2.: Planteamiento de un eslabón bajo el diseño paramétrico.

Fuente: Creación Propia

Esta flexibilidad permite considerar las interacciones con otros componentes, ya que un componente sin salidas no tendrá interacción por interfaces con otros componentes, y un componente sin entradas se convierte en la pieza base de un producto. Sin embargo, en el proceso de diseño detallado, el diseñador debe operar sobre este componente para establecer interfaces forzadas con el entorno.

Adicionalmente, en la Figura 4-2.(c) se muestra la posibilidad de agregar otras operaciones que respondan a modificaciones dinámicas del componente. Esto implica que el conocimiento del componente fabricado puede ser transmitido en gran medida mediante el desarrollo de un modelo paramétrico y calculado según las especificaciones buscadas en esta tesis de maestría.

Sin embargo, este enfoque tiene limitaciones. En primer lugar, se basa en componentes previamente diseñados y modelados, lo que implica restricciones en la creación automática de componentes y en la capacidad de modificar su forma de manera libre. Los valores obtenidos deben ser seleccionados y representados en el modelo paramétrico por el diseñador.

Otra limitación se observa en la definición de interfaces. Para cada posible definición de interfaz de entrada o salida, se debe plantear un modelo paramétrico para cada configuración. Alternativamente, se puede automatizar la configuración de ensambles rígidos entre el cuerpo del componente y las interfaces, lo que implica tener una base de datos de diseños paramétricos de interfaces. Estas configuraciones se pueden definir mediante etiquetas en cada componente, expresando el tipo de interfaces a plantear, y bajo operaciones de estructuras, determinar cuál es la mejor configuración.

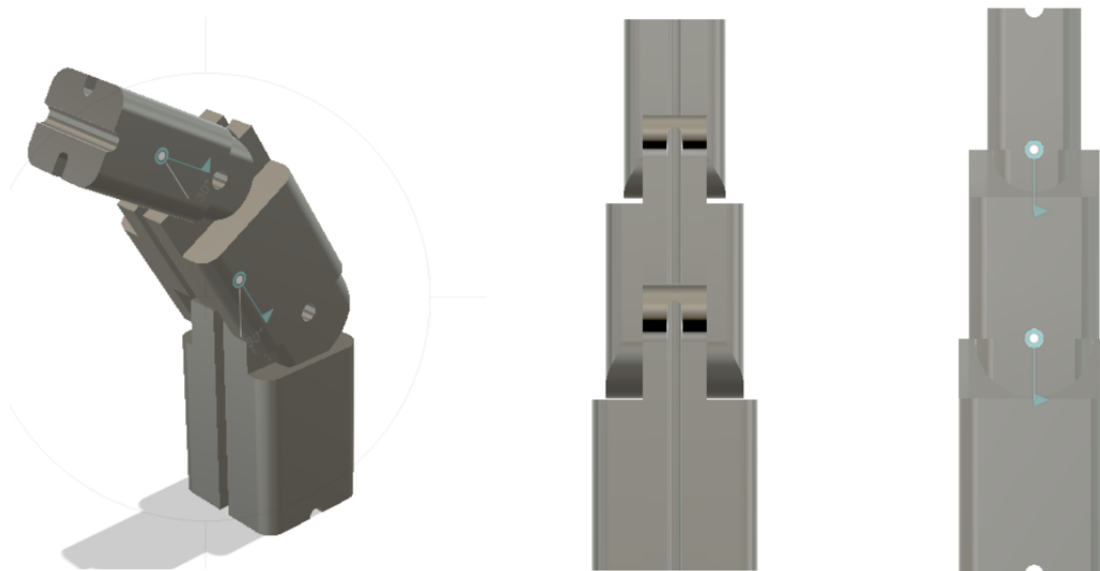


Figura 4-3.: Cadena cinemática generada por la interacción del concepto de eslabón.

Fuente: Creación Propia

En el caso de la aplicación de este trabajo en un Gripper, la unidad mínima sería un eslabón, como se plantea en la Figura 4-2.(a), donde las interfaces de interacción son conexiones mediante ejes en las entradas y salidas. Como se explicó en el capítulo anterior, el estudio de un eslabón define la representación de un nodo MUCO en la estructura basada en primitivas. En este caso, se aprecia mejor cómo este concepto puede aplicarse en ambas estructuras de datos.

En la Figura 4-3, se muestra cómo un eslabón interactúa consigo mismo para generar automáticamente el ensamblaje de una cadena cinemática. Dado que el análisis de las estructuras se realiza en dos dimensiones y la representación del modelo responde a tres dimensiones, el acoplamiento en la dimensión adicional depende de la automatización realizada por el diseñador en el diseño paramétrico, siguiendo la estructura de grafo direccionales de ambas estructuras. Es importante identificar cuál es el eslabón base y cuál depende del siguiente en el diseño del componente.

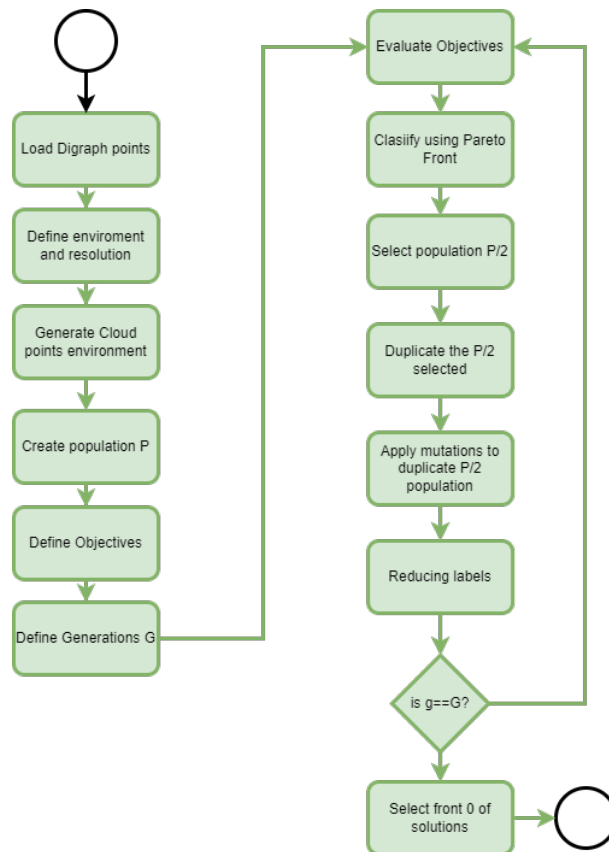


Figura 4-4.: Flujo planteado para la representación de un componente como una nube de puntos.

Fuente: Creación Propia

4.1.2. Basado en la representación por nube de puntos

En este trabajo de tesis, se propone la representación por nube de puntos como una forma de obtener una representación de un componente basada en las primitivas de la estructura de datos. Esta representación se basa en el desarrollo de un componente utilizando un modelo físico CAD que sea operable. Al establecer un ambiente que contenga estas primitivas a gran escala, se amplía la concepción de primitivas planteada en la estructura de datos. La

propuesta de representación se desarrolla a partir de un componente contenido en un grafo direccional y sigue el flujo descrito en la figura 4-4.

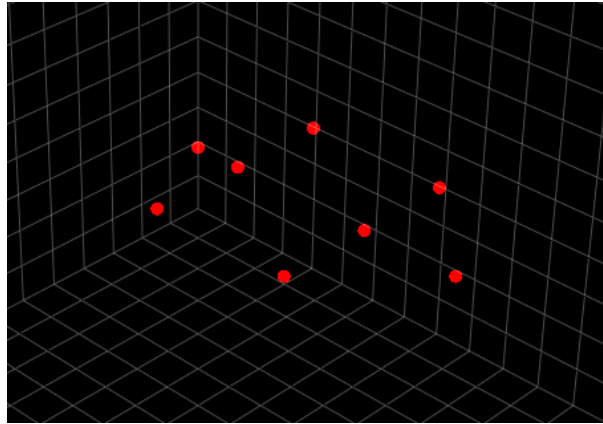
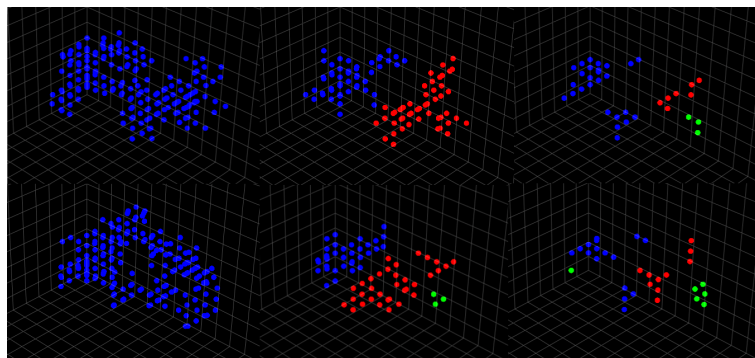


Figura 4-5.: Representación del grafo inicial en la nube de puntos.

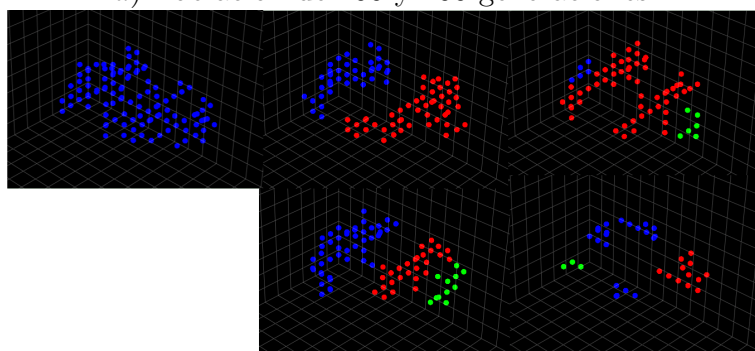
Fuente: Creación Propia

El primer paso en este enfoque consiste en cargar los puntos del grafo direccional, que representa el modelado realizado. La figura 4-5 muestra la ubicación de los puntos iniciales y finales de una definición de eslabón en el entorno de nube de puntos. La definición del entorno implica generar una matriz de nube de puntos de tamaño superior a los límites obtenidos por la descripción del componente en el grafo, y la resolución indica la separación espacial entre los puntos de la nube.

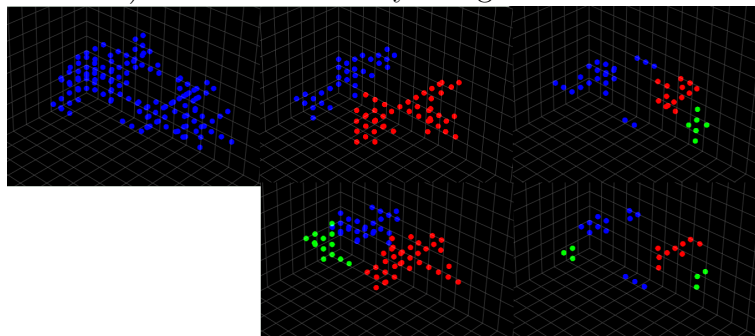
La generación de la población implica replicar el entorno con los puntos del grafo y realizar mutaciones de adición o sustracción de puntos vecinos en estas nubes de puntos para crear el componente. Dado que este trabajo se basa en el diseño generativo realimentado, que busca crear soluciones basadas en los requisitos planteados, se definen dos objetivos contradictorios: minimizar la cantidad de puntos y minimizar la cantidad de etiquetas o segmentos de puntos. Las etiquetas se asignan a cada punto vecino de los puntos del grafo, y si comparten un vecindario con otra etiqueta, se mantiene solo una etiqueta con el valor más bajo.



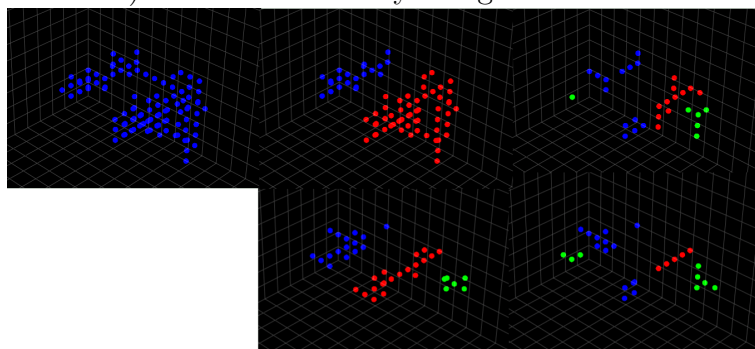
a) Población de 100 y 100 generaciones.



b) Población de 200 y 100 generaciones.



c) Población de 500 y 200 generaciones.



d) Población de 500 y 1000 generaciones.

Figura 4-6.: Representación generativa por nube de puntos.

Fuente: Creación Propia

El desarrollo de la población se basa en el principio de NSGA-II para la resolución de problemas multiobjetivo. Se define un número específico de generaciones que evalúan los objetivos y, mediante el uso del frente de Pareto, se genera un ordenamiento de la población. Siguiendo el enfoque de NSGA-II, se selecciona la mitad de la población y se duplica. A la población duplicada se le aplican mutaciones de adición o sustracción de puntos vecinos en cada segmento de puntos, y luego se verifica si es posible reducir la cantidad de etiquetas en el componente.

En la figura 4-6, se pueden observar diferentes soluciones obtenidas a partir de variaciones en la población y generaciones. Los resultados del lado izquierdo corresponden a valores que cumplen con el objetivo de tener la menor cantidad de etiquetas, mientras que en el lado derecho se muestran soluciones que se ajustan mejor a mantener la menor cantidad de puntos. El uso del frente de Pareto nos permite clasificar estas soluciones, lo que proporciona al diseñador las bases necesarias para seleccionar la solución deseada.

Aunque esta aproximación da las bases de como un entorno de desarrollo por nube de puntos se puede expandir para el desarrollo de forma de un componente, se establece que un sistema de dos objetivos siguen conteniendo pocas propiedades para representar un modelamiento de un componente a manufacturar. Por lo anterior se resalta para trabajos futuros el mejoramiento de esta técnica debido a la capacidad que demuestra este entorno de nube de puntos para complementar procesos semi-automáticos en la etapa de diseño y que en esta tesis de maestría no se desarrollan debido a que seleccionan los procesos de manufactura estandarizados en el LabFabEx-UN, como lo es el diseño paramétrico.

4.2. Concepciones del diseño generativo en la manufactura y almacenamiento

En esta sección, se exploran las distintas aplicaciones del diseño generativo en la etapa de manufactura y almacenamiento, considerando la diversidad de tecnologías y enfoques que pueden ser empleados. El flujo de las salidas de la etapa de diseño proporciona representaciones del modelado CAD, que posteriormente deben ser manufacturadas, adaptándose a las capacidades tecnológicas de cada fábrica, como se muestra en la figura 4-7.

En términos generales, las aplicaciones del diseño generativo en manufactura y almacenamiento pueden clasificarse en tres categorías:

1. Software comercial para generación de código G: Esta categoría abarca el uso de software comercial para la generación automatizada de código G utilizado en maquinaria CNC, tanto para manufactura sustractiva como aditiva. En este ámbito, se observa una privatización significativa en algunos software, especialmente en el contexto de la

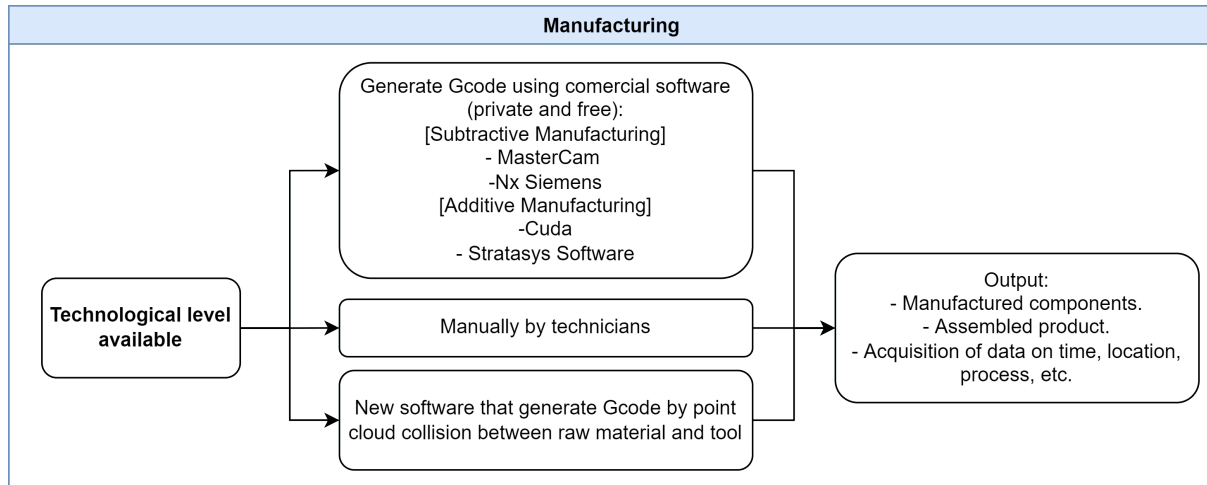


Figura 4-7.: Salidas de la etapa de manufactura en base al diseño generativo.

Fuente: Creación Propia

manufactura sustractiva. No obstante, la democratización de la manufactura aditiva ha llevado a una reducción de los precios de fabricación, así como al desarrollo de software gratuito de alta calidad para impresoras 3D FDM.

2. Manufactura manual asistida por técnicos: En esta categoría, se realiza la manufactura de piezas de manera manual por técnicos, lo que implica generar código G manualmente o incluso controlar la maquinaria CNC directamente para determinados movimientos. El principal desafío radica en el tiempo invertido en la generación del código, lo cual puede limitar la capacidad de producción en masa. No obstante, en el desarrollo de familias de productos, es posible generar macros que se ajusten a parámetros específicos, lo que agiliza el proceso.
3. Desarrollo de software personalizado: El último nivel de tecnología se refiere a la capacidad de generar nuevos software que satisfagan las necesidades de pequeñas y medianas empresas. En la sección 4.5.3, se presenta una solución basada en modelos obtenidos mediante nubes de puntos, lo cual permite desarrollar software en centros educativos enfocados en la capacitación continua de personal especializado para estas empresas. Esta perspectiva no solo democratiza la capacidad de producción tecnológica, sino que también fomenta la creación de empleo adaptado a las necesidades específicas de cada empresa.

El diseño generativo ofrece múltiples enfoques y aplicaciones en la manufactura y el almacenamiento, lo que permite adaptarse a las capacidades tecnológicas y necesidades de cada contexto productivo. Desde el uso de software comercial para automatizar la generación de código G hasta la generación de software personalizado, el diseño generativo se presenta

como una herramienta versátil y poderosa para impulsar la eficiencia y la innovación en la industria manufacturera.

4.3. Concepciones del diseño generativo en el uso

En la etapa de uso, se pueden obtener valiosas informaciones del componente, pero en este trabajo de tesis, se centra en establecer las bases para la identificación de componentes mediante el uso de imágenes sintéticas obtenidas del proceso de modelación y etiquetas derivadas de los grafos de representación. Estas etiquetas permiten expresar las características de forma más relevantes, en particular, se enfocan en longitudes y ángulos.

El enfoque de identificación de componentes a través de imágenes sintéticas es una técnica prometedora que ofrece múltiples ventajas. Mediante el uso de herramientas de diseño paramétrico planteados, se pueden generar imágenes sintéticas de los componentes en una variedad de configuraciones, lo que permite abarcar una amplia gama de posibles escenarios. Estas imágenes pueden representar tanto el componente en su estado ideal como en situaciones más realistas, considerando tolerancias y variaciones propias del proceso de manufactura. Este proceso se desarrolla en la sección 4.4 y en la figura 4-11 se puede observar un ejemplo de este proceso.

Además de las imágenes, se incorporan etiquetas que se derivan de la representación del componente en el grafo. Estas etiquetas capturan información clave sobre las longitudes y ángulos relevantes del componente, lo que facilita su identificación y clasificación en el proceso de uso. Al combinar las imágenes sintéticas y las etiquetas, se crea una poderosa herramienta para la identificación de componentes de manera automática y precisa.

El uso de esta metodología puede tener una amplia variedad de aplicaciones, desde la inspección y verificación de componentes en líneas de producción, hasta la identificación de piezas para el mantenimiento y reparación de equipos industriales. Además, esta técnica puede ser útil en el seguimiento de componentes a lo largo de su ciclo de vida, lo que permite llevar un registro detallado de su desgaste y evolución con el tiempo.

Aunque este trabajo de tesis sienta las bases para el uso de imágenes sintéticas y etiquetas de grafos en la identificación de componentes, es importante destacar que existen múltiples oportunidades para futuras investigaciones y desarrollos. Por ejemplo, se podrían explorar técnicas avanzadas de visión artificial y aprendizaje automático para mejorar la precisión y robustez del sistema de identificación. Asimismo, se pueden incorporar otros atributos relevantes en las etiquetas, como materiales, acabados superficiales, entre otros, para enriquecer la información obtenida.

En resumen, la concepción del diseño generativo en el uso, enfocada en la identificación de componentes mediante imágenes sintéticas y etiquetas de grafos, representa un avance

significativo en la fabricación inteligente. Esta metodología ofrece una herramienta potente y versátil que puede mejorar la eficiencia, calidad y seguimiento de los componentes en diversas aplicaciones industriales. Con el continuo desarrollo y aplicación de estas técnicas, se espera que el diseño generativo realimentado tenga un impacto positivo en la industria manufacturera y en la gestión de activos a lo largo de toda su vida útil.

4.4. Generación de objetos inteligentes por visión de maquina

En esta sección, se aborda la concepción de objetos inteligentes en el contexto de los modelos contemporáneos de fabricación inteligente, como se presentó en la sección 2.3.1. Estos modelos buscan utilizar nuevas tecnologías para crear fábricas más adaptables a las demandas del mercado. Una de las ventajas clave de tener objetos inteligentes es la capacidad de realizar un seguimiento detallado de todos los componentes manufacturados, lo que facilita la toma de decisiones informadas en todas las etapas del ciclo de vida del producto.

Sin embargo, se destaca que un objeto inteligente debe proporcionar información al sistema sobre sí mismo. En el caso de la creación de componentes, esta información se proporciona indirectamente mediante la maquinaria de fabricación o mediante sistemas de visión de máquina, como se plantea en esta sección.

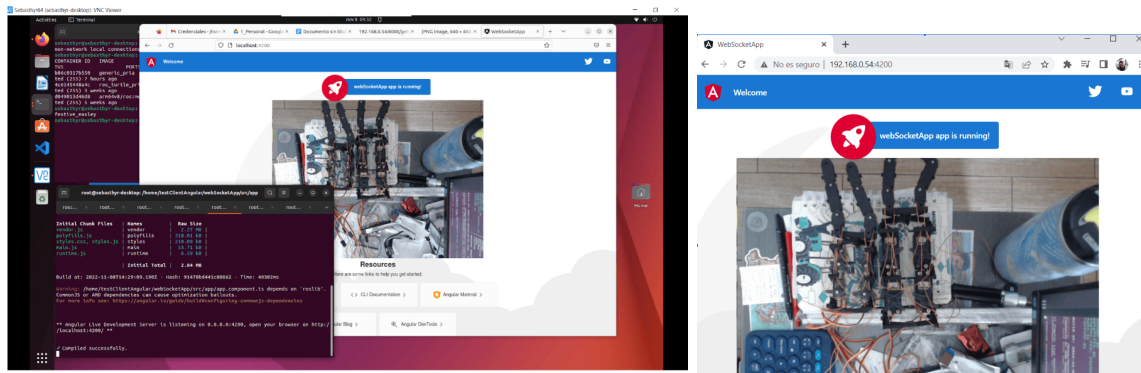


Figura 4-8.: Despliegue de servidor de visión en el LabFabEx-UN.

Fuente: Creación Propia

Los sistemas de visión de máquina se utilizan para obtener información de los componentes mediante el procesamiento de imágenes. Por lo general, estos sistemas requieren que el componente ya esté fabricado, lo que implica ciertas limitaciones. Sin embargo, existen desarrollos como el presentado por Restrepo et al. [1] y otros trabajos como el de Viana et al. [86] que muestran cómo es posible entrenar sistemas de visión basados en modelos CAD,

lo que permite el seguimiento de manufactura desde la etapa de diseño hasta la fabricación real.

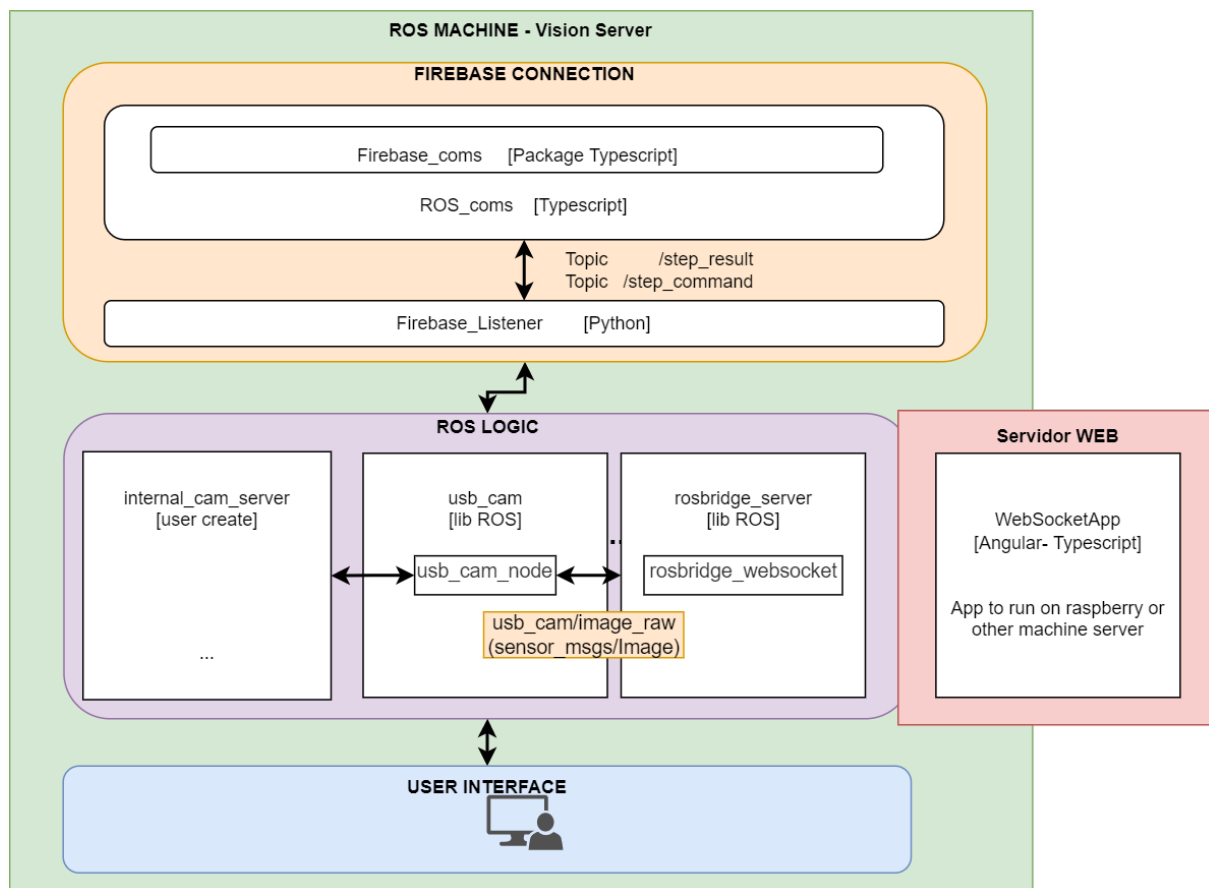


Figura 4-9.: Esquema de integración de servidores de visión en PRIA-UN.

Fuente: Creación Propia

En la figura 4-8, se ilustra el despliegue de los servidores de visión que procesan la información de las cámaras. Estos servidores se integran bajo la arquitectura de PRIA-UN, lo que permite un despliegue masivo y la conexión con bases de datos comunes para todos los robots conectados a la plataforma. La arquitectura planteada contempla la posibilidad de realizar el procesamiento de visión de máquina tanto internamente en la Raspberry Pi 4 como servidor de bajo costo, o externamente transmitiendo la información a otro servidor.

En la figura 4-9 se ve el desarrollo de la arquitectura para integrar bajo el esquema de PRIA-UN. Esta implementación está planeada para el despliegue masivo de servidores de visión que complementen las maquinarias y provean a un sistema de manufactura de la capacidad para la aplicación de visión maquina con la finalidad de obtener información de los componentes creados como objetos inteligentes alimentando tanto el motor de diseño generativo como el software MES.

La arquitectura planteada contempla una conexión con bases de datos que es común para todos los robots conectados a PRIA-UN y que se estandariza para ser en cualquier nuevo robot que se integre a la plataforma. La lógica del robot se desarrolla en un entorno de ROS el cual se define mediante nodos y tópicos. Dado que el procesamiento de visión de maquina se puede desarrollar tanto internamente en la Raspberry Pi 4 como servidor de bajo costo y externamente al transmitir la información a otro servidor, el bloque de servidor Web se encuentra en la mitad de la maquina de ROS.

El desarrollo realizado por Restrepo et al. [1], mostrado en la figura 4-10, se basa en el uso de imágenes sintéticas obtenidas de software CAD y su aplicación en el proceso de adquisición de fotos durante la fabricación de los componentes. Esto permite la clasificación de la familia de productos y la obtención de información valiosa para realizar seguimiento y control.

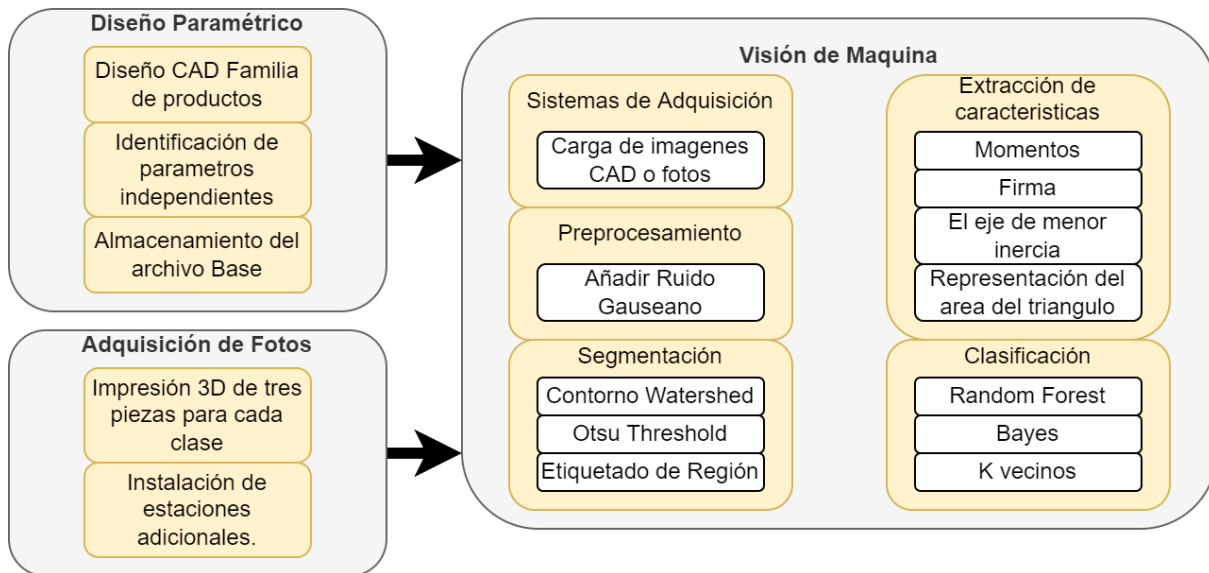


Figura 4-10.: Ejemplo aplicado del modelo de vision de maquina.

Fuente: Congreso CIMMX [1]

Este sistema entonces provee información para hacer seguimiento en un sistema de software MES como lo es la plataforma robótica industrial automatizada (PRIA-UN) y/o al motor de servicios de diseño generativo realimentado. La información transmitida no corresponde a fotografías, sino al tratamiento derivado por visión de maquina para ser aplicado en las diferentes etapas y diferentes aplicaciones.

Se resalta que este enfoque de visión de máquina puede complementarse con el uso de redes neuronales, que permiten describir modelos de sistemas y realizar clasificaciones más rápidas y precisas. Sin embargo, las redes neuronales requieren grandes cantidades de información etiquetada para su entrenamiento, lo que puede ser una limitación en ciertos casos. Con la

finalidad de ampliar su aplicación en trabajos futuros sistemas de etiquetado planteado en la subsección 4.4.1 puede extrapolarse para estas técnicas.

4.4.1. Obtención Imágenes CAD

Para el desarrollo de los modelos de aprendizaje supervisado, como los mencionados anteriormente, es requerido tener bases de datos con una gran cantidad de imágenes etiquetadas que permiten realizar el entrenamiento y reducir el error al momento de realizar la clasificación. Dado que se busca realizar un entrenamiento sin la necesidad de manufacturar las piezas, estas imágenes deben ser sintéticas que se obtienen por medio de software CAD.

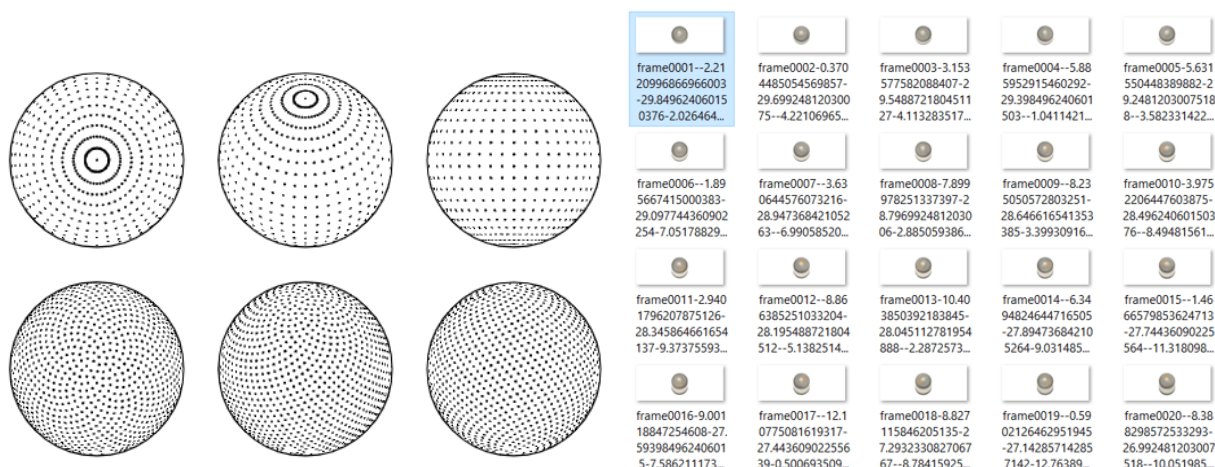


Figura 4-11.: Puntos de ubicación de cámara y etiqueta de resultados.

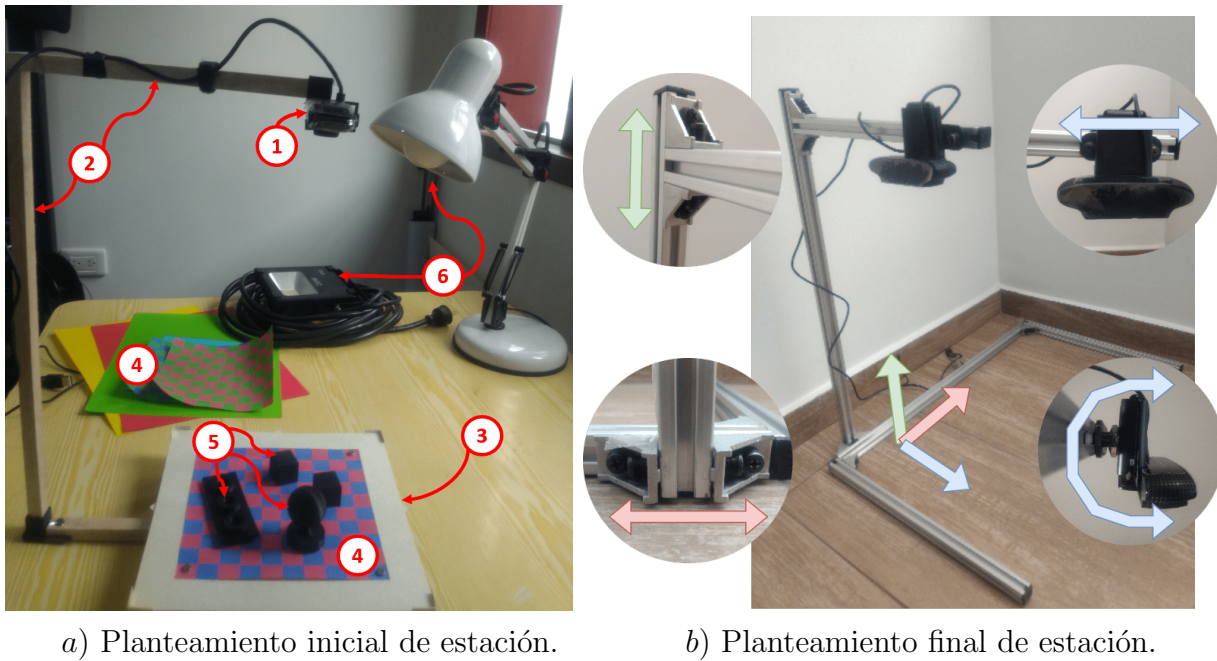
Fuente: Gonzalez [4] y Creación Propia

En la figura 4-11 se observa el proceso para obtener una base de datos de un componente por medio de API de software Fusion 360. En el artículo desarrollado por Gonzalez [4] se establecen las formulas para crear una esfera de fibonacci y ubicar la cámara, estas son usadas para obtener una medidas distribuidas de ubicación y etiqueta, las cuales se integraran a cada imagen respectivamente. Con el planteamiento de la esfera de fibonacci y como lo menciona Gonzales se tiene una mayor precisión en la medición de áreas de una esfera mientras se conserva una distribución uniforme en los puntos que la componen. Esta configuración para trabajar archivos CAD bajo el concepto de esfera y generar bases de datos es planteado por Huang et al. [87].

4.4.2. Obtención registro fotográfico

Para aplicar los modelos entrenados mediante imágenes sintéticas del software CAD, es fundamental contar con un sistema de adquisición de fotografías que sea flexible y adaptable

a diversas máquinas y entornos de trabajo. En esta sección, se plantea la implementación de estaciones de bajo costo basadas en servidores en tarjetas de desarrollo Raspberry Pi 4 con 4GB y 8GB de RAM, siguiendo la arquitectura mostrada en la figura 4-9.



a) Planteamiento inicial de estación.

b) Planteamiento final de estación.

Figura 4-12.: Planteamiento de estaciones para despliegue y ajuste en fabrica.

Fuente: Creación propia

En la figura 4-12.a se aprecia el desarrollo conceptual de la propuesta de adquisición de fotografías, en el cual se numeran los siguientes elementos: 1) Cámara web. 2) Estructura soporte de cámara. 3) Espacio de trabajo efectivo. 4) Fondos del espacio de trabajo. 5) Componentes a clasificar. 6) Iluminación complementaria.

En la evolución de la estación de adquisición de fotografías que se observa en la figura 4-12.b, se reemplaza la cámara web inicial por una cámara Logitech C922 Pro, enfocada en servicios de streaming, que puede proporcionar vídeos en full HD a 30fps. Esta cámara es reconocida por sistemas operativos basados en Linux, lo que permite su integración con Raspberry Pi y ROS. Además, se mejora la estructura utilizando perfiles de aluminio, lo que soluciona las deformaciones causadas por el peso de la cámara.

La nueva estructura es más flexible y cuenta con cuatro grados de libertad manuales, lo que permite ubicar la cámara en diferentes posiciones para adaptarse a las diversas estaciones del LabFabEx-UN. La estabilidad se logra mediante tres puntos de apoyo en la base inferior, sin invadir el espacio de trabajo de las fotos, como se aprecia en la figura 4-13.

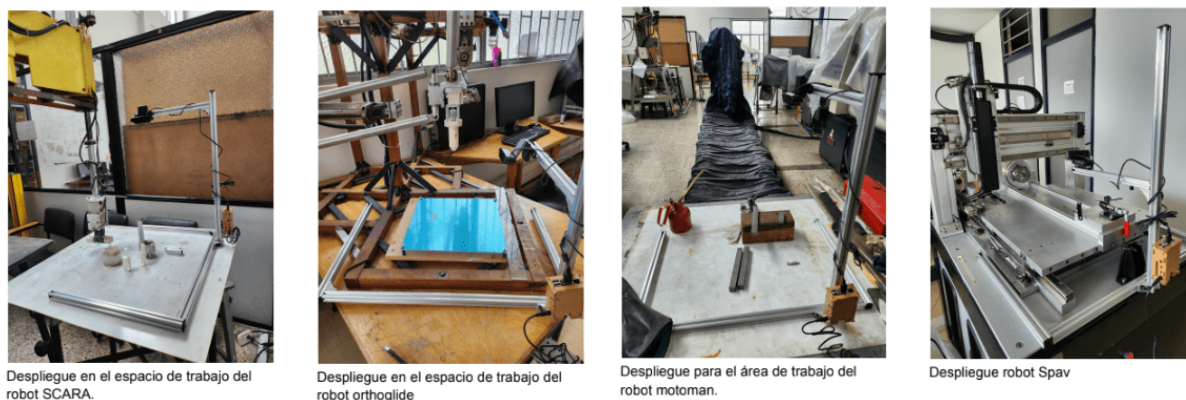


Figura 4-13.: Prueba de ubicación de la estación en diferentes espacios de trabajo del LabFabEx-UN.

Fuente: Creación propia

4.4.3. Identificación de características en componentes

Como se expreso al comienzo de esta sección la clasificación se puede realizar por medio de la agrupación de las respuestas de características o por medio de redes neuronales. Sin embargo, se va a desplegar como es que las características pueden dar información que ayuden a evaluar un componente por identificación de parámetros o inclusive primitivas del diseño generativo.

Debido a que no es la intención de este trabajo establecer parámetros para seleccionar un proceso de visión de maquina, se desarrollo el análisis funcional que provee el calculo de características para ser usado dentro del diseño generativo, por lo que la simplificación de un componente se realiza por medio del procesamiento de los bordes exteriores de cada uno en un proceso que se muestra en la figura 4-14. Para realimentar con características de bordes interiores al componente, técnicas de transformar la imagen de un plano cartesiano a un dominio polar-logaritmico como lo propone Hui et al. [88] son una buena propuesta, o como en el trabajo desarrollado por Oh et al. [89] donde identifican patrones dentro de componentes de la industria metal mecánica y su interacción con interfaces como un brazo robótico.

En la parte superior de la figura 4-14 se observa una etapa que al igual que en el trabajo con imágenes sintéticas de Huang et al. [87] se menciona que estas deben someterse a ruidos, para generar condiciones similares a las fotografías, debido que un análisis de contornos en imágenes sintéticas dan características continuas (perfectas), situación que no esta presente en la adquisición de material fotográfico.

En la parte intermedia de la figura se muestra el proceso para obtener una mascara gruesa de la zona donde es mas probable se encuentren solo los contornos exteriores de los componentes. La parte inferior muestra la aplicación del algoritmo Watershed para etiquetar contorno

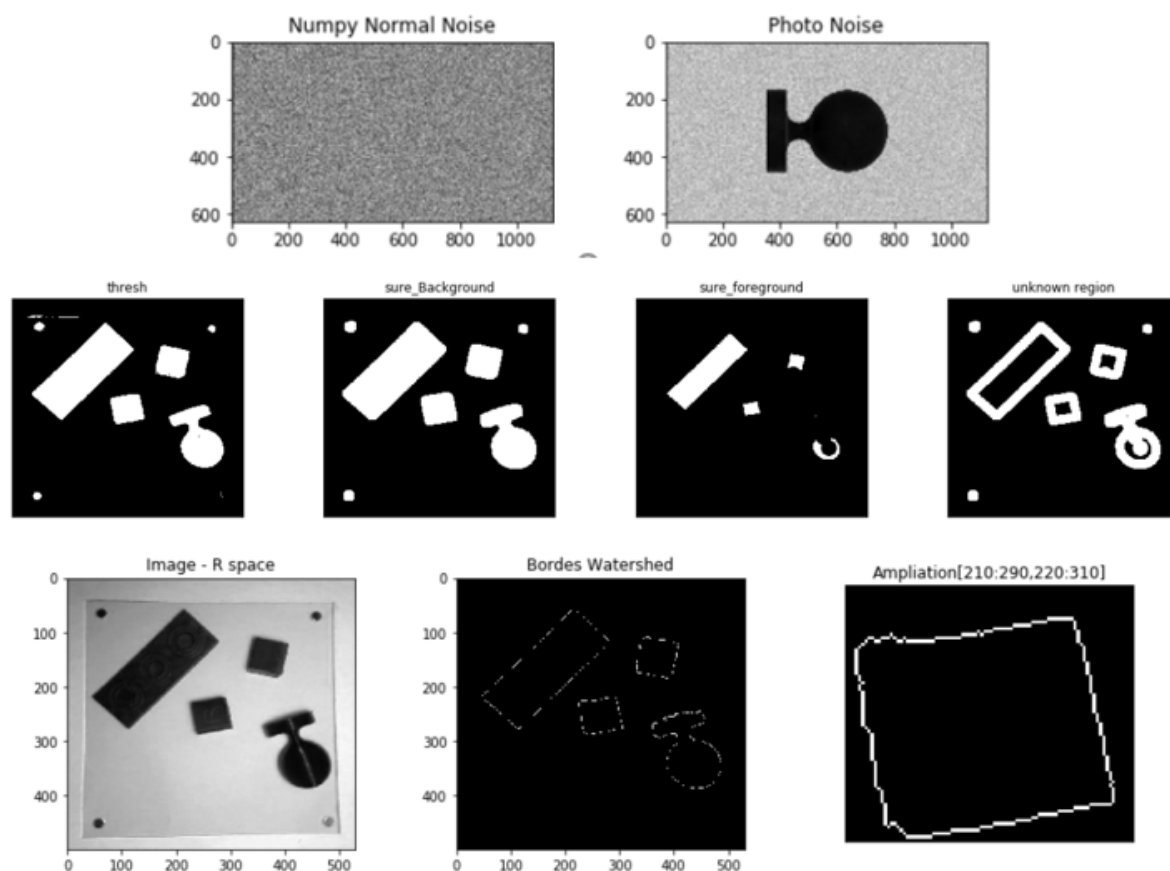


Figura 4-14.: Proceso de identificación de contornos en imágenes sintéticas y fotografías de componentes.

Fuente: Creación propia

por objeto, este algoritmo tiene una salida de interés debido a que cada etiqueta de un componente contiene el contorno con grosor de un píxel. En base a esta definición se puede contar la longitud del contorno (perímetro); obtener un centro de masas, debido a que todo el contorno tiene el mismo peso en el calculo; obtener el eje de menor inercia para identificar orientación; también obtener una cantidad discreta de longitudes del contorno respecto a este eje de inercia; respecto al eje de inercia se puede obtener la distancia mayor y la distancia menor del componente; también se pueden obtener las mismas distancias anteriores sin considerar el eje de inercia; entre muchas otras características.

Algunas de estas características están directamente relacionadas con la definición de parámetros en el diseño CAD o en la definición de MUCO en las estructuras de datos, por lo que la identificación de estos valores en el uso sirven para realimentar el sistema de diseño generativo. Estas características deben ser invariantes respecto al ángulo, pero ser variantes respecto al tamaño, por lo cual tener la medición de la cámara por medio de la estructura es una

técnica para la realimentación al sistema, aunque existen técnicas para realizar la calibración por medio de estructuras en el fondo de trabajo, siguiendo un patrón y ajustando la escala.

Dichas características se integran en el diseño generativo realimentado como respuesta al planteamiento de algún objetivo o restricciones del espacio búsqueda. Por lo cual el cálculo por medio de simulación provee valores que son comparados con los obtenidos previamente para determinar que el modelo de uso sea correcto y en caso negativo ajustarlo para tener una respuesta más cercana a la realidad. Es de resaltar que la visión de máquina es principalmente usado para procesos de evaluación debido a que no modifica el modelo, sino que realiza mediciones que alimentan sistemas de ajuste por error.

4.5. Transmisión de información a etapa de fabricación

Como se mencionó en la sección 4.2, en entornos de maquinaria CNC, la generación de código G para la manufactura puede realizarse automáticamente mediante el uso de diversos softwares, ya sean comerciales o de acceso libre, o de forma manual. Sin embargo, al establecer un entorno de realimentación, se busca que la respuesta sea estructurada para poder integrarse con la estructura de datos propuesta. De lo contrario, los datos tendrían que ser adicionados manualmente por el operario. Una forma de lograr esta semi-automatización es mediante la creación de macros de familias de productos, lo que permite llevar a cabo una automatización en términos de la ejecución de la manufactura.

En muchos casos, los softwares CAD se enfocan principalmente en el diseño, pero algunos de ellos también cuentan con funcionalidades para la fabricación asistida por computadora (CAM). Esto significa que facilitan la transición entre el modelado (ya sea mediante contornos o superficies) y la manipulación de estos modelos respecto a la maquinaria CNC, para generar el código G necesario. La complejidad de representar el comportamiento de la maquinaria CNC en respuesta a la obtención de la forma de un modelo, hace que las opciones para automatizar este proceso sean más limitadas, lo que puede implicar la utilización de software adicional.

Con el objetivo de establecer mecanismos actuales que complementen la simulación en etapas de diseño, se revisan opciones en software comercial y en una alternativa basada en modelos en nube de puntos. Estas opciones serán evaluadas para determinar su viabilidad en el contexto del diseño generativo y la realimentación en el proceso de fabricación. La elección de la mejor opción dependerá de factores como la complejidad del modelo, la disponibilidad de software y la integración con el flujo de trabajo establecido en la plataforma robótica industrial automatizada (PRIA-UN). La selección adecuada de estas herramientas permitirá una integración más efectiva entre el diseño y la manufactura, facilitando la transmisión de información y asegurando una producción más eficiente y precisa de los componentes diseñados.

4.5.1. En modelo de software comercial

Basado en software comercial la primera opción es que este incluya las estrategias CAM en el mismo software, por lo tanto la integración del modelado esta resuelto por el fabricante y cada modificación realizada al modelo se vera reflejada en la etapa CAM. Este resulta ser el caso donde se ofrece una alternativa en términos de automatización, sin embargo, en la manufactura por remoción de material los procesos de generación de operaciones requiere la intervención humana, por este motivo en este tipo de manufactura no es viable una realimentación a la estructura de datos de diseño generativo.

Es de resaltar que la mayoría de software CAM trabaja con tipo de archivos que modelan superficies y no contornos como si lo maneja nativamente el software CAD. Esta conversión de formato genera perdida de información del modelado original y por lo tanto se pierde procesos de realimentación inversa del proceso de manufactura al modelado o diseño.

En el caso de adición de material, y debido a la democratización de la tecnología 3d que lo hace accesible a cualquier persona sin necesidad de incurrir en violaciones de patentes, los desarrollos open source son una de las principales motores en este mundo, dado que muchas de las empresas que distribuyen esta maquinaria basan sus software en distribuciones open source, lo que genera uniformidad y fácil acceso.

Estos software CAM para impresión 3d han adoptado el tipo de archivo STL ligado a la estereolitografía (stereolithography) bajo concepto de prototipado rápido, por tal motivo este tipo de archivo describe el modelo como una superficie sin la descripción de color, texturas o otros atributos que si describen otros modelados de tipo CAD. Lo que genera un archivo demasiado simplificado del modelo y que además recrea la superficies por medio de triángulos.

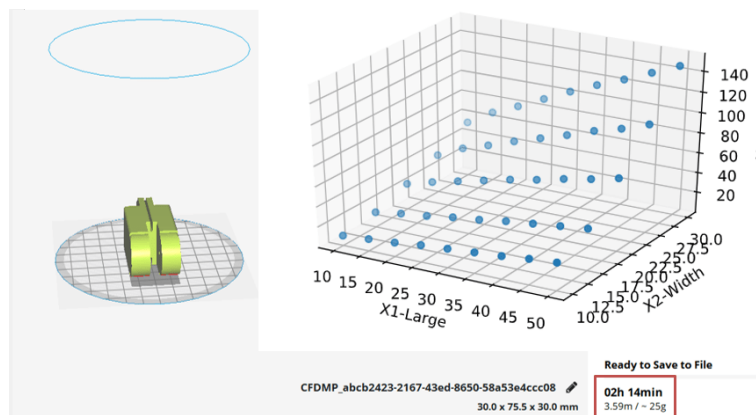


Figura 4-15.: Definición de una superficie de salida de parámetros de software CAM.

Fuente: Creación propia

Debido a que como se menciona no existe una realimentación del software CAM al software CAD, se establece la creación de nubes de puntos según parámetros de diseño donde responde a salidas de software CAM $\vec{X} = x_1, x_2, \dots, x_k$, como lo es el tiempo. Este tipo de consolidación de datos nos permiten por medio de procesos de interpolación multivariables obtener valores que pueden representar la salida y generalizar una realimentación para la etapa de diseño generativo, donde el proceso por el vecino más cercano o la interpolación n-lineal son alternativas que no tienen problema en este esquema.

En la figura 4-15 se observa como por medio del software CURA se pueden obtener los tiempos de manufactura por medio de impresión 3d, basado en dos parámetros principales del componente como lo es el largo y el ancho, dado que el análisis de diseño generativo se realiza en 2 dimensiones. Usando el diseño paramétrico y modificando los parámetros de largo entre 10 y 50 mm (restricciones del espacio de búsqueda) y el ancho entre 5 y 30 mm se obtienen los tiempos del software. Aunque esta es una ejemplificación de como se puede realimentar el motor de diseño generativo realimentado para evaluar objetivos según tiempo de manufactura, las salidas se pueden expresar también en términos de peso del componente en este ejemplo y software usado.

En el ejemplo se muestra una representación visual en la cual por experticia del diseñador puede tomar estas decisiones en un proceso convencional, al observar claramente que entre menor sea el largo y menor sea el ancho el tiempo de manufactura es menor. Pero al extrapolar este ejemplo en más características ajustables en parámetros del componente y en parámetros de manufactura (como el porcentaje y tipo de llenado interno), se vuelven decisiones que al no corresponder a una dimensionalidad menor o igual a 3 requiere de procesos automáticos de análisis de datos que pueden satisfacer esta experticia y obtener parámetros adecuados a satisfacer objetivos más complejos.

Esto abarca otra área del conocimiento como lo es el análisis de datos, en el cual se habla de reducción de dimensionalidad o correlación entre variables, en el cual se puede soportar mejor como el procesamiento de una persona frente a un problema con múltiples variables no responde como si lo hacen estas técnicas computacionales. En mecánica se van realizando múltiples comparaciones entre estos parámetros revisando las correlaciones y analizando las complejidades entre cada par de características funcionales, por lo que estamos hablando de otro tipo de reducción de dimensionalidad dando pesos a las comparaciones que a entendimiento del diseñador puede ser más influyente. Por lo tanto el poder establecer relaciones de varias variables en procesos directos de manufactura por medio de estructuras, permite automatizar y responder computacionalmente a las limitaciones en la toma de decisiones que tiene cualquier diseñador.

4.5.2. En programación de macros

La programación de macros es una opción manual de generación de código G, en esta se pueden identificar variables que van ligados a los conceptos de parámetros y que además responden a lógica de programación como declaraciones IF, WHILE, etc. Por lo tanto el código G responde inmediatamente a una familia de productos, las limitantes se observan en la capacidad para obtener valores simulados según la variación de parámetros, como lo son el cálculo de tiempos, pesos y demás; es necesario tener un simulador de código G genérico universal y en algunos casos un postprocesador de los controladores de esta maquinaria CNC. Tanto el software simulador como el postprocesador incurren a suscripciones o compra de los mismos lo que dificulta el poder estandarizar el proceso y Mipymes no pueden asumir dichos costos.

Por lo cual se concluye que aunque responde mejor a modificaciones a parámetros, debido a que ajusta el código G inmediatamente según variables de un producto, en la evaluación simulada no proporciona la información de salida necesaria para realimentar la evaluación de objetivos en el sistema de diseño generativo propuesto.

4.5.3. En modelo de nubes de puntos

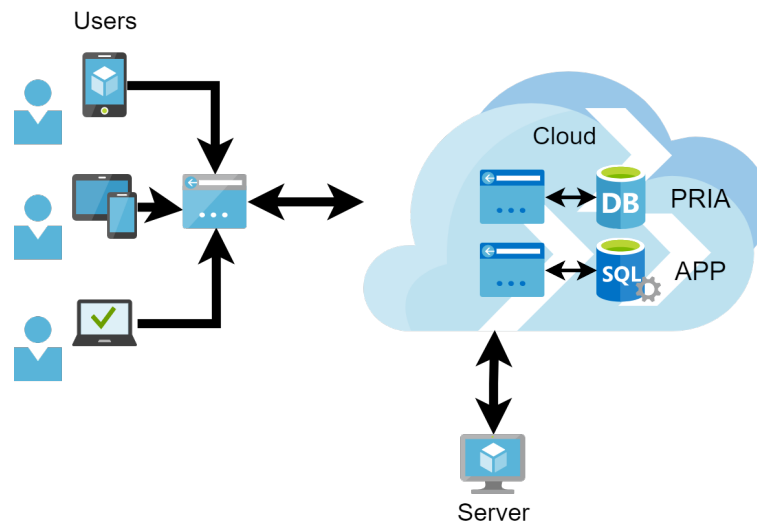


Figura 4-16.: Arquitectura de uso de generación de código G para remoción de material.

Fuente: Creación propia

La arquitectura propuesta se basa en uso de servidores remotos para explotar la capacidad computacional de las GPU, como se aprecia en la figura 4-16. Se observa que el usuario accede por medio de una aplicación web que comunica el frontend con el backend que almacena las estructuras en el software MES del LabFabEx, este usa una base de datos para documentos o

bases de datos adicionales para datos estructurados como la SQL. Adicionalmente se concibe la necesidad de un servidor para ejecutar la lógica basada en GPU para desarrollo del código G, aunque también puede considerarse un servicio de GPU.

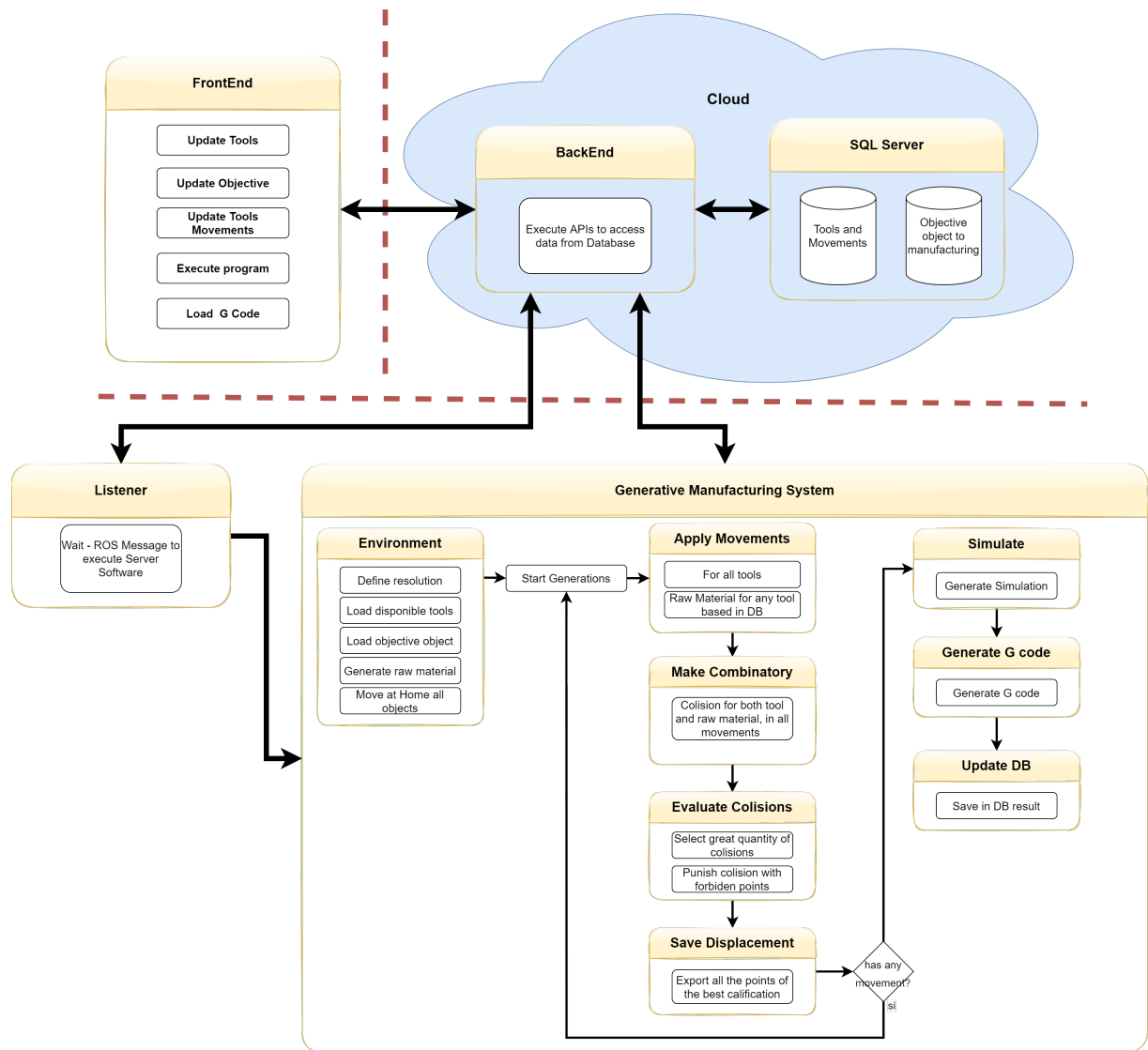


Figura 4-17.: Funcionalidad presente en los actores del sistema de generación de código G.
Fuente: Creación propia

Un desglose en detalle de lo que representa la arquitectura se aprecia en la figura 4-17, en el cual se puede apreciar las actividades que puede ejecutar el usuario por medio del frontend: Puede actualizar herramientas de maquinaria en nube de puntos. Actualizar el modelo en nube de puntos. Actualizar movimientos de maquina. Ejecutar el programa de generación.

El proceso del backend del servidor en la nube realiza la conexión con las bases de datos

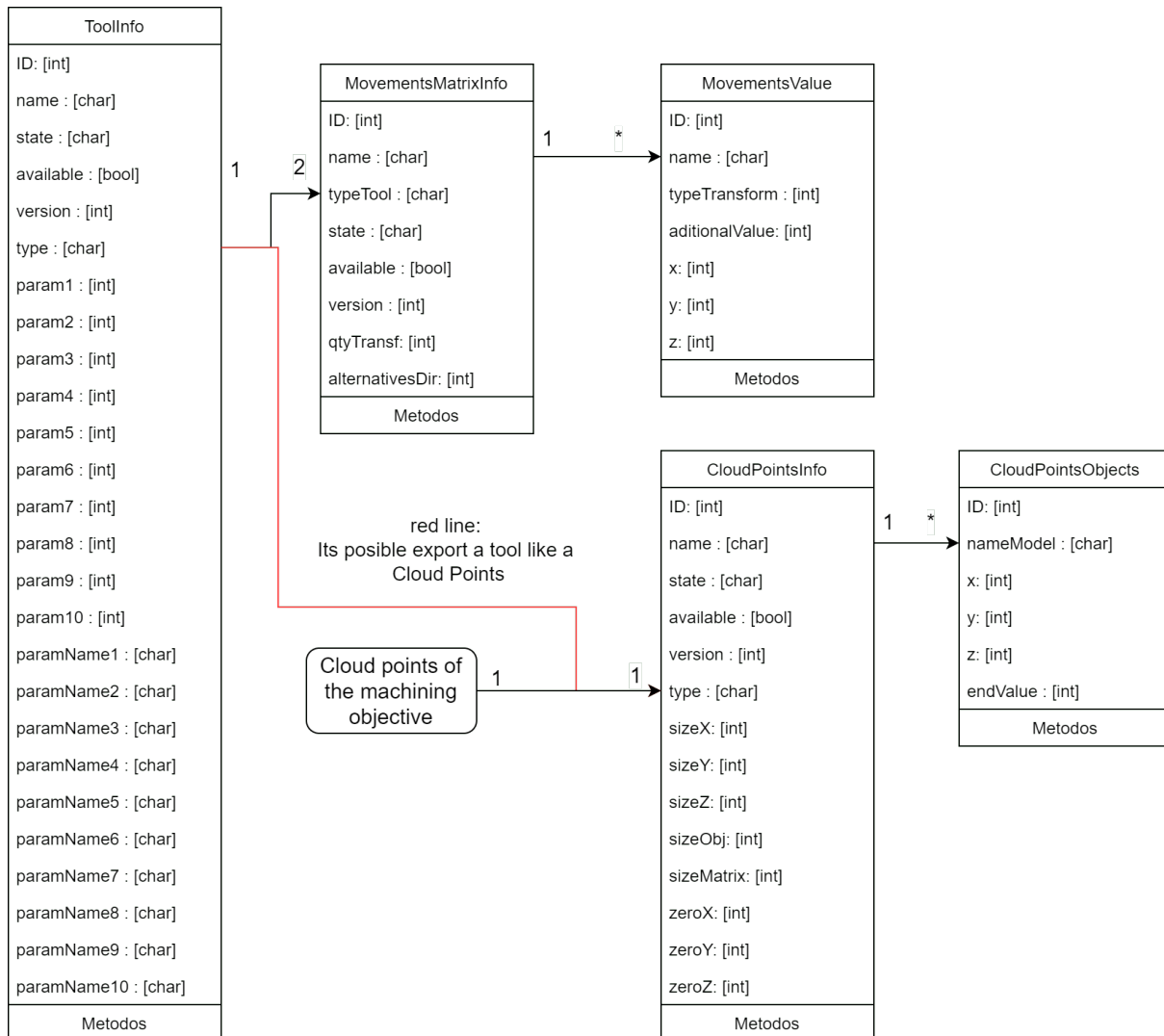


Figura 4-18.: Modelo de clases para el sistema de generación de código G.

Fuente: Creación propia

para actualizar los modelos en nube de puntos y ejecutar APIs para ejecutar la lógica en el servidor con capacidad de cómputo de GPU. Este servidor de lógica GPU cuenta con dos aplicaciones en las que la primera se encarga de actuar como un servidor de APIs, el cual al recibir un llamado va a ejecutar la lógica del programa. Para la ejecución de la lógica el servidor ejecuta la segunda aplicación que establece la conexión con el servidor de bases de datos para actualizar los modelos y demás estructuras que se observan en la figura 4-18.

La lógica de la segunda aplicación se expresa por medio de un flujo, en el cual las entradas se basan en cargar todos los modelos en nube de puntos definidos por el usuario y la salida es la generación de código G. Para obtener la salida el proceso se representa como un proceso iterativo que realiza colisiones de una herramienta (definida en nube de puntos), con un

material en bruto que trabaja la maquinaria CNC. Los movimientos de la herramienta y la material en bruto estan definidos por la arquitectura de la maquinaria CNC, por lo cual las transformaciones son realizados solo considerando vectores de movimientos.

El proceso de colisiones es donde se resalta la capacidad de GPU para realizar combinatorias punto por punto de las nubes de puntos. Por lo anterior la evaluación de mejores trayectorias se realiza en base a procesos de optimización donde se busca optimizar la cantidad de puntos con los que colisiona, pero tanto la herramienta como el material tienen identificados restricciones como puntos en los cuales no puede llevarse a cabo una colisión, también tiene limitados la cantidad de puntos máximos que puede colisionar. Por lo tanto se almacena como respuesta del proceso iterativo los vectores de movimientos de la herramienta y el material en bruto que obtengan la mejor evaluación y que al final se traduce en el código G.

Debido a que el componente objetivo ya se hereda del modelado, falta definir estructuras para vincular la herramienta en nube de puntos para el proceso de colisiones que se muestran en la figura 4-18. En este sentido una herramienta se puede almacenar en bases de datos de dos maneras: 1) Almacenando los parámetros de herramienta. 2) Almacenando la herramienta como una nube de puntos.

Para el primer caso se presenta una manera ligera de almacenar, en el cual en la lógica de programación generara las herramientas basados en sus parámetros. Es un método similar al presente en los software comerciales, en los cuales se tiene parametrizada una herramienta y el usuario debe definir los parámetros de la herramienta. La segunda opción de almacenar una herramienta implica mas espacio y debe ir acompañado por la clase "ToolInfo" debido a que es necesario otros parámetros adicionales para generar el código G, como pueden ser velocidades de husillo, avance de herramienta, entre otros parámetros que sean necesarios en futuros desarrollos.

Desarrollo de código G

En la generación del código G por medio de colisiones en nubes de puntos se establecen tres operaciones que se pueden llevar a cabo en la arquitectura del torno suizo del LabFabEx-UN: 1) Fresado en dirección del eje Y. 2) Fresado en dirección del eje Z. 3) Tronzado en dirección del eje X.

Adicionalmente se establece el tamaño de la barra del torno suizo en de material Nylon y diámetro de 25 mm. En el interior del material en bruto se define un cubo de 6mmX6mmX4mm el cual es el componente objetivo. Se utiliza una fresa de 4mm de diámetro y de 90 grados recomendados para mecanizado en plásticos de ingeniería. Las velocidades del husillo se recomiendan cercanas a las 10000RPM, pero debido a las limitaciones del torno suizo se usaran de 7000RPM.

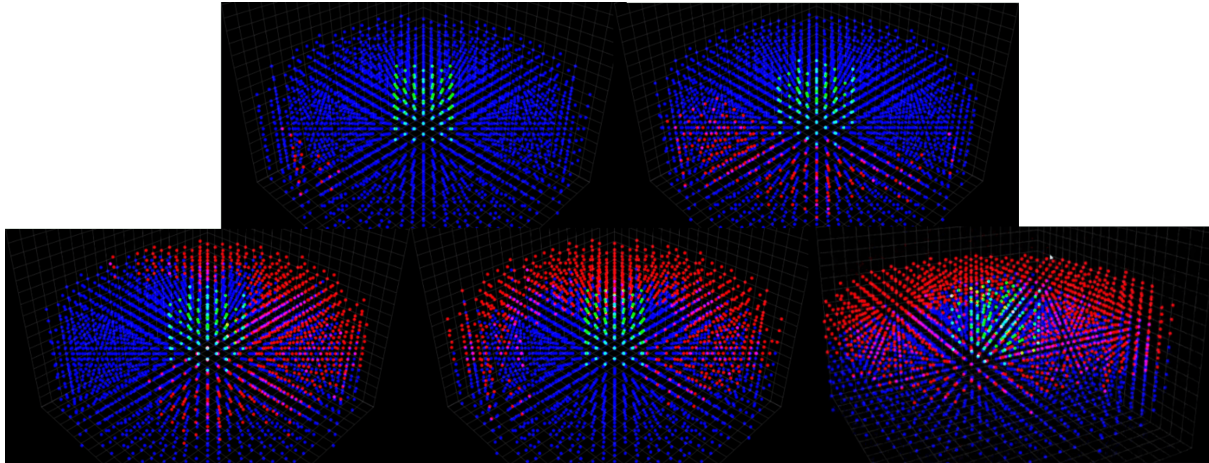


Figura 4-19.: Visualización colisión en nube de puntos de fresado en Y.

Fuente: Creación propia

Con los parámetros del ambiente y de las herramientas a utilizarse genera el proceso iterativo de generación de la trayectoria al generar las colisiones que se observan en la figura 4-19 para la herramienta en ataque por el eje Y y en la figura 4-20 para el ataque en el eje Z. En las figuras se puede apreciar imágenes tomadas de la secuencia para completar la operación, en el cual se nota la preferencia en los movimientos del giro en el eje C para evitar que la herramienta no pueda avanzar en movimientos laterales por los límites en la cantidad de material que puede remover.

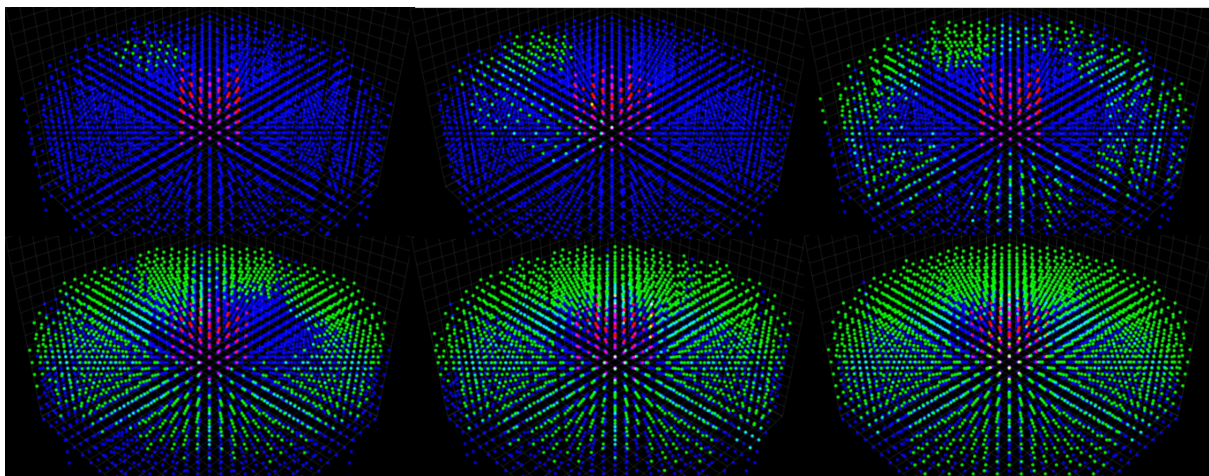


Figura 4-20.: Visualización colisión en nube de puntos de fresado en Z.

Fuente: Creación propia

El código G resultante se puede observar en el anexo C y D para el fresado en el eje Y el Fresado en Z respectivamente. El desarrollo de la operación de tronzado se puede apreciar en

la figura 4-21 con una dirección de avance de la herramienta en el eje X. Esta operación de tronzado debido al avance que tiene para generar el corte y la necesidad de que el material en bruto girara 360 para cada avance, supero las limitaciones de iteraciones generaciones establecidas de 2000, sin terminar de ejecutar el componente objetivo, por tal motivo se establece necesario en trabajos futuros replantear los procesos de torneado ante la no viabilidad de este planteamiento en el objetivo establecido.

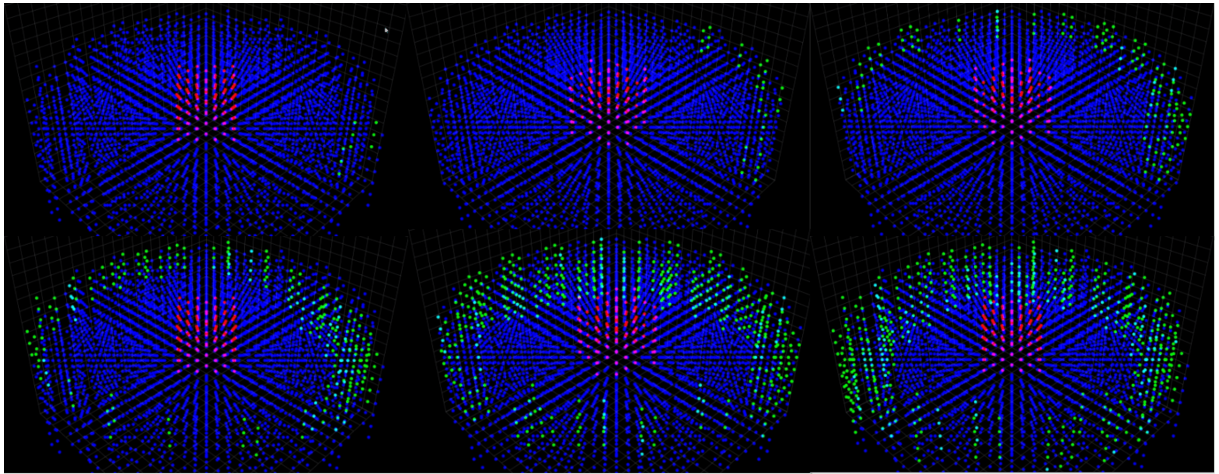
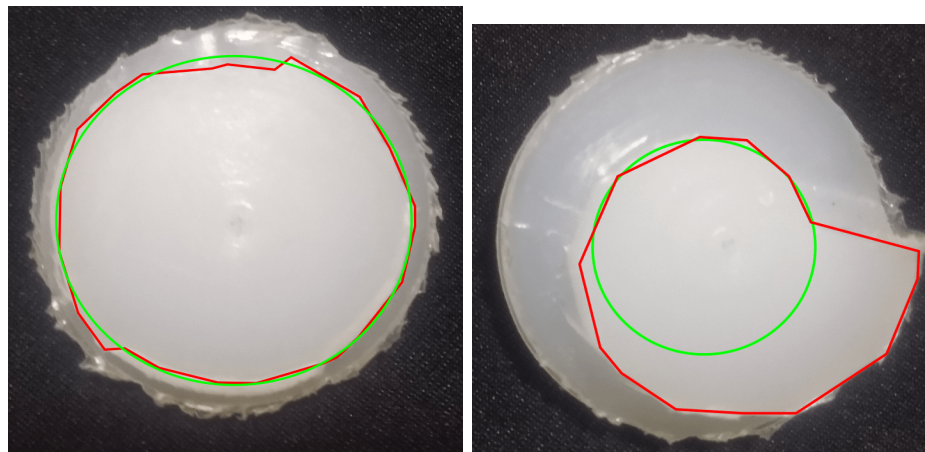


Figura 4-21.: Visualización colisión en nube de puntos de tronzado en X.
Fuente: Creación propia

Ejercicios prácticos

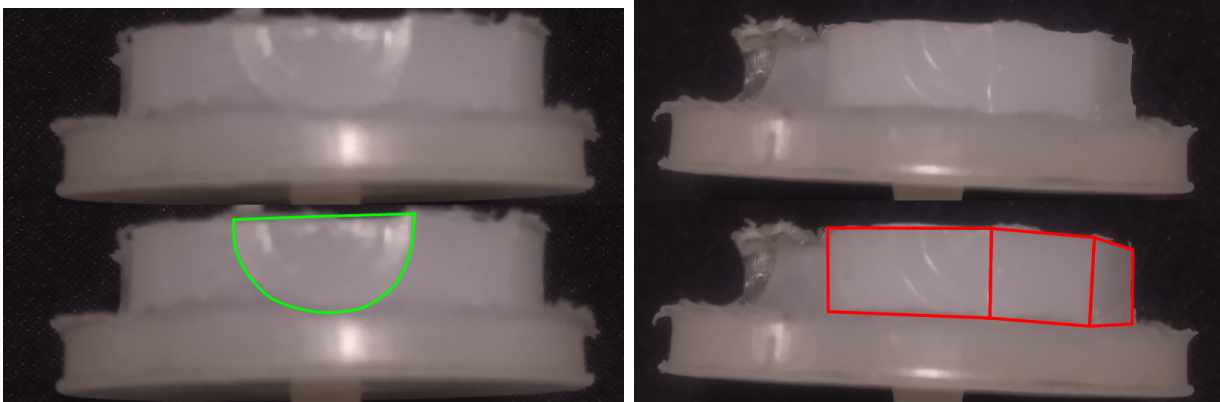


a) Resolución círculo de diámetro 20 mm. b) Avance sin limitaciones de remoción de material.

Figura 4-22.: Vista superior ejercicios prácticos.
Fuente: Creación propia

Para el desarrollo del primer ejercicio practico que se observa en la figura 4-22.a se establece un circulo de diámetro objetivo de 20 mm que se observa en color verde, en color rojo se observa el contorno obtenido por los movimientos de aproximación de la herramienta. La característica especial de este ejercicio es que no era necesario modificar la profundidad en el eje Z y los grados de libertad son en el eje X y Y, lo que se observa en la figura 4-23.a en la vista lateral en el cual la profundidad de mecanizado corresponde a la mitad del diámetro de la herramienta.

Debido a que la velocidad de avance se establece en 80mm/min, el tiempo de ejecutar 300 lineas de codigo G representan aproximadamente media hora. Lo anterior para responder ante cualquier propiedad de corte que se pueda presentar en el proceso como vibraciones, colisiones entre herramienta y material, etc. Para evitar tiempos prolongados en las pruebas se establecen círculos objetivos cercanos al del material en bruto. En la figura 4-22.b se observa un proceso que tuvo que detenerse debido a que la simulación no se aplico las restricciones de cantidad de material removido, medida que se establece necesaria y que las simulaciones realizadas lo incluye, lo anterior representa cortes con peligro de fractura herramental y/o calores excesivos en el plástico.



a) Resolución circulo de diámetro 20 mm. b) Avance sin limitaciones de remoción de material.

Figura 4-23.: Vista lateral ejercicios prácticos.

Fuente: Creación propia

Respecto al cumplimiento del objetivo las respuestas observadas se dan debido a la precisión de la nube de puntos la cual es de 1mm en los ejercicios planteados y los movimientos definidos que puede realizar la herramienta. La precisión mencionada se observa tanto en la figura 4-22.a y en mayor detalle en la figura 4-23.b en el cual se nota como la colisión se realiza en pasos constantes del eje C, ante una mayor precisión estos pasos sera menores pero es inversamente proporcional al tamaño de la nube de puntos. Movimientos que se originan naturalmente en el desarrollo practico son los de retracción de la herramienta, en

este caso para no remover material de la pieza objetivo y lo realiza antes de ajustar el paso en el eje C por tal motivo la figura 4-23.a se observa en detalle un avance y un retroceso de la herramienta y es por lo cual en la figura 4-22.a se observan unos cortes de mayor profundidad que del círculo objetivo establecido.

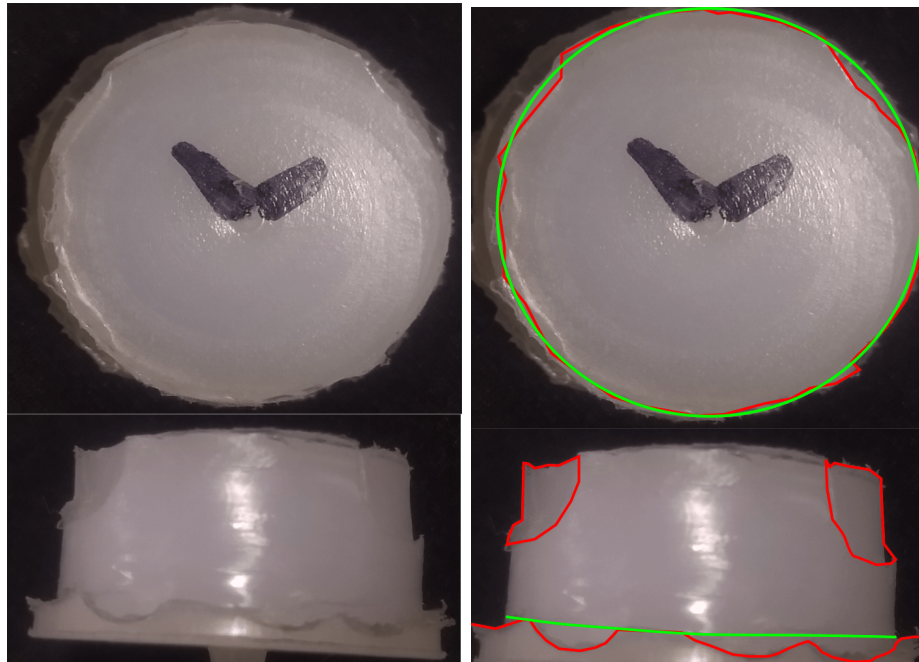


Figura 4-24.: Ejercicios prácticos de altura mayor al diámetro de herramienta.
Fuente: Creación propia

El ultimo ejercicio practico incluye un grado de libertad adicional, en el cual ya puede tomar decisiones de trayectoria en el eje Z y se observa en la figura 4-24. En este se observan las mismas características expresados en el ejercicio practico anterior pero adicionalmente se observan desfases por la precisión en los avances del eje Z, que en el primer ejercicio al estar fijo no ocurren.

Estos ejercicios prácticos dan muestra que es posible por medio de un planteamiento ejecutar secuencias de mecanizado basados en el modelamiento de componentes por nube de puntos, sin embargo, y dado que no es el objetivo del trabajo generar un mecanizado generativo por nube de puntos, se ejemplifica la posibilidad de integrar modelados por nube de puntos en procesos automatizados, dado que la estructura propuesta incluye un segmentos de herramientas y la planeación de trayectoria se hace sin necesidad de la intervención del diseñador. Para un trabajo futuro desarrollar este tipo de herramientas que también explotan la principal característica del procesamiento de GPU, y es realizar una gran cantidad de operaciones sencillas y de poco tiempo. Igualmente se observa características que pueden brindar una

solución entre la realización de código G generado manualmente y los software comerciales y a la articulación entre la academia y la industria.

5. Caso de estudio desarrollo del Gripper

Para el desarrollo del caso de estudio, se ha definido una metodología para ser aplicada en el LabFabEX-UN que se presenta en la figura 5-1. Este flujo de trabajo está diseñado para abordar la interacción entre el diseño y la manufactura, tal como se plantea en la figura 2-7 de la metodología de esta tesis de maestría. El flujo se divide en tres etapas principales:

1. Definición de características de diseño: En esta etapa se establecen las características necesarias para el diseño del producto.
2. Esquema de diseño generativo realimentado: En esta etapa, se lleva a cabo el desarrollo y la evaluación del sistema mediante el uso de la estructura de datos basada en primitivas y la técnica multiobjetivo NSGA-II. Se busca obtener soluciones generativas que se retroalimenten y mejoren a lo largo del proceso.
3. Integración con la etapa de manufactura: En esta etapa final, se lleva a cabo la integración del diseño generado con el proceso de manufactura. Se obtienen los consumibles necesarios para la fabricación del producto.

Además, cabe destacar que el flujo de trabajo contempla la interacción del diseñador en el sistema, representada en color amarillo, la ejecución automática del sistema en verde, y un proceso que puede ser automático o requerir interacción del diseñador, dependiendo de la capacidad tecnológica de cada fábrica, representado en morado.

5.1. Desarrollo del proceso propuesto

En la primera etapa del proceso, se realiza la configuración inicial del ambiente. Esto implica cargar la información de objetos previamente fabricados y los requerimientos del sistema, estableciéndolos como objetivos a cumplir.

La segunda etapa se enfoca en la resolución del flujo convencional, pero con la incorporación de operaciones adicionales que mejoran la capacidad de cómputo del sistema. Estas operaciones se diseñan para optimizar el proceso de resolución y garantizar un rendimiento eficiente.

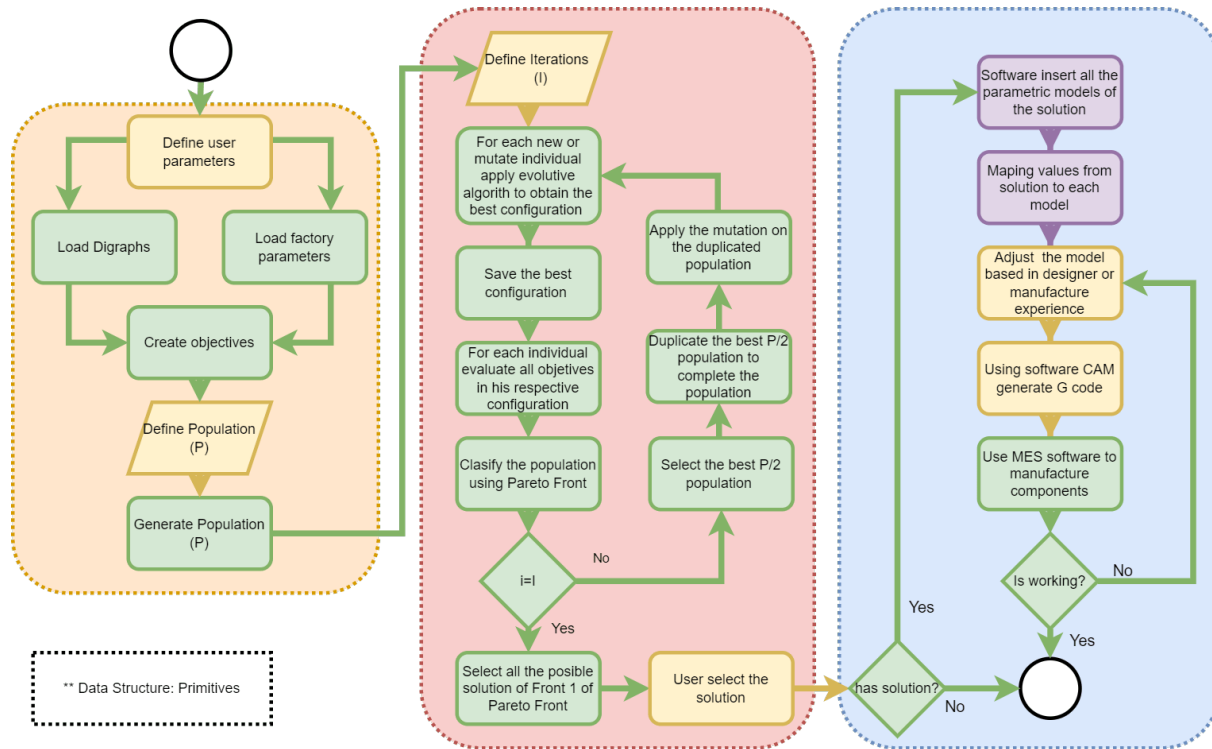


Figura 5-1.: Estructura planteada de diseño generativo para resolver el diseño del Gripper.
Fuente: Creación propia

La última parte del proceso es más variada, ya que depende de la capacidad de la fábrica y de la integración con el software MES (Manufacturing Execution System). En esta etapa, se transmite la información generada por la solución seleccionada a los procesos de fabricación. El objetivo es simular el comportamiento real de la manufactura del producto y minimizar la necesidad de iteraciones en esta etapa.

Es importante tener en cuenta las especificaciones de desarrollo tecnológico de cada fábrica, ya que este factor puede variar en cada una de ellas. Por lo tanto, el proceso también puede ser configurado en función de los recursos disponibles. Para ello, se consideran las alternativas presentadas en la figura 4-1 y la figura 4-7, tanto para la realimentación como para el desarrollo integral del producto (ver Sección 4.3).

Además, la selección de parámetros como la población, las iteraciones y los objetivos a resolver, dependen de la capacidad computacional disponible. En el caso del Gripper, se utiliza un servidor personal con una tarjeta gráfica NVIDIA GeForce GTX 1050 Ti, que cuenta con una memoria de 4 GB GDDR5. En la Tabla 5-1, se detallan otras propiedades relevantes para el planteamiento y desarrollo de los componentes del Gripper.

Dado que el enfoque de resolución propuesto se basa en programación paralela, pero solo en la dimensión X, se puede crear una cuadrícula de hasta 2,147,483,647 casillas. Cada cuadrícula

se divide en bloques de 1,024 y cada bloque puede ejecutar un máximo de 1,024 hilos. En cada bloque, se puede compartir una memoria máxima de 65 KB.

Tabla 5-1.: Especificaciones CUDA tarjeta NVIDIA Geforce GTX 1050 Ti.

| Especificación | Valor |
|-----------------------------|--------------|
| Max Threads Per Block | 1024 |
| Max Block Dim X | 1024 |
| Max Block Dim Y | 1024 |
| Max Block Dim Z | 64 |
| Max Grid Dim X | 2147483647 |
| Max Grid Dim Y | 65535 |
| Max Grid Dim Z | 65535 |
| Max Registers Per Block | 65536 |
| Max Shared Memory Per Block | 49152 |

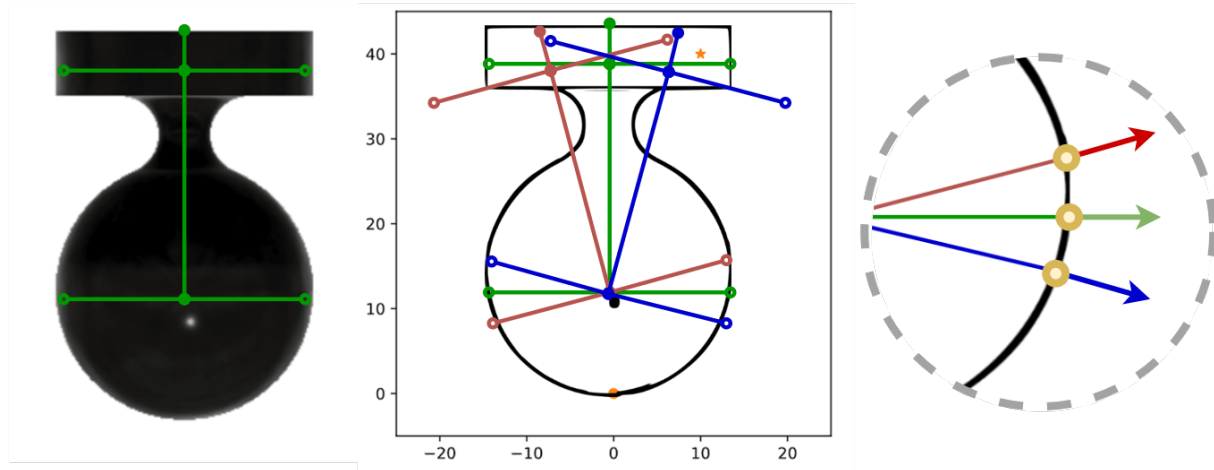
5.1.1. Parámetros de usuario

En el planteamiento del problema, se busca desarrollar un Gripper que permita manipular la orientación de un objeto en procesos de ensamblaje robotizados. En la figura 5-2.a, se muestra el componente a manipular junto con una descripción del modelo representado por MUCOs en un grafo direccional. Esta representación se basa en la experiencia y conocimiento del diseñador. El grafo identifica dos puntos de sujeción en la parte central del círculo, así como tres puntos de salida en la parte rectangular, los cuales permiten la interacción del componente con el entorno.

En el proceso, el usuario genera el grafo representado en la figura 5-2.a y la rotación de la estructura se realiza operando el primer MUCO del grafo en la estructura de datos. Esta operación se muestra gráficamente en la figura 5-2.b. Debido a la simetría del componente a manipular y con el objetivo de reducir la cantidad de objetivos, en la figura 5-2.c se presentan tres puntos de sujeción, acompañados de tres vectores normales que indican la orientación deseada del contacto. Estas características son las que el usuario representa como objetivos que se desglosan en la sección 5.1.4.

5.1.2. Cargar los grafos direccionales

En el contexto de capturar la experiencia del diseñador mediante estructuras basadas en primitivas, es posible almacenar y transmitir estos conceptos para la interacción y creación de nuevos productos. En este caso, se han heredado conceptos desarrollados a través de



a) Objeto con descripción de un grafo de MUCOs. b) Representaciones de grafos respecto orientaciones. c) Definición de objetivos.

Figura 5-2.: Definición del objeto a manipular.

Fuente: Creación propia

métodos tradicionales en el diseño del Gripper, los cuales se detallan en los H, I y J. Estos desarrollos se basaron en un proceso que emplea estructuras de datos y modelos matemáticos en el diseño generativo para generar la primera versión del sistema.

Se propone la utilización de dos tipos de grafos que representan un eslabón y una pieza de contacto general, los cuales alimentarán el sistema. Estos componentes se basan en un modelo previamente parametrizado en un software CAD (Fusion 360), lo cual permite la interacción para encontrar soluciones del componente.

Eslabón

El primer tipo de grafo direccional corresponde al eslabón, el cual es el concepto fundamental de un único nodo MUCO en la estructura basada en primitivas. Sin embargo, se observa en la figura 5-3 que agregar más nodos MUCO al grafo permite un modelado más complejo del grafo o componente. La figura 5-3.a muestra el caso más simple en el que solo interactúa con un único MUCO, mientras que la figura 5-3.b presenta una representación con complementos en el modelado, donde los MUCOs adicionales permiten más puntos de contacto para la evaluación de objetivos, lo cual representa características del componente como el grosor del eslabón.

En la figura 5-3.c se observa un MUCO de mayor complejidad que se ha desarrollado a partir de varias iteraciones y del análisis funcional de los tensores, como se describe en el anexo H. Este MUCO incluye puntos de sujeción en el interior del cuerpo para aplicar el efecto de accionamiento mediante tensores. Como respuesta al planteamiento de los grafos y

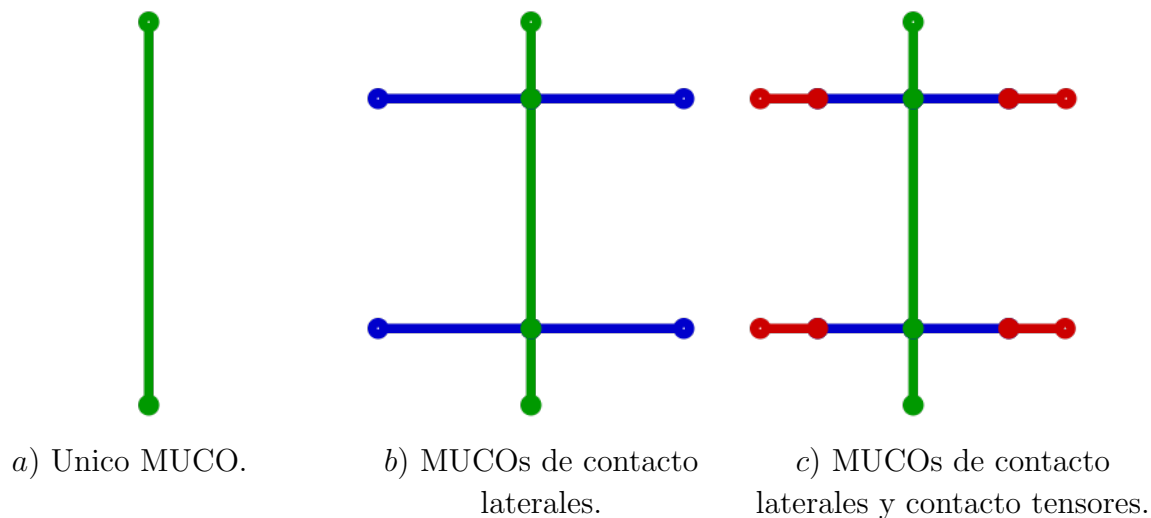
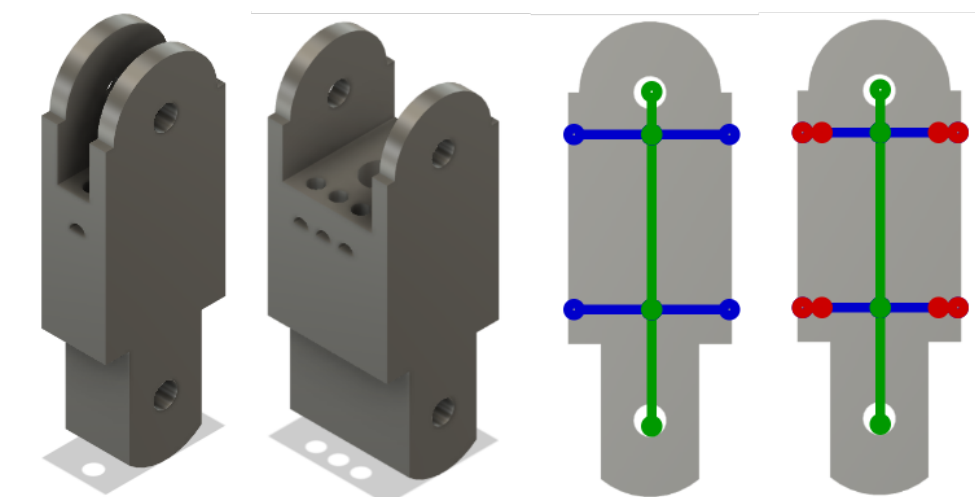


Figura 5-3.: Representación de distintos grafos para un eslabón.

Fuente: Creación propia

al conocimiento previo sobre un eslabón accionado por tensores, se muestra en la figura 5-4 un modelo paramétrico del eslabón creado con el software Fusion 360.

En la figura 5-4.a se resalta cómo el diseñador incluye características de acople en el eslabón para asegurar la conectividad de las entradas y salidas del modelo en respuesta a interfaces rotacionales. Por otro lado, en la figura 5-4.b se muestra cómo el modelo responde a los valores obtenidos mediante la estructura de datos, manifestándose en el grosor del eslabón y su interacción con el objetivo inicial.



a) Según posición del eslabón.

b) Respuesta 2d a la estructura del grafo.

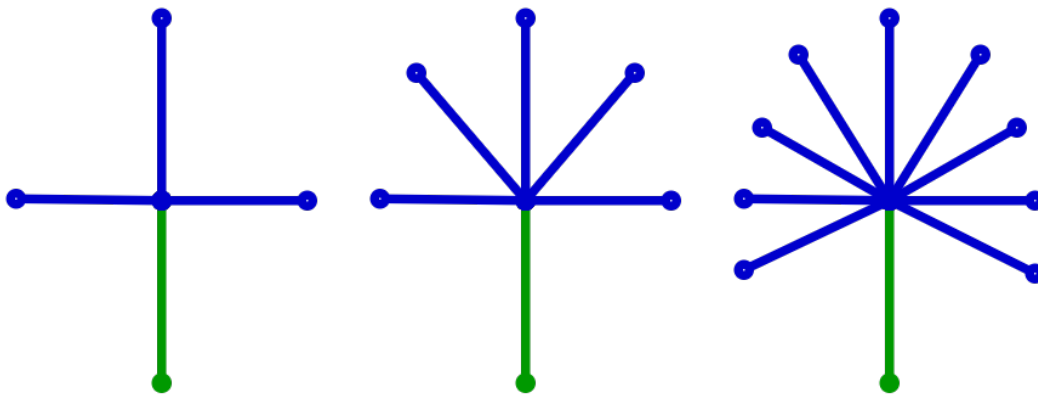
Figura 5-4.: Representación paramétrica de un grafo en el modelo CAD.

Fuente: Creación propia

Dado el planteamiento del problema, los tensores no son un componente a modelar, por lo tanto, se selecciona el grafo de la figura 5-3.b como base para el desarrollo del proceso iterativo. Aunque en el modelo paramétrico se cuenta con un valor para ubicar los tensores, será el diseñador quien defina el valor al momento de la fabricación.

Componente de contacto general

Como complemento a los eslabones y con el objetivo de mejorar la evaluación de los objetivos relacionados con la orientación del contacto, se realiza un desglose bio-inspirado de la yema del dedo en el anexo H. Esta bio-inspiración, basada en la geometría de la yema del dedo humano, permite mejorar el contacto fino. Se representa esta idea mediante diferentes grafos direccionales, como se muestra en la figura 5-5. A medida que aumenta la cantidad de nodos MUCO en el grafo, se incrementan las opciones de contacto, lo que se convierte en una alternativa para aumentar la cantidad de soluciones posibles.



a) Tres opciones de contacto. b) Cinco opciones de contacto. c) Nueve opciones de contacto.

Figura 5-5.: Grafos direccionales para el componente de contacto general.

Fuente: Creación propia

En términos del material utilizado, se selecciona un caucho de silicona con una dureza Shore A - 25 para el componente de contacto general. Dado que este componente es una conexión rígida auxiliar a los eslabones, no permite la conectividad de salida en los MUCOs del grafo para añadir otro eslabón. Por lo tanto, la conexión de este grafo en el sistema determina el tamaño de la cadena cinemática al identificar este componente como el final de la misma.

La conexión rígida implica que se debe desarrollar una interfaz diferente a la planteada para la interacción entre los eslabones. En la figura 5-6, se muestra cómo se utiliza el modelado del eslabón para interactuar con el modelo paramétrico del componente de contacto general y realizar operaciones de corte. Por lo tanto, este componente paramétrico muestra los

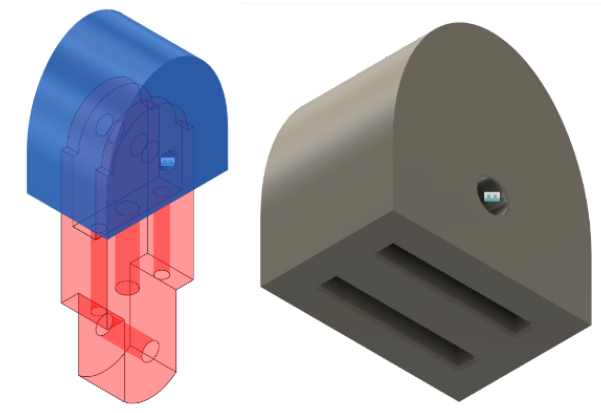


Figura 5-6.: Interfaz rígida de componente de contacto general.

Fuente: Creación propia

mismos comportamientos de respuesta que la interacción entre eslabones, ya que el diseño paramétrico se basa en los parámetros definidos en el eslabón y responde rápidamente a esta conectividad. La parametrización se basa en una elipse, teniendo en cuenta el diámetro mayor y menor.

Con lo anterior, se garantiza la transmisión de conceptos previamente creados y adaptados, como los eslabones. Además, se pueden integrar conceptos no definidos previamente en el sistema, asegurando que el planteamiento se realice desde la representación del modelo en un sistema CAD para que sea coherente con las etapas del ciclo de vida del producto en la fábrica.

5.1.3. Cargar parámetros de fabrica

En esta sección, se abordará la carga de parámetros relacionados con la configuración y operación de la fábrica. Estos parámetros son esenciales para alimentar el sistema de información sobre la distribución de maquinaria, la cantidad de herramientas y otros aspectos relevantes. Además, se enfoca en la filosofía de la fábrica, específicamente en el caso del LabFabEx-UN, que busca cumplir con métricas de flexibilidad.

Para determinar la complejidad de los métodos de fabricación, se asigna un valor basado en la experiencia del diseñador. Esto implica considerar mediciones cuantitativas y datos cualitativos proporcionados para cada proceso. Los criterios utilizados para establecer la complejidad de los procesos de fabricación incluyen:

1. Número de componentes: calculado a partir de los grafos direccionales.
2. Forma: calculado a partir de la cantidad de los nodos MUCO.

3. Disponibilidad tecnológica.
4. Complejidad tecnológica.
5. Tolerancia.
6. Tiempo de ciclo de fabricación.

Dentro de la integración del LabFabEx-UN, se consideran los siguientes procesos de fabricación, así como otros procesos adicionales que pueden adquirirse localmente y ser evaluados en el proceso de diseño generativo:

- Manufactura aditiva.
- Manufactura sustractiva.
- Corte láser.
- Moldes de inyección.

Estos procesos de fabricación representan una amplia gama de opciones disponibles en la configuración de la fábrica y serán utilizados en la generación de soluciones a través del diseño generativo. La selección adecuada de estos parámetros de fábrica depende de la evaluación realizada por el algoritmo en el proceso de fabricación de los productos diseñados.

Con el objetivo de analizar la influencia de las métricas de flexibilidad y la complejidad del producto, se realiza inicialmente la resolución del sistema utilizando los valores del escenario actual del LabFabEx-UN. Posteriormente, se plantean cambios en el escenario del laboratorio con el fin de comparar la influencia de estas variaciones en el desarrollo del mismo producto, en este caso, un gripper. Este enfoque permite evaluar cómo las modificaciones en los parámetros de fábrica afectan la generación de soluciones y proporciona información valiosa para la toma de decisiones en el diseño y la configuración de la fábrica.

Escenario actual del LabFabEx-UN

En el contexto de esta tesis de maestría, se aborda el análisis de los procesos de manufactura utilizados en el LabFabEx-UN. Los procesos de manufactura son fundamentales en la producción de componentes y productos, y su elección adecuada puede tener un impacto significativo en la calidad, la eficiencia y la viabilidad económica de los proyectos.

En primer lugar, se analiza la Disponibilidad Tecnológica de cada proceso de manufactura, considerando criterios cualitativos de ubicación, puesta en marcha, material disponible, entre otros. El respectivo análisis se describe a continuación:

- **Manufactura Aditiva:** Esta tecnología se encuentra disponible en el LabFabEx, y cuenta con una ubicación física adecuada. Además, se encuentra lista para su puesta en marcha, con la capacidad de encenderse, alimentar el material necesario y transmitir el código G. Asimismo, se dispone de material PLA para su utilización. Otra ventaja que presenta es que en caso de fallo localmente es posible adquirir estos servicios.
- **Sustractiva CAM:** Este proceso también está disponible en el LabFabEx-UN por medio de un torno suizo y requiere el uso de un software CAM para generar el código G. Aunque se dispone de una licencia en Mastercam, no se cuenta con un postprocesador específico. Sin embargo, se ha verificado que es posible establecer la configuración de herramientas y materiales, así como transmitir el código G requerido para la puesta en marcha. El material es Nylon 6 para todos los procesos de manufactura sustractiva.
- **Sustractiva Manual:** En el LabFabEx-UN, se dispone de un torno suizo que puede ser operado manualmente sin necesidad de utilizar código G. Este proceso se encuentra listo para su puesta en marcha, y se cuenta con los materiales adecuados para llevarlo a cabo.
- **Sustractiva Nube de puntos:** Este proceso se basa en un algoritmo desarrollado en esta tesis, que permite generar el código G a partir de un componente descrito en una nube de puntos. Aunque se encuentra en fase de desarrollo, se ha establecido que la puesta en marcha implica la conversión de modelos CAD a nube de puntos objetivos.
- **Corte Láser:** Los laboratorios de la Universidad Nacional de Colombia disponen de la tecnología de corte láser, la cual se encuentra disponible. Sin embargo, se debe tener en cuenta el costo asociado a la adquisición del servicio. En cuanto a los materiales, se dispone de láminas metálicas, acrílico y MDF.
- **Moldes de inyección:** A diferencia de los demás procesos, los moldes de inyección no están disponibles en el LabFabEx-UN. En su lugar, se debe recurrir a terceros para adquirir este servicio, lo cual implica un costo adicional. Asimismo, se cuenta con materiales sintéticos adecuados para su uso en frío o caliente.

A continuación, se presenta una tabla que resume la Disponibilidad Tecnológica de cada proceso de manufactura:

Tabla 5-2.: Disponibilidad tecnológica en el LabFabEx-UN.

| Proceso de manufactura | Disponibilidad Tecnológica |
|--|-----------------------------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 2 |
| Manufactura sustractiva – Manual | 2 |
| Manufactura sustractiva – nube de puntos | 2 |
| Corte láser | 4 |
| Moldes de inyección | 5 |

Además de la Disponibilidad Tecnológica, se evalúa la Complejidad Tecnológica de cada proceso de manufactura. Para ello, se consideran los siguientes criterios de la generación de documentos de manufactura y la dificultad de uso.

- **Manufactura Aditiva:** Para este proceso, se utiliza un generador de código G open source, que es capaz de leer distintos tipos de archivos CAD. En cuanto a la dificultad de uso, permite su implementación sin importar el programa de CAD utilizado, lo cual facilita su adopción.
- **Sustractiva CAM:** En este caso, no se dispone de un postprocesador específico para generar el código G. Sin embargo, se requiere el uso de un software CAM y se cuenta con una licencia en Mastercam. La dificultad de uso está asociada al manejo de dicho software y las limitantes en movimientos que genera el no poder traducir los movimientos del software CAM a código G.
- **Sustractiva Manual:** La generación de documentos de manufactura se realiza mediante la creación de macros y parámetros, los cuales pueden ser extendidos por un operario. Aunque se puede probar en vacío, los sistemas de validación son complejos y requieren análisis adicionales.
- **Sustractiva Nube de puntos:** Este proceso se basa en un algoritmo en desarrollo, el cual necesita estabilizarse. La generación de documentos de manufactura implica la conversión de modelos CAD a nubes de puntos objetivos. En cuanto a la dificultad de uso, requiere conocimientos específicos en la conversión y manipulación de las nubes de puntos.
- **Corte Láser:** El software CAD utilizado permite generar los planos necesarios para la fabricación mediante corte láser. La dificultad de uso está relacionada con el conocimiento en la generación de dichos planos de corte.

- **Moldes de inyección:** La generación de documentos de manufactura para este proceso implica la fabricación del molde utilizando otras técnicas de manufactura. La dificultad de uso depende del material utilizado y puede requerir análisis adicionales.

A continuación, se muestra una tabla que resume la Complejidad Tecnológica de cada proceso de manufactura:

Tabla 5-3.: Complejidad Tecnológica en el LabFabEx-UN.

| Proceso de manufactura | Complejidad Tecnológica |
|--|--------------------------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 4 |
| Manufactura sustractiva – Manual | 3 |
| Manufactura sustractiva – nube de puntos | 5 |
| Corte láser | 3 |
| Moldes de inyección | 5 |

Además de los criterios cuantitativos y cualitativos previamente mencionados, se añadirán dos medidas cualitativas adicionales para evaluar los procesos de manufactura en el LabFabEx-UN: Tolerancias y Tiempo de ciclo de fabricación.

Las Tolerancias se refieren a la precisión y exactitud con la cual los procesos de manufactura son capaces de producir componentes dentro de las especificaciones requeridas. En la mayoría de los procesos que utilizan control numérico, como la manufactura aditiva y sustractiva CAM, la tolerancia suele ser alta debido a la precisión de los códigos utilizados. Sin embargo, en el caso de la manufactura sustractiva basada en nube de puntos, la resolución de los puntos que describen el objeto puede afectar la tolerancia obtenida. Para evaluar este criterio, se utilizará una escala del 1 al 5, donde 1 indica una alta tolerancia y 5 indica una baja tolerancia. A continuación, se presentan la tabla que resume las Tolerancias:

Tabla 5-4.: Tolerancias de los procesos de manufactura en el LabFabEx-UN.

| Proceso de manufactura | Tolerancias |
|--|--------------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 1 |
| Manufactura sustractiva – Manual | 1 |
| Manufactura sustractiva – nube de puntos | 3 |
| Corte láser | 1 |
| Moldes de inyección | 1 |

El Tiempo de ciclo de fabricación se refiere a la duración total del proceso de manufactura para producir un componente o producto. Este criterio se normalizará utilizando como referencia el menor y el mayor tiempo de los ciclos de fabricación en los diferentes procesos evaluados. En aquellos procesos que no proporcionen un tiempo simulado, se penalizará la calificación. Sin embargo, en el caso de familias de productos, se podrán proporcionar promedios como una alternativa. La evaluación del tiempo de ciclo de fabricación se realizará en una escala del 1 al 5, donde 1 indica un menor tiempo de ciclo y 5 indica un mayor tiempo de ciclo.

A continuación, se presentan las tablas actualizadas que resumen el Tiempo de ciclo de fabricación de cada proceso de manufactura:

Tabla 5-5.: Tiempo de ciclo de fabricación en el LabFabEx-UN.

| Proceso de manufactura | Tiempo de ciclo de fabricación |
|--|--------------------------------|
| Manufactura Aditiva | 2 |
| Manufactura sustractiva – CAM | 1 |
| Manufactura sustractiva – Manual | 1* |
| Manufactura sustractiva – nube de puntos | 2* |
| Corte láser | 2 |
| Moldes de inyección | 3* |

Es importante destacar que algunos procesos, como la manufactura aditiva y algunos procesos sustractivos CAM, proporcionan información de tiempo simulado que se puede representar como una superficie para evaluar el tiempo de ciclo de fabricación de manera más precisa.

Se presenta una tabla con los resultados establecidos en cada criterio para las métricas de flexibilidad, basado en el estado actual del LabFabEx-UN:

Tabla 5-6.: Resumen de criterios de flexibilidad actual del LabFabEx-UN.

| Proceso de manufactura | Disponibilidad Tecnológica | Complejidad Tecnológica | Tolerancias | Tiempo de ciclo de fabricación |
|--|----------------------------|-------------------------|-------------|--------------------------------|
| Manufactura Aditiva | 1 | 1 | 1 | 2 |
| Manufactura sustractiva – CAM | 2 | 4 | 1 | 1 |
| Manufactura sustractiva – Manual | 2 | 3 | 1 | 1* |
| Manufactura sustractiva – nube de puntos | 2 | 5 | 3 | 2* |
| Corte láser | 4 | 3 | 1 | 2 |
| Moldes de inyección | 5 | 5 | 1 | 3* |

Escenario con mejoras de flexibilidad del LabFabEx-UN

Con el objetivo de explorar diversas soluciones que mejoren las calificaciones de los criterios de flexibilidad, se proponen las siguientes mejoras para crear un entorno en el LabFabEx-

UN que fomente la variabilidad de soluciones. Estas propuestas podrían implementarse en este entorno, pero se descartarán debido a que su implementación requeriría más tiempo del establecido para un trabajo de maestría.

En cuanto a la disponibilidad tecnológica, se sugiere la integración de nuevos procesos en el sistema de ejecución del laboratorio. A modo de ejemplo, se plantea la implementación de un proceso que permita generar moldes de inyección 3D y utilizar la inyección de silicona en frío. Como se muestra en la tabla 5-7, la manufactura de moldes de inyección es fácilmente implementable en el esquema del laboratorio, y el uso de silicona para los componentes de contacto general, mencionados en secciones anteriores, resulta en una disminución de la calificación a 2.

Tabla 5-7.: Disponibilidad tecnológica con mejoras en flexibilidad en el LabFabEx-UN.

| Proceso de manufactura | Disponibilidad Tecnológica |
|--|----------------------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 3 |
| Manufactura sustractiva – Manual | 2 |
| Manufactura sustractiva – nube de puntos | 2 |
| Corte láser | 4 |
| Moldes de inyección | 2 |

En cuanto a la complejidad tecnológica, se propone disponer de un postgenerador para el torno suizo y capacitar a un operario en su uso. Esto permitiría una generación más eficiente del código G y la fabricación de componentes con formas complejas. Además, en el caso de los moldes de inyección en frío, la tecnología de impresión 3D simplifica su generación, lo que reduciría la complejidad. Sin embargo, sigue siendo necesario contar con un operario para el proceso de llenado de moldes con silicona. La nueva calificación se observa en la tabla 5-8.

Tabla 5-8.: Complejidad Tecnológica con mejoras en flexibilidad en el LabFabEx-UN.

| Proceso de manufactura | Complejidad Tecnológica |
|--|-------------------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 2 |
| Manufactura sustractiva – Manual | 3 |
| Manufactura sustractiva – nube de puntos | 5 |
| Corte láser | 3 |
| Moldes de inyección | 2 |

En relación a las tolerancias, la única medida que afectaría a los procesos sería la manufactura sustractiva basada en nube de puntos. La resolución de la nube de puntos influye directamente en las tolerancias obtenidas. Aunque un incremento en la resolución podría mejorar las tolerancias, se requerirían más desarrollos en el software planteado y no se alcanzaría el nivel de las tecnologías previamente mencionadas. Por lo cual la nueva calificación de la manufactura sustractiva por nube de puntos se aprecia en la tabla 5-9.

Tabla 5-9.: Tolerancias de los procesos de manufactura con mejoras en flexibilidad en el LabFabEx-UN.

| Proceso de manufactura | Tolerancias |
|---|-------------|
| Manufactura Aditiva | 1 |
| Manufactura sustractiva – CAM | 1 |
| Manufactura sustractiva – Manual | 1 |
| Manufactura sustractiva – nube de puntos | 2 |
| Corte láser | 1 |
| Moldes de inyección | 1 |

A continuación, se muestra un resumen de las modificaciones propuestas:

Tabla 5-10.: Resumen de criterios de flexibilidad con mejoras en flexibilidad en el LabFabEx-UN.

| Proceso de manufactura | Disponibilidad Tecnológica | Complejidad Tecnológica | Tolerancias | Tiempo de ciclo de fabricación |
|--|----------------------------|-------------------------|-------------|--------------------------------|
| Manufactura Aditiva | 1 | 1 | 1 | 2 |
| Manufactura sustractiva – CAM | 3 | 2 | 1 | 1 |
| Manufactura sustractiva – Manual | 2 | 3 | 1 | 1* |
| Manufactura sustractiva – nube de puntos | 2 | 5 | 2 | 2* |
| Corte láser | 4 | 3 | 1 | 2 |
| Moldes de inyección | 2 | 2 | 1 | 3* |

Es importante destacar que la calificación mencionada es válida únicamente para el entorno específico del LabFabEx-UN. En este laboratorio, debido a la maquinaria disponible y los diferentes trabajos realizados, no se han llevado a cabo ejercicios prácticos mediante la inyección de material, aunque sí se han trabajado moldes mediante la remoción de material. Por lo tanto, se asigna la calificación máxima en complejidad para este proceso.

Además, en el laboratorio existen dos celdas especializadas que cubren tanto la manufactura aditiva como la sustractiva. Sin embargo, se debe tener en cuenta que características como el uso de software gratuito para la maquinaria FDM, en contraste con el uso de software con licencias parciales para el torno suizo, generan una clasificación de complejidad menor para la manufactura aditiva en comparación con la sustractiva.

Es relevante señalar que el uso práctico de la maquinaria industrial de remoción de material en el LabFabEx-UN difiere del uso en una empresa, ya que en el laboratorio se utiliza principalmente para estudios de procesos y no para la producción masiva de componentes. Por otro lado, la impresión 3D se ajusta perfectamente a las características de prototipado rápido y permite la realización de varios casos prácticos para la creación de componentes.

En cuanto a la complejidad, se ha considerado la construcción de superficies para generar interpolaciones al tomar ciertos parámetros de un componente y obtener una estimación del tiempo, tal como se muestra en la figura 4-15. Estas superficies se construyen manualmente en el caso del proceso de impresión 3D. Por otro lado, en el proceso de remoción de material se crea otra superficie que también tiene en cuenta el ancho y alto del eslabón, lo cual proporciona una estimación del tiempo. Sin embargo, hasta el momento no se ha establecido un proceso automático para la generación de estas superficies.

5.1.4. Definición de los objetivos

En esta sección se establecen los objetivos del sistema para el desarrollo del gripper. Los objetivos se dividen en tres categorías: espaciales, de fuerza y de flexibilidad.

Los objetivos espaciales se definen en función de la descripción del componente a manipular, utilizando grafos de nodos MUCO. Se analiza la mitad del componente, aprovechando su simetría para desarrollar las cadenas cinemáticas del gripper.

En cuanto a los objetivos de fuerza, se busca minimizar el momento en las articulaciones del gripper mientras se ejerce una fuerza normal sobre el componente a manipular. Se considera una fuerza normal máxima de 2 N, teniendo en cuenta la influencia de los coeficientes de fricción entre los materiales.

Por último, los objetivos de flexibilidad se definen en relación al proceso de manufactura y la estructura de datos que describe el gripper. Se plantean dos entornos para evaluar la flexibilidad en condiciones actuales y mejoradas que incluyen la etapa de diseño y manufactura.

Objetivos espaciales

En base a la definición del componente a manipular como un grafo con varios nodos MUCO, se puede observar en la figura 5-7 que el componente se describe mediante una serie discreta de MUCOs. Sin embargo, en el proceso generativo, cada salida MUCO (punto) que involucre al grafo del componente debe ser evaluado. Esto implica que el sistema debe explorar cada nodo MUCO planteado, pero debido a las limitaciones de la tarjeta gráfica, es necesario limitar estos puntos.

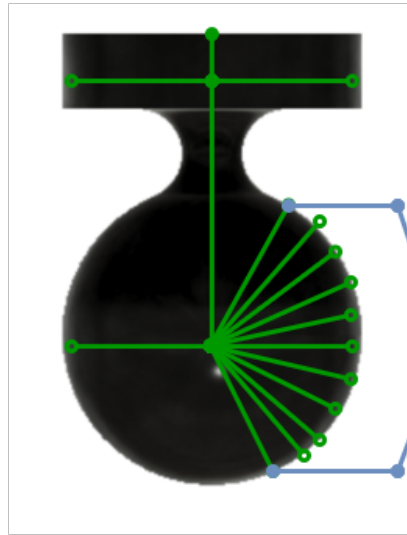


Figura 5-7.: Definición espacio de trabajo del objeto a manipular.

Fuente: Creación propia

Por lo tanto, se analiza solo la mitad del componente a manipular, aprovechando la simetría para el desarrollo de las cadenas cinemáticas del gripper. Las soluciones del gripper se centran en generar una única cadena cinemática, y la solución general implica duplicar la solución seleccionada o combinar diferentes soluciones.

En la mitad del componente a manipular, se definen seis puntos en las fronteras y en el centro, donde los eslabones deben hacer contacto para asegurar un espacio de trabajo adecuado para la manipulación del componente deseado. Estos puntos son cruciales para el contacto de los eslabones con la pieza, ya que la salida de los nodos MUCO del grafo que define el eslabón al evaluarse tiende a tener una distancia cercana a cero para los seis puntos del espacio de trabajo.

Es importante destacar que esta exploración se lleva a cabo mediante el proceso generativo, en el cual los eslabones exploran el espacio al modificar su rotación relativa entre ellos para cumplir con esta condición de la mejor manera posible en todos los puntos. Los detalles de este proceso se explican en profundidad en una sección posterior.

Objetivos de fuerza

En este objetivo se considera la masa del componente a manipular, la cual no debe exceder los 100 g. Durante la interacción entre el eslabón o el componente de contacto general y el componente a manipular, se genera una fuerza normal. Se evalúa el momento resultante en las articulaciones con el objetivo de minimizarlo.

Para el diseño del sistema de accionamiento, se ajusta el modelo mediante el planteamiento

de tensores. La fuerza normal se establece con un límite máximo de 2 N, teniendo en cuenta que su magnitud depende de las condiciones de fricción entre los materiales utilizados. Por ejemplo, se ha observado que dos piezas impresas en PLA pueden tener un coeficiente de fricción cercano a 0.4, mientras que el caucho de silicona utilizado en los componentes de contacto general tiene un coeficiente de fricción alrededor de 0.9, según bases de datos de materiales como MatWeb. Es importante destacar que los coeficientes de fricción pueden verse afectados por diversos factores externos, como la rugosidad de las superficies, que es más notable en los métodos de adición de material. Por lo tanto, en la práctica, se analiza el agarre proporcionado por el gripper al componente planteado, considerando estos factores.

Objetivos de flexibilidad

Los objetivos de flexibilidad se basan en la selección del proceso de manufactura y en la descripción de la estructura de datos del gripper. Es importante tener en cuenta que en la subsección 5.1.3 "Cargar parámetros de fábrica" se especifican dos entornos para evaluar la flexibilidad: el estado actual del laboratorio y posibles mejoras en las propiedades de flexibilidad, tanto cualitativas como cuantitativas, durante la etapa de diseño y manufactura.

5.1.5. Generar población

La población se genera de manera aleatoria, considerando la cantidad de grafos direccionales (componentes) y su conectividad en el sistema. En el sistema, se define el tipo de salida de los grafos direccionales en dos categorías: conexiones fijas e interactivas. Las conexiones interactivas permiten la manipulación de la posición angular del primer nodo MUCO de un grafo (componente) en relación al grafo anterior, con el fin de cumplir los objetivos establecidos. Por otro lado, las conexiones fijas no permiten modificaciones en sus nodos intermedios.

En este contexto, se generan cadenas cinemáticas, teniendo en cuenta que el tipo de grafo de conexión general, al no tener salidas para continuar la cadena, representa el último componente. La aleatoriedad también se aplica en las longitudes de los MUCOs para generar la diversidad en la población.

La cantidad de sistemas que conforman la población se establece en 40 individuos debido a las limitaciones de memoria de la GPU durante el proceso iterativo de evaluación. Aunque es posible aumentar esta cantidad, se requeriría almacenar las poblaciones parciales en grupos de 40 en una base de datos. Esta técnica implica un mayor tiempo de computación y sincronización de memoria de cpu o bases de datos, pero una población más grande en problemas genéticos permite explorar un espacio de trabajo más amplio. En este trabajo, dado que el objetivo no es obtener la mejor respuesta en un sistema de optimización, sino identificar posibles soluciones, se determina que no es necesario el almacenamiento intermedio

de sistemas y se opta por mantener una única población durante todo el proceso iterativo, considerándolo la mejor opción.

5.1.6. Evaluación iterativa

La segunda etapa del flujo planteado en la Figura 5-1 se basa en el algoritmo NSGA-II. En este proceso, las evaluaciones de las funciones objetivo se ordenan utilizando el frente de Pareto y se conserva la mitad de la población con las mejores evaluaciones. Esta población seleccionada se duplica y se somete a mutaciones para generar diversidad en las soluciones. Sin embargo, debido a la naturaleza dinámica de las cadenas cinemáticas, este proceso genético requiere de una resolución genética adicional para facilitar el procesamiento computacional de las generaciones en términos de la dinámica de movimiento.

En el siguiente apartado se explicará con mayor detalle la resolución genética aplicada para abordar la dinámica de movimiento en las cadenas cinemáticas.

Algoritmo Genético para resolver configuraciones

Considerando los objetivos establecidos, se requiere que cada posición espacial sea alcanzada en diferentes momentos, lo que implica que mantener las conexiones interactivas con un único valor angular no permite evaluar adecuadamente los objetivos. Por lo tanto, se propone utilizar un enfoque adicional que permita almacenar las configuraciones angulares de las conexiones interactivas para cada objetivo específico.

Este algoritmo genético se aplica únicamente a la nueva población generada en cada generación, evitando así el recálculo de los sistemas que ya han sido evaluados previamente. La implementación de este desarrollo genético adicional permite reducir la cantidad de iteraciones globales necesarias. Para el caso del Gripper, se establece un total de 200 iteraciones globales. Dentro de cada iteración global, se realizan 100 iteraciones de configuraciones para cada sistema, utilizando una población de 20 individuos en el algoritmo genético para resolver un objetivo espacial específico en cada caso.

Es importante destacar que se utiliza memoria dinámica en la GPU para almacenar la población del algoritmo genético, mientras que los sistemas del algoritmo multiobjetivo principal hacen uso de una memoria compartida entre la CPU y la GPU. Si bien esta última memoria es más lenta, permite una transmisión eficiente de las estructuras de datos complejas. En el caso de la memoria dinámica generada en la GPU, no es posible copiar estas estructuras a la CPU, por lo que solo se retroalimenta el valor de las configuraciones angulares de las conexiones interactivas.

Selección de población

Luego de tener para cada individuo configuraciones ideales para satisfacer los objetivos planteados, se evalúa el cumplimiento y por medio del ordenamiento por frentes de Pareto en GPU para funciones multiobjetivo, se selecciona la mitad de la población que contenga los primeros frentes, adicionalmente en el ordenamiento de los frentes incluye un ordenamiento por distancias euclídeas para mejorar la distribución de las soluciones y preferiblemente no seleccionar un conjunto de soluciones que estén cercanas entre ellas.

Aplicación de mutaciones

En el contexto de los algoritmos genéticos, el objetivo es realizar mutaciones que promuevan la exploración del espacio de soluciones en las primeras iteraciones y, posteriormente, enfocarse en la explotación de las soluciones potenciales mediante mutaciones más sutiles.

En este sentido, las mutaciones se aplican de manera aleatoria a cada grafo, comenzando con una tasa de exploración del 30% y disminuyendo gradualmente hasta un 5% al final del algoritmo. Entre las mutaciones de exploración se incluye la adición o eliminación de grafos direccionales, correspondientes a los eslabones de la cadena, así como la modificación de las magnitudes internas de un MUCO dentro de los grafos. En cuanto a las mutaciones de explotación, se limitan a modificar únicamente las magnitudes internas de un MUCO.

Es importante destacar que, aunque no se aplica en el desarrollo del Gripper debido al esquema de diseño paramétrico en el CAD, en modelos basados en nubes de puntos las mutaciones de los grafos pueden incluir la modificación angular para que el modelo CAD se ajuste a estas características durante el proceso de generación. Incluso, bajo este modelo de nubes de puntos, las mutaciones pueden implicar la composición de un nuevo grafo direccional, ya que no existe dependencia de un modelo paramétrico.

5.1.7. Selección de soluciones finales

En esta etapa, se procede a seleccionar las soluciones finales mediante la ejecución del código en dos escenarios: el escenario actual del LabFabEx y un escenario que considera mejoras en los parámetros de flexibilidad previamente descritos. En cada escenario, se aplican técnicas de ordenamiento basadas en el frente de Pareto utilizando el algoritmo NSGA-II para seleccionar las 5 mejores soluciones o individuos. Estas soluciones no dominadas corresponden al frente 0 del Pareto.

En el primer escenario, se analizan los resultados obtenidos en la tabla **5-11** y se observa una clara tendencia hacia el uso de la Manufactura Aditiva en la mayoría de las soluciones. De las 25 soluciones pertenecientes a los frentes de Pareto 0 y 1, 23 de ellas se alinean con esta metodología. En cuanto al frente 0 de Pareto, 18 de las 20 soluciones están compuestas

por dos eslabones, mientras que únicamente dos soluciones contienen tres eslabones. Es importante destacar que ninguna de las soluciones incluye la opción de contacto general.

En términos de rendimiento, se observa que el tiempo promedio requerido para llevar a cabo cada ejecución es de aproximadamente 2 horas y media. Como indicador clave de viabilidad y estabilidad del sistema, se ha tomado en cuenta el hecho de que el programa tuvo que ser ejecutado en 18 ocasiones para obtener las 5 ejecuciones deseadas. Durante este proceso, se presentaron algunos errores de memoria y dificultades de convergencia, los cuales representaron los principales desafíos encontrados, ligados a un error de la API de CUDA v11 usada en esta tesis.

Tabla 5-11.: Soluciones obtenidas en el entorno actual del LabFabEx.

| Individuo | Ejecución | Frente Pareto | Proceso de Fabricación | Número de Eslabones | Longitud del Primer Eslabón (mm) | Longitud del Segundo Eslabón (mm) | Longitud del Tercer Eslabón (mm) |
|-----------|-----------|---------------|--------------------------------|---------------------|----------------------------------|-----------------------------------|----------------------------------|
| 1 | 1 | 0 | Manufactura Aditiva | 2 | 29.86 | 38.62 | - |
| 2 | 1 | 0 | Manufactura Aditiva | 2 | 27.43 | 39.25 | - |
| 3 | 1 | 0 | Manufactura Sustractiva Manual | 2 | 34.52 | 43.17 | - |
| 4 | 1 | 0 | Manufactura Aditiva | 2 | 32.15 | 44.29 | - |
| 5 | 1 | 0 | Manufactura Aditiva | 2 | 30.06 | 41.91 | - |
| 6 | 2 | 0 | Manufactura Aditiva | 3 | 22.76 | 35.19 | 17.84 |
| 7 | 2 | 0 | Manufactura Aditiva | 2 | 28.93 | 42.15 | - |
| 8 | 2 | 0 | Manufactura Aditiva | 2 | 29.25 | 42.18 | - |
| 9 | 2 | 1 | Manufactura Aditiva | 2 | 30.12 | 44.12 | - |
| 10 | 2 | 1 | Manufactura Aditiva | 2 | 31.84 | 40.74 | - |
| 11 | 3 | 0 | Manufactura Aditiva | 2 | 33.27 | 38.26 | - |
| 12 | 3 | 0 | Manufactura Aditiva | 2 | 25.92 | 42.59 | - |
| 13 | 3 | 0 | Manufactura Aditiva | 2 | 27.61 | 37.92 | - |
| 14 | 3 | 0 | Manufactura Aditiva | 2 | 32.38 | 44.02 | - |
| 15 | 3 | 1 | Manufactura Aditiva | 2 | 30.09 | 40.38 | - |
| 16 | 4 | 0 | Manufactura Aditiva | 2 | 29.75 | 39.15 | - |
| 17 | 4 | 0 | Manufactura Aditiva | 2 | 26.87 | 36.98 | 22.34 |
| 18 | 4 | 0 | Manufactura Aditiva | 2 | 31.56 | 41.79 | - |
| 19 | 4 | 0 | Manufactura Aditiva | 2 | 27.93 | 37.62 | - |
| 20 | 4 | 0 | Manufactura Aditiva | 2 | 33.14 | 40.92 | - |
| 21 | 5 | 0 | Manufactura Aditiva | 2 | 32.67 | 39.85 | - |
| 22 | 5 | 0 | Manufactura Aditiva | 2 | 28.41 | 43.64 | - |
| 23 | 5 | 0 | Manufactura Sustractiva Manual | 2 | 33.96 | 44.26 | - |
| 24 | 5 | 1 | Manufactura Aditiva | 2 | 31.08 | 41.14 | - |
| 25 | 5 | 1 | Manufactura Aditiva | 2 | 27.77 | 37.98 | - |

En el segundo escenario, se presentan los resultados obtenidos, los cuales se encuentran detallados en la tabla 5-12. A continuación, se resumen las principales observaciones:

Se destaca el uso de procesos de remoción de material por Control Numérico por Computadora (CAM) como método de fabricación en las soluciones generadas, sustituyendo las de

remoción de material manual presentes en el primer escenario. Aunque sigue representando una baja presencia en comparación a la manufactura aditiva.

Tabla 5-12.: Soluciones obtenidas en el entorno con mejoras en flexibilidad del LabFabEx.

| Individuo | Ejecución | Frente | Proceso de Fabricación | Número de Eslabones | Longitud del Primer Eslabón (mm) | Longitud del Segundo Eslabón (mm) | Componente de contacto general |
|-----------|-----------|--------|-----------------------------|---------------------|----------------------------------|-----------------------------------|--------------------------------|
| 1 | 1 | 0 | Manufactura Aditiva | 2 | 26.84 | 40.56 | 0 |
| 2 | 1 | 0 | Manufactura Aditiva | 2 | 27.21 | 36.73 | 0 |
| 3 | 1 | 1 | Manufactura Aditiva | 2 | 31.46 | 44.92 | 0 |
| 4 | 1 | 1 | Manufactura Aditiva | 2 | 28.57 | 42.35 | 0 |
| 5 | 1 | 1 | Manufactura Aditiva | 2 | 30.98 | 39.67 | 0 |
| 6 | 2 | 0 | Manufactura Aditiva | 2 | 29.33 | 41.19 | 0 |
| 7 | 2 | 0 | Manufactura Aditiva | 2 | 25.97 | 37.98 | 0 |
| 8 | 2 | 0 | Manufactura Aditiva | 2 | 33.12 | 40.82 | 0 |
| 9 | 2 | 1 | Manufactura Sustractiva CAM | 2 | 32.67 | 38.56 | 0 |
| 10 | 2 | 1 | Manufactura Aditiva | 2 | 31.27 | 39.92 | 0 |
| 11 | 3 | 0 | Manufactura Aditiva | 2 | 27.84 | 43.01 | 0 |
| 12 | 3 | 0 | Manufactura Aditiva | 2 | 28.16 | 37.72 | 0 |
| 13 | 3 | 0 | Manufactura Sustractiva CAM | 2 | 30.49 | 45.21 | 0 |
| 14 | 3 | 1 | Manufactura Aditiva | 2 | 32.72 | 39.15 | 0 |
| 15 | 3 | 1 | Manufactura Sustractiva CAM | 2 | 31.93 | 42.14 | 0 |
| 16 | 4 | 0 | Manufactura Aditiva | 2 | 26.39 | 40.42 | 0 |
| 17 | 4 | 0 | Manufactura Aditiva | 2 | 29.75 | 37.99 | 0 |
| 18 | 4 | 1 | Manufactura Aditiva | 2 | 27.22 | 41.82 | 0 |
| 19 | 4 | 1 | Manufactura Sustractiva CAM | 2 | 31.88 | 44.29 | 0 |
| 20 | 4 | 1 | Manufactura Aditiva | 2 | 29.05 | 39.78 | 0 |
| 21 | 5 | 0 | Manufactura Aditiva | 2 | 32.16 | 42.96 | 0 |
| 22 | 5 | 0 | Manufactura Aditiva | 2 | 27.87 | 36.19 | 1 |
| 23 | 5 | 1 | Manufactura Aditiva | 2 | 28.52 | 43.15 | 0 |
| 24 | 5 | 1 | Manufactura Aditiva | 2 | 26.94 | 40.07 | 0 |
| 25 | 5 | 1 | Manufactura Aditiva | 2 | 30.71 | 39.52 | 0 |

En este escenario, no se encontraron soluciones que involucraran tres eslabones. Esta diferencia significativa se atribuye directamente al método de manufactura utilizado en cada caso, en el cual la presencia de más soluciones de manufactura sustractiva afecta la generación de la población. A diferencia del primer escenario se identifica un componente de contacto general y sin embargo en esa ejecución no converge ninguna solución al método de manufactura sustractiva.

El tiempo promedio requerido para una ejecución se mantiene en aproximadamente 2 horas y media, lo cual se mantiene consistente con los resultados obtenidos anteriormente. Sin embargo, en este caso particular, fue necesario ejecutar el software en 32 ocasiones para

alcanzar la convergencia deseada en tan solo 5 instancias. Esta variabilidad en la convergencia se atribuye tanto a la naturaleza inherente del problema, en el cual en el primer escenario en las primeras generaciones converge a la manufactura aditiva su población, pero en este segundo escenario tiene una población más variada en cuanto al proceso de manufactura y la presencia de componentes de contacto general. Igualmente se destacan las limitaciones impuestas por el hardware utilizado, especialmente el GPU, así como a la propuesta específica planteada en este estudio.

Estos hallazgos evidencian la importancia de considerar múltiples escenarios y evaluar el impacto de los diferentes parámetros y métodos de fabricación en la generación de soluciones finales.

5.1.8. Aplicación al concepto CAD

Esta sección corresponde a la tercera etapa del flujo propuesto en la figura 5-1, en la cual se realiza la retroalimentación de las soluciones obtenidas en los modelos paramétricos en Fusion 360. En este flujo, la etapa de retroalimentación puede llevarse a cabo tanto de forma manual por parte del diseñador como a través de procesos automáticos.

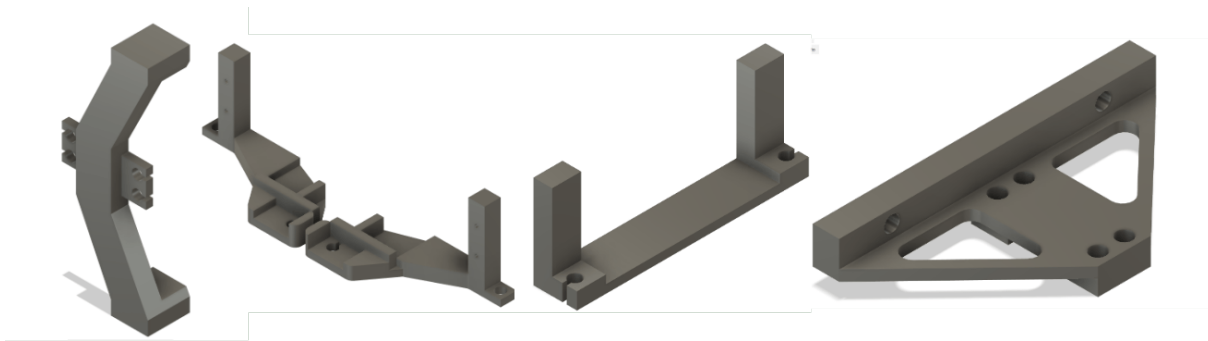


Figura 5-8.: Muestra de componente modulares para desarrollo de estructura y sujeción de actuadores.

Fuente: Creación propia

Aunque es posible agregar la cantidad de componentes necesarios, ubicarlos en el espacio y modificar sus parámetros utilizando las APIs de Fusion 360, el mapeo de las soluciones se establece internamente a través de estas APIs. Sin embargo, surge un inconveniente en la complejidad del proceso para agregar un nuevo componente de manera automática. De hecho, en muchos casos resulta más eficiente realizar esta tarea manualmente en un menor tiempo. En el presente escenario, dado que existen ajustes que el diseñador debe realizar para la integración de todos los componentes, la automatización de la interacción no resulta obligatoria.

Uno de los ajustes que debe realizar el diseñador es agregar los componentes que interactuarán, como los tensores y los actuadores. En este caso, basado en conceptos de modularidad, la estructura responde a la cantidad de eslabones definidos por el sistema generativo. Dado que los elementos electrónicos, como los motores, son discretos, su acople puede estandarizarse mediante la adquisición y el escalado de los mismos. La figura 5-8 ilustra el desarrollo de componentes modulares que se ajustan a dos tipos de servomotores, lo cual permite que el sistema pueda adaptarse a diferentes tipos de servomotores para abordar el problema planteado.

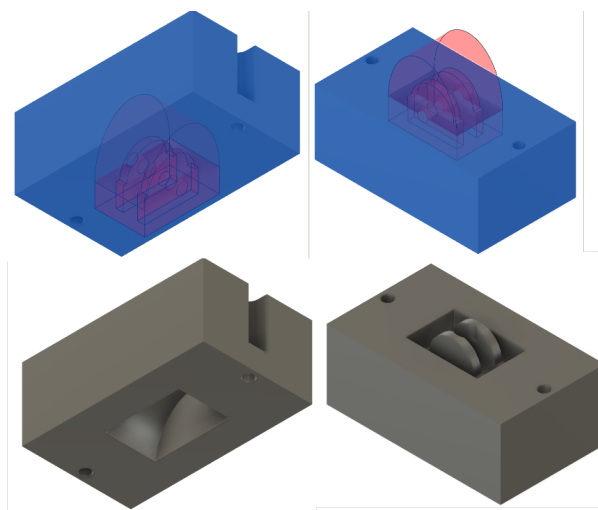


Figura 5-9.: Generación molde en CAD para grafo de contacto general.

Fuente: Creación propia

Adicionalmente, aunque no se trata de una solución óptima obtenida, se ha desarrollado un molde para el grafo direccional de contacto general, como se muestra en la figura 5-9. Este concepto de molde se basa en los mismos principios de parametrización y creación de interfaces rígidas. El molde consta de dos secciones que incluyen canales para la aplicación del caucho de silicona y tornillos de sujeción en los extremos.

Estos avances demuestran la aplicación práctica del enfoque CAD en la generación de componentes modulares y moldes que se adaptan a diferentes configuraciones y requisitos del sistema, permitiendo una mayor flexibilidad y eficiencia en el proceso de fabricación.

5.1.9. Manufactura del producto

La manufactura de las dos cadenas cinemáticas se compone de tres eslabones cada una. El eslabón base se conecta a la estructura y es fijo, sin la libertad rotacional de los otros dos eslabones. Aunque la solución seleccionada consta de dos eslabones, se ha identificado la necesidad de un tercer eslabón con parámetros que no afecten el funcionamiento de los dos

primeros. Estos parámetros incluyen un ancho mínimo para el eslabón siguiente, un largo que permita el giro del siguiente eslabón y un espesor adecuado para soportar las cargas aplicadas en la cadena.

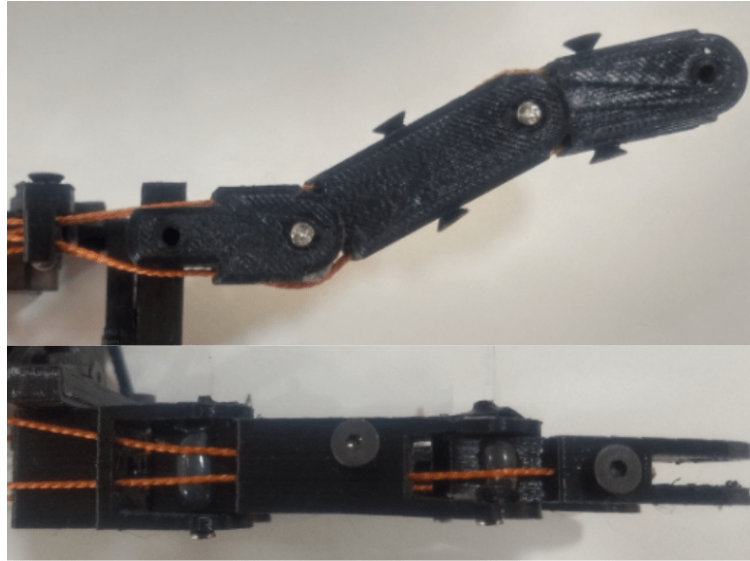


Figura 5-10.: Vista lateral y frontal de la cadena cinemática impresa.

Fuente: Creación propia

La figura **5-10** muestra el desarrollo de la cadena cinemática. La manufactura se realiza mediante adición de material e incluye detalles como las guías internas para los tendones. Se observa que los tendones se fijan mediante tornillos. Tras varias iteraciones, se han realizado mejoras, como la adición de radios de giro en las secciones giratorias de los eslabones para generar el efecto de polea y mejorar la rotación en cada eslabón. Es importante tener en cuenta consideraciones críticas para la manufactura por remoción de material, como el uso de brocas con diámetros pequeños y largas longitudes para las guías internas. Estas consideraciones adicionales deben ser tenidas en cuenta por el diseñador al plantear un cambio en el proceso de manufactura.

La estructura, basada en el concepto de modularidad que se muestra en la figura **5-11**, sigue un enfoque de manufactura secuencial. Se fabrican varias piezas de menor complejidad en lugar de producir piezas más grandes y con geometrías más complejas. Durante el proceso iterativo, se han realizado mejoras, como el refuerzo de los canales de los tendones y de la estructura en general, para evitar deformaciones debido a los esfuerzos que se transmiten a través de los tendones.

La figura **5-12** muestra la fabricación del molde y los elementos de contacto general mediante impresión 3D. El molde es utilizado para la inyección en frío de caucho. Una ventaja de este

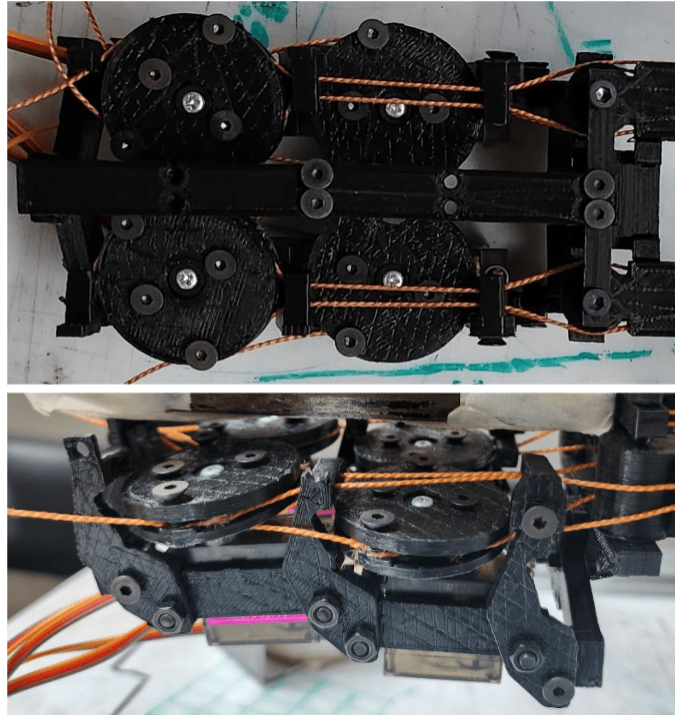


Figura 5-11.: Vista frontal y lateral de la estructura del Gripper.

Fuente: Creación propia

tipo de molde fabricado con esta tecnología es que el porcentaje de relleno puede ser bajo, ya que no estará sometido a esfuerzos ni a variaciones de temperatura. Esto permite reducir el tiempo de fabricación de los moldes y el consumo de material.

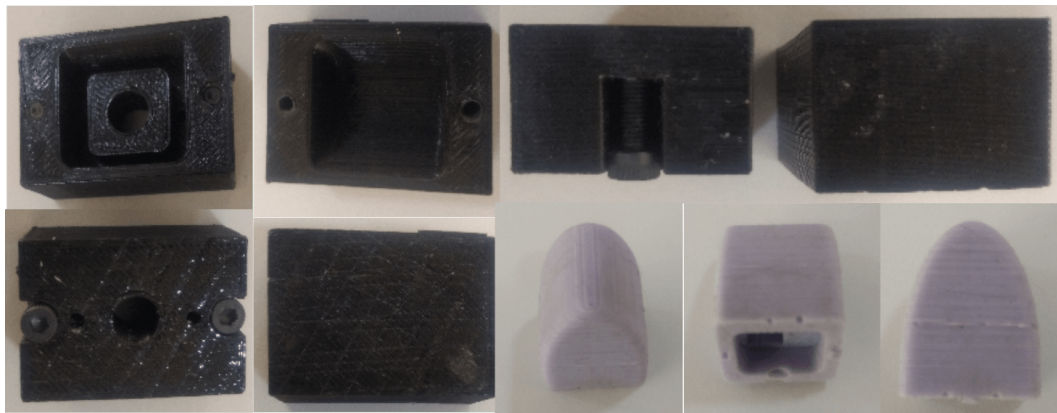


Figura 5-12.: Manufactura de los moldes y elementos de contacto general.

Fuente: Creación propia

Estos avances en la manufactura del producto demuestran la aplicación de diversas técnicas y tecnologías para la producción de las cadenas cinemáticas, la estructura del Gripper y los

moldes de contacto general. La consideración de parámetros, mejoras iterativas y la elección de tecnologías adecuadas contribuyen a la eficiencia y calidad en la fabricación del producto final.

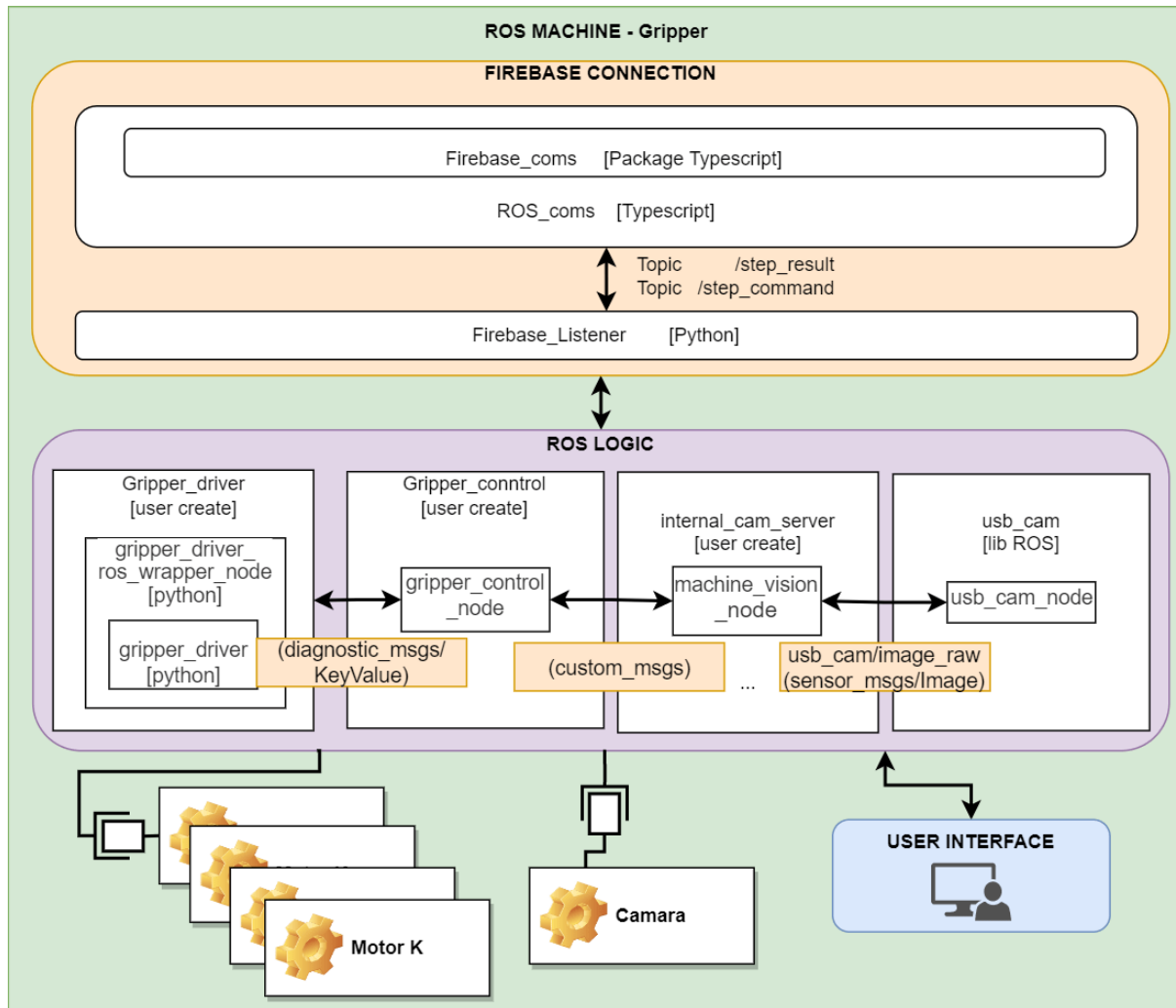


Figura 5-13.: Arquitectura del Gripper para integración con software MES.

Fuente: Creación propia

5.1.10. Integración de control basado en ROS para el sistema PRIA-UN

Se ha desarrollado una integración del Gripper con el software PRIA-UN del LabFabEx-UN, como se muestra en la arquitectura de la figura 5-13. Esta arquitectura consta de un módulo de conexión con la base de datos y un módulo de lógica del dispositivo basado en ROS (Robot Operating System). El módulo de conexión es genérico para PRIA-UN y permite la rápida incorporación de nuevos robots, mientras que en el módulo de ROS se han desarrollado

paquetes que permiten interactuar con los periféricos y calcular la lógica de trabajo.

Es importante destacar que se ha utilizado Docker para desplegar esta arquitectura en servidores de bajo costo, como el Raspberry Pi 4. Estos servidores también permiten agregar el robot al sistema PRIA-UN del laboratorio. Anteriormente, integrar un robot en este sistema resultaba complejo, pero con el uso de Docker se ha logrado una integración más sencilla y eficiente.

Entre los periféricos utilizados se encuentran los motores, cuya cantidad depende del desarrollo del sistema generativo. En el caso del Gripper, se utilizan cuatro motores que corresponden a las dos cadenas cinemáticas. Además, se cuenta con una cámara que permite realizar mediciones indirectas del Gripper. El desarrollo del módulo de control se centra en recibir la posición que cada motor debe alcanzar. Esta posición puede ser enviada desde la base de datos o interactuar con los paquetes de visión de ROS, que se comunican directamente con la cámara.

El código fuente del controlador se encuentra disponible en los anexos K y L, donde se detallan las implementaciones en Python y en ROS, respectivamente.

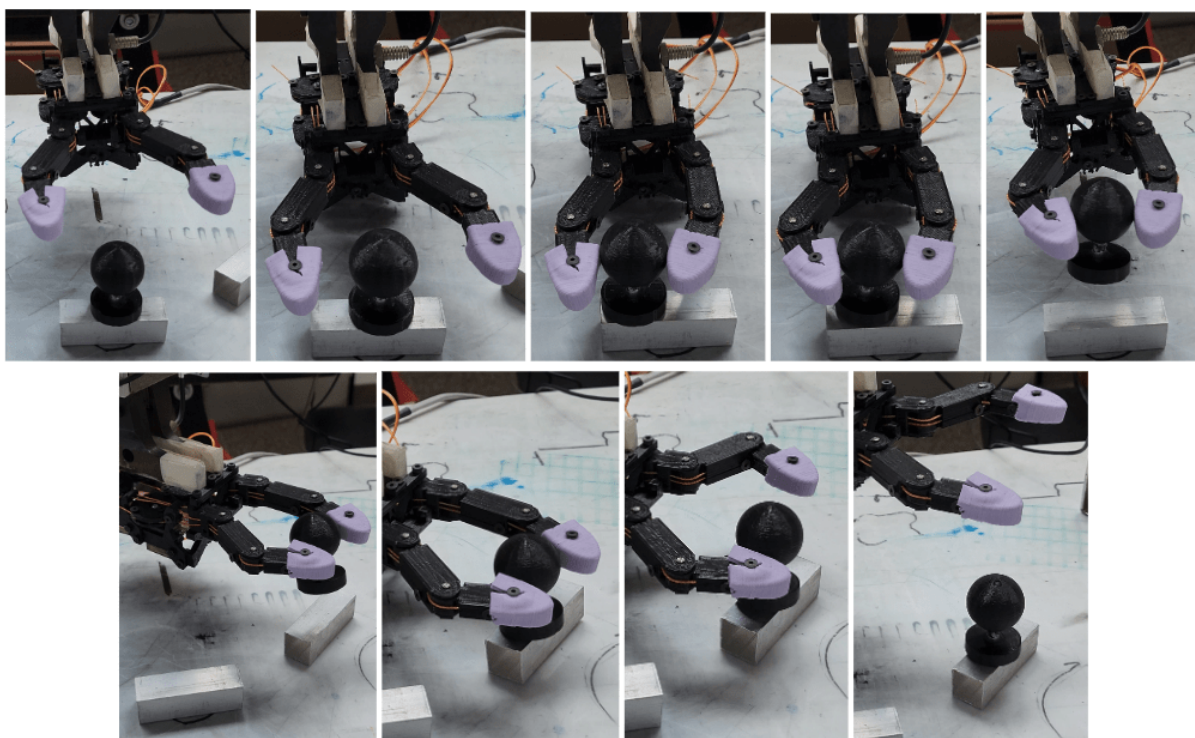


Figura 5-14.: Secuencia del Gripper realizando la operación de Pick and Place.
Fuente: Creación propia

Se propone una rutina para llevar a cabo la operación de Pick and Place en el entorno del LabFabEx-UN mediante el uso de PRIA-UN. En este caso, la integración se ejecuta por medio de NodeRED, que utilizará el robot SCARA para sujetar el Gripper, el cual tomará el objeto analizado de una posición A a una posición B. En la figura 5-14 se observa la secuencia tomada de un vídeo de la operación del Pick and Place.

En el contexto de NodeRED, en la figura 5-15 se observan los bloques funcionales para ejecutar los comandos tanto en el robot SCARA como en el Gripper llamado ROBOT1. Dado que esta metodología permite desplegar más Grippers bajo la arquitectura del PRIA-UN, se utiliza una notación genérica para distintos robots.

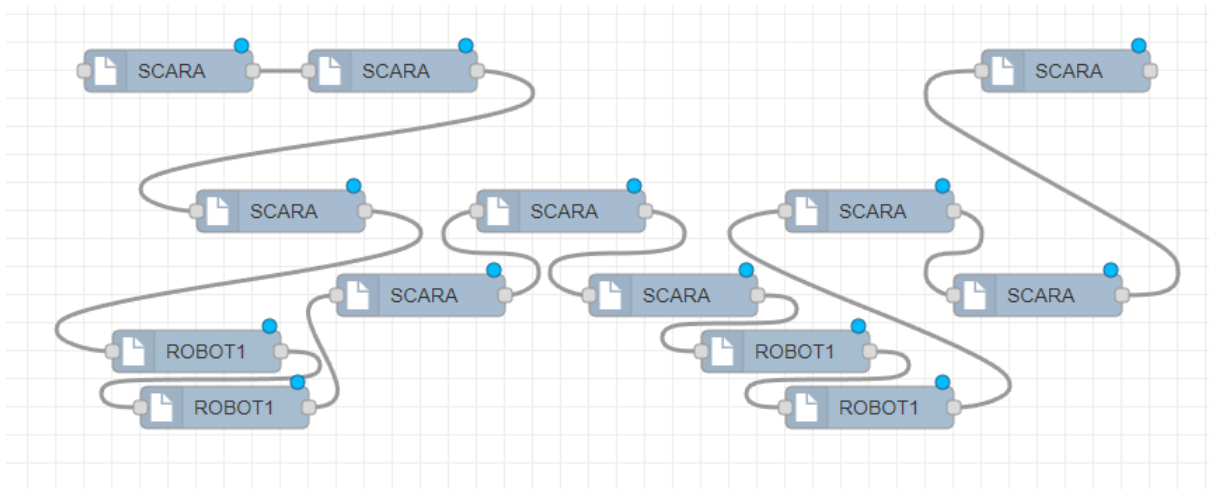


Figura 5-15.: Descripción de la secuencia de ejecución en NodeRED.

Fuente: Creación propia

Basado en la descripción de la arquitectura de la figura 5-13, se detalla cómo el nodo de visión de máquina permite alimentar el nodo de control del Gripper que se encuentra en el rectángulo rojo de la figura 5-16. En el cual se ejecuta en dos etapas por secuencia, en la primera realiza una secuencia en lazo abierto que aproxima las falanges finales al objeto y luego, por medio de regresión usando las características del capítulo anterior, calcula el punto centro para agregarlo al ángulo de rotación de la última falange y realizar el agarre del componente.

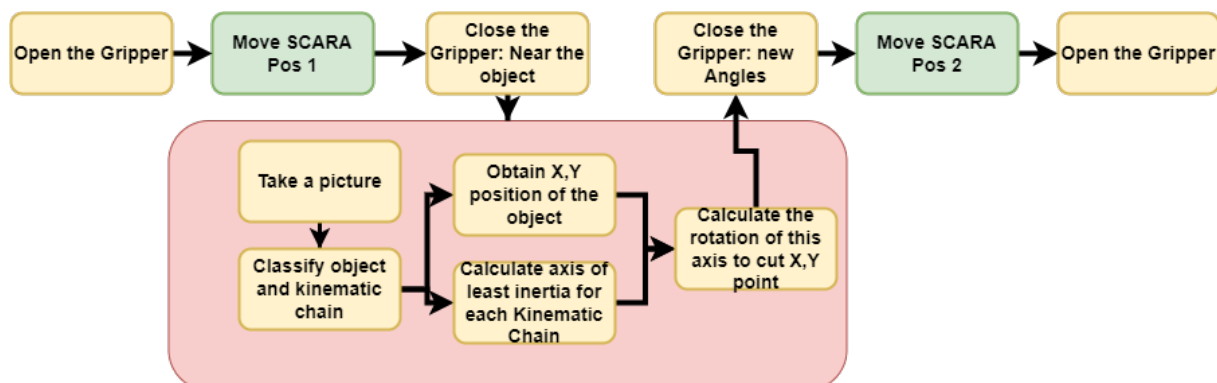


Figura 5-16.: Flujo de operación del Pick and Place.

Fuente: Creación propia

La estructura de visión de maquina se encuentran descritas en la figura 4-10, en el cual por medio de la adquisición de las imágenes sintéticas de la figura 4-11 se entrena el sistema para clasificar los objetos. Las características descritas se calculan por medio de los contornos obtenidos por la segmentación de la figura 4-14. Este control se desarrolla en términos de mostrar la integración con los procesos de visión, sin embargo, para trabajos futuros se propone un desarrollo óptimo de control por visión de maquina, que realmente en tiempo real la posición de todas las falanges respecto al objeto a tomar.

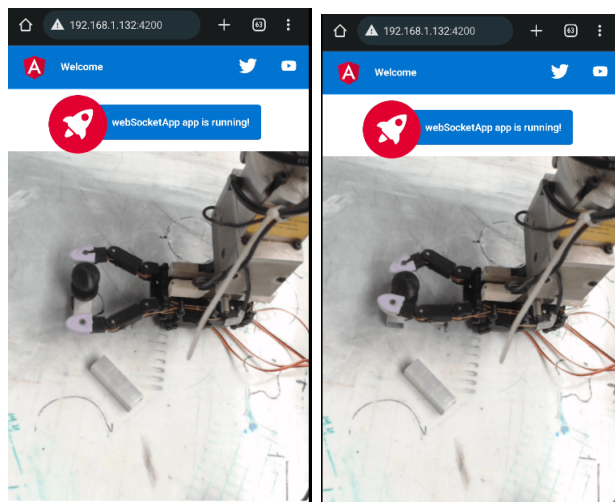


Figura 5-17.: Visualización del proceso de Pick and Place bajo la configuración de servidor.

Fuente: Creación propia

Se indica que la estación de la cámara se encuentra ubicada de la misma manera que se planteó en la figura 4-13 de la primera imagen y en la figura 5-18 se observa cómo desde un dispositivo externo se puede consultar el nodo *usbcamnode* que transmite las imágenes en ROS de la cámara por medio de WebSocket.

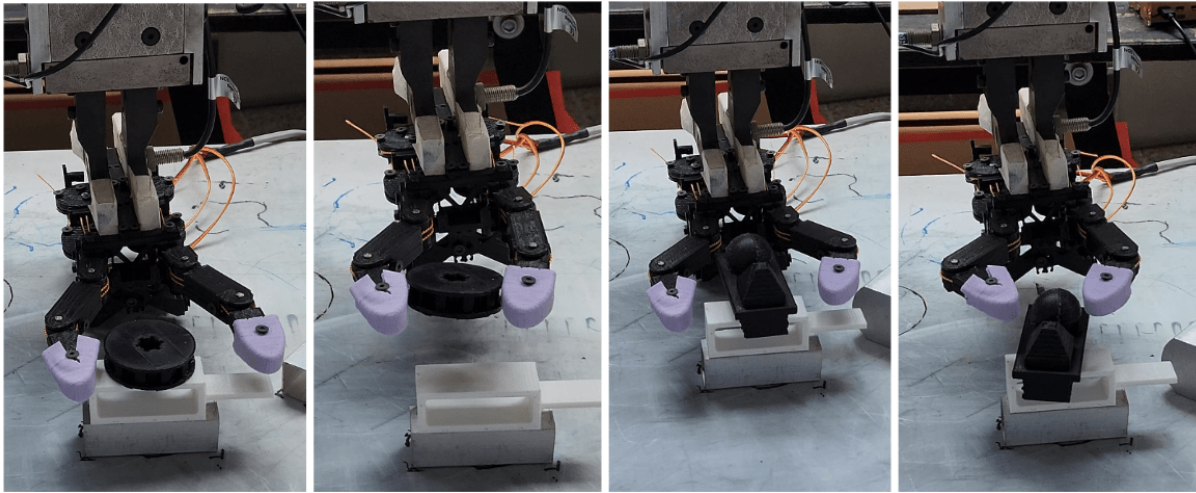


Figura 5-18.: Secuencia de sujeción del objeto Circular y rectangular.

Fuente: Creación propia

Las operaciones de Pick and Place se llevaron a cabo con dos objetos adicionales para verificar la respuesta del sistema ante un objeto desconocido. Se seleccionó una pieza impresa en 3D de forma circular, que conserva la característica principal del objeto entrenado, y una pieza rectangular que presenta una diferencia mayor de la característica principal al objeto a tomar. Los objetos se pueden observar en el entorno en la figura 5-18.

En la tabla 5-13 se resumen los resultados de la aplicación de la operación de Pick and Place. Se resalta que la pieza circular es la que tiene mejor respuesta en la operación, que al compararla con la pieza con la que se entrena el sistema. Esto se debe a que una esfera es menos estable para el agarre que una circunferencia con paredes planas. Adicionalmente, se nota que la pieza rectangular no es detectada correctamente en casi ningún intento, lo que genera que no se pueda llevar a cabo esta operación con esta pieza.

Tabla 5-13.: Resultados de operación de Pick and Place con diferentes piezas.

| Pieza | Intentos | Correctos | Fallidos |
|-------------|----------|-----------|----------|
| Entrenada | 23 | 9 | 14 |
| Circular | 15 | 12 | 3 |
| Rectangular | 7 | 1 | 6 |
| TOTAL | 45 | 22 | 23 |

En conclusión, se logra un control integrado con la plataforma PRIA-UN por medio de la interfaz de NodeRED para la ejecución secuencial de la operación de Pick and Place. Se recalca nuevamente que la intención de esta tesis de maestría no se enfoca en el control por sistemas de visión, sino que denota su viabilidad dentro la metodología general de diseño

generativo para la creación de nuevos productos y recalcar su usabilidad en la etapa de ciclo de vida del producto en el uso.

5.2. Discusión de resultados

La metodología de diseño generativo de producto desarrollada en esta tesis de maestría tiene como objetivo principal brindar soporte a la etapa de diseño en sistemas de manufactura flexibles, tomando en cuenta el ciclo de vida del producto. Los resultados obtenidos se resumen de la siguiente manera:

5.2.1. Estructuras de Datos Basadas en Grafos para Modelar Componentes en el Diseño Generativo

En relación al primer objetivo específico, se han desarrollado dos estructuras de datos basadas en grafos direccionales para el flujo de información de los parámetros de los componentes. La primera estructura se basa en un modelo matemático y requiere una descripción de variables por parte del diseñador. Esta estructura se ejecuta en programación secuencial y tiene limitaciones en términos de cómputo y tiempo de ejecución. La segunda estructura se basa en la definición de primitivas y permite modelar un componente con la interacción de varios nodos en el grafo. Esta estructura es escalable en computación paralela utilizando la API CUDA de NVIDIA, lo cual permite operar más objetivos en menos tiempo. Sin embargo, su implementación requiere configuración y conocimientos especializados en programación paralela.

Estructura de datos basada en el modelo matemático:

En este hallazgo, se observa que la estructura de datos basada en el modelo matemático logra cumplir con los objetivos principales de posición espacial y fuerza en un tiempo de ejecución promedio de 2 horas. Esto indica que la estructura es efectiva para resolver problemas de diseño generativo que requieran optimización en términos de posición y fuerza.

Sin embargo, al integrar un tercer objetivo de flexibilidad basado en el tiempo de impresión 3D de un eslabón, se evidencia un aumento significativo en el tiempo de resolución promedio, llegando a 7 horas. Este aumento en el tiempo de cálculo sugiere que la estructura de datos puede enfrentar limitaciones al abordar múltiples objetivos a la vez. Es importante considerar que este incremento puede estar relacionado con la complejidad del problema al incluir la impresión 3D y los tiempos asociados.

Para futuras investigaciones, se podría analizar el desempeño de esta estructura al abordar diferentes combinaciones de objetivos y cómo podría optimizarse su rendimiento para reducir el tiempo de ejecución en escenarios con múltiples objetivos.

Estructura de datos basada en primitivas:

En el caso de la estructura de datos basada en primitivas, se evaluó con éxito un conjunto más amplio de objetivos, incluyendo 6 objetivos espaciales, uno de fuerza y uno de flexibilidad, con un tiempo de ejecución promedio de 2 horas y media. Esto sugiere que esta estructura presenta una mayor versatilidad para abordar problemas más complejos que involucren múltiples objetivos en el diseño generativo.

A pesar de los buenos resultados, se observaron problemas de convergencia que afectaron la ejecución del algoritmo, lo que requirió realizar múltiples ejecuciones (cerca de 50 ejecuciones) para obtener los resultados deseados (10 conjuntos de soluciones). Estos problemas pueden estar vinculados al manejo de memoria utilizando la API CUDA 11, lo que sugiere áreas de mejora en futuras versiones o ajustes en la configuración.

En conclusión, ambos hallazgos demuestran la eficacia de las estructuras de datos en abordar objetivos específicos en el diseño generativo de producto. Mientras que la estructura basada en el modelo matemático se destaca por su rendimiento en objetivos principales, la estructura basada en primitivas muestra una mayor flexibilidad al lidiar con un conjunto más amplio de objetivos. Estas conclusiones pueden ser de gran relevancia para el desarrollo y aplicación de metodologías de diseño generativo en sistemas de manufactura flexibles. Además, se señalan áreas de mejora en la optimización del rendimiento y convergencia de las estructuras de datos, lo que abre oportunidades para futuras investigaciones en el campo del diseño generativo.

5.2.2. Desarrollo del Gripper Utilizando Diseño Generativo de Producto

El segundo objetivo específico de esta investigación consistió en el desarrollo de un gripper utilizando la metodología de diseño generativo de producto. El gripper se diseñó tomando en cuenta los componentes modelados con las estructuras de datos basadas en grafos direccionales mencionadas anteriormente. Se desarrolla en el capítulo 5 por medio de la estructura de datos basado en primitivas, debido a que es un entorno más completo respecto a la cantidad de objetivos, sin embargo, en el anexo E se muestra la validación realizada en el LabFabEX-UN con la estructura de datos basado en el modelamiento matemático.

El desarrollo del gripper se ha llevado a cabo en el Laboratorio de Fábrica Experimental - UN (LabFabEX-UN), aprovechando las tecnologías disponibles en este entorno. Las validaciones

se realizaron en etapas, ejecutando los algoritmos generativos para alcanzar los objetivos espaciales, de fuerza y de flexibilidad previamente definidos.

Con el uso respecto a primitivas se obtuvieron los resultados condensados en la tabla **5-11** y la tabla **5-12**, donde se destaca el proceso de manufactura aditiva como el dominante en el desarrollo de la evaluación del diseño generativo. Para la estructura de datos basado en el modelo matemático que se encuentra en el anexo E se establece por defecto la manufactura aditiva y con el uso de una superficie que se observa en la figura **4-15**, que describe el tiempo de manufactura basado en los parámetros de largo y ancho del eslabón, se obtienen las soluciones del sistema generativo.

Las soluciones obtenidas para ambos casos indica que la configuración de 2 eslabones es la principal para cumplir con los objetivos establecidos, sin embargo, se presentan en menor medida soluciones de 3 eslabones.

Las soluciones generadas fueron representadas utilizando software CAD, utilizando un enfoque de diseño paramétrico basado en sólidos. También se exploró el método de representación mediante nubes de puntos, aunque esta técnica se encuentra en una etapa temprana de desarrollo y su implementación en entornos de fábricas aún está en proceso de evaluación y mejora.

Para destacar los resultados obtenidos por el uso, se menciona que solo se integra una solución del método de la estructura de datos basada en primitivas del capítulo 5. Este desarrollo del gripper se ejecuta en pruebas de campo en el LabFabEx-UN para llevar a cabo al operación de Pick and Place. El sistema funciono durante tres horas seguidas bajo la arquitectura propuesta al desplegarla en el servidor Raspberry Pi 4 y la tecnología de docker de fácil despliegue. Lo que recalca que con servidores de bajo costo se puede tener una integración robusta del Gripper en el sistema PRIA-UN y asegurando la disponibilidad de este robot.

La repetibilidad del Gripper como se observa en la tabla **5-13** se ve afectada por el lazo de control basado en sistemas de visión de maquina implementado y que se aprecia en la figura **5-18**. Este sistema depende de medidas indirectas tomadas por la cámara, por lo cual pretende interactuar en el entorno de una manera parecida a los humanos, por lo cual la precisión y resolución no son los requerimientos principales y no se toman mediciones de este tipo.

En conclusión, el desarrollo del gripper mediante el diseño generativo de producto ha sido exitoso y ha permitido obtener resultados satisfactorios en términos de cumplimiento de objetivos y funcionalidad. No obstante, se reconocen oportunidades de mejora, particularmente en la integración con sistemas de visión de máquina para mejorar la repetibilidad y precisión. Estos hallazgos nos proporcionan valiosos conocimientos para futuras investigaciones y desarrollos en el campo de los sistemas de manufactura flexibles y diseño generativo. El

enfoque de diseño adoptado representa un avance significativo en la creación de productos optimizados y adaptados a las necesidades cambiantes del ciclo de vida del producto.

5.2.3. Evaluación de la Flexibilidad en el Proceso de Diseño

El tercer objetivo específico de este trabajo de tesis consistió en la evaluación de la flexibilidad del sistema de manufactura flexible mediante la metodología de diseño generativo. En lugar de esperar hasta el final del proceso de diseño, se integró la evaluación de la flexibilidad como un objetivo clave en los algoritmos generativos. En el cual basado en la propuesta de Kim et al. [69] se plantean la metrica de medición de flexibilidad que se usa en esta tesis de maestría.

La flexibilidad se evaluó utilizando valores relativos que fueron normalizados en una escala del 1 al 5. Se consideraron tanto propiedades cuantitativas como cualitativas, y en el caso de las propiedades cualitativas, se utilizaron criterios establecidos por el diseñador para asignar los valores normalizados.

Las estructuras de datos desarrolladas permitieron incorporar la evaluación de la flexibilidad desde el inicio del proceso de diseño, ya que cada grafo direccional representaba un componente y la conexión entre los grafos reflejaba la interacción entre los componentes.

Además, las estructuras de datos proporcionaron información sobre la forma de los componentes, basada en la cantidad de elementos de descripción MUCO utilizados para cada componente. De esta manera, la flexibilidad fue evaluada en todo momento durante el proceso de diseño.

Para evaluar las propiedades de flexibilidad, se realizó la evaluación en el entorno del laboratorio, centrándose en la disponibilidad tecnológica, la complejidad tecnológica, las tolerancias y el tiempo de fabricación. Estas propiedades se basaron en criterios cualitativos que fueron cuantificados por el diseñador.

Es importante destacar que las evaluaciones de flexibilidad fueron realizadas considerando diferentes calificaciones de los criterios establecidos por el diseñador, lo que permitió explorar diversas soluciones de manufactura. También se tuvo en cuenta la interacción de otros componentes, como el componente de contacto general, que se fabricó utilizando un molde de inyección.

En general, los resultados obtenidos en esta investigación demuestran la viabilidad y utilidad de la metodología de diseño generativo como soporte a la etapa de diseño en sistemas de manufactura flexibles. Las estructuras de datos desarrolladas permitieron la transferencia de información descriptiva en diferentes etapas del ciclo de vida del producto, lo que facilita la

creación de nuevos productos con mejores descripciones y la optimización de las propiedades de flexibilidad.

Sin embargo, es importante mencionar que existen áreas de mejora y oportunidades para futuras investigaciones. Por ejemplo, se identificaron problemas de convergencia en la estructura de datos basada en primitivas, que podrían ser solucionados con mejoras en el manejo de memoria. Además, se sugiere explorar el control en lazo cerrado utilizando sistemas de visión, así como avanzar en la búsqueda de un sistema robusto de representación por nubes de puntos.

5.3. Conclusiones del proceso

El presente trabajo de tesis de maestría ha demostrado la aplicación exitosa de una metodología de diseño generativo basada en múltiples hilos conductores, adaptándose a diferentes tipos de distribución de fábricas. Esto implica que existen metodologías que promueven el uso de software comercial, pero también se plantea un ecosistema fundamentado en nuevos enfoques para futuros trabajos, donde el proceso responda a la automatización y reduzca la interacción del usuario. De esta manera, un sistema de diseño generativo realimentado deja de ser simplemente un soporte y se convierte en una herramienta para la creación de productos.

La programación paralela se establece como un campo que responde a la automatización del diseño, ya que aborda la necesidad de resolver funciones multiobjetivo de mayor complejidad, algo inviable en la programación secuencial. Además, bajo el concepto de modelado por primitivas, las GPU permiten realizar operaciones simples y de corta duración, que se pueden ejecutar en paralelo en grandes cantidades.

La adopción de diseños basados en nubes de puntos se ajusta directamente a la capacidad de las GPU. Si bien las representaciones basadas en contornos simplifican la descripción del modelo, también implican el desarrollo de herramientas para su manipulación y descripción. Por lo tanto, las operaciones realizadas por GPU adquieren un papel fundamental. Por otro lado, el modelado por nubes de puntos permite que cualquier modificación realizada en el entorno de trabajo se refleje inmediatamente en el modelo, aunque esto conlleva un mayor consumo de espacio. Sin embargo, esta opción gana fuerza debido a la infraestructura actualmente disponible y a la facilidad para satisfacer las demandas de almacenamiento de las tarjetas gráficas del mercado.

Las soluciones basadas en los modelos obtenidos a través de nubes de puntos ofrecen un escenario ideal para el desarrollo de software en centros educativos. Esto permite el continuo desarrollo de personal capacitado en estas instituciones, democratizando la capacidad de asistencia computacional como una producción tecnológica. Además, genera empleo continuo y personalizado, adaptable a las necesidades específicas de una empresa.

La toma de decisiones en el proceso de diseño se realiza con el objetivo de explorar múltiples objetivos. Al final de cada iteración, el diseñador cuenta con un conjunto de soluciones que abarcan estos requerimientos, lo que facilita la gestión del proceso a escala humana. Áreas del conocimiento como el análisis de datos serán importantes para permitir que el sistema reduzca la dimensionalidad y maneje las correlaciones entre variables. Aunque actualmente los diseñadores realizan la reducción de dimensionalidad al realizar múltiples comparaciones entre parámetros y generar matrices en escenarios de 2 o 3 dimensiones, en el enfoque planteado del diseño generativo realimentado se pretende evaluar todos los posibles objetivos a resolver, incluso en dimensiones superiores.

En resumen, este trabajo de tesis ha sentado las bases para la aplicación exitosa de la metodología de diseño generativo, con enfoques en la automatización y reducción de la interacción humana. El uso de programación paralela y diseños basados en nubes de puntos ha demostrado su eficacia en la generación de soluciones óptimas y en la simplificación del proceso de diseño. Además, se ha destacado la importancia de la toma de decisiones basada en múltiples objetivos y se ha señalado el potencial de la reducción de dimensionalidad y el análisis de datos en este contexto. En definitiva, este trabajo sienta las bases para futuras investigaciones y desarrollos en el campo del diseño generativo, con el objetivo de mejorar la eficiencia y la calidad de los productos manufacturados.

5.4. Trabajos futuros

El análisis de estructuras de datos realizado en este trabajo se limitó a dos dimensiones con el propósito de demostrar su aplicabilidad. Sin embargo, como trabajo futuro se plantea la ampliación de este enfoque al espacio tridimensional, lo que permitiría abordar problemas y aplicaciones más complejas.

Se propone el desarrollo de la propagación inversa en un grafo, con el objetivo de retroalimentar los valores desde la última ubicación del grafo hacia el inicio. Esto evitaría la necesidad de implementar sistemas con poblaciones muy grandes y generaciones adicionales, ya que se busca lograr la convergencia a una solución del problema. Este enfoque sería aplicable en casos específicos donde se requiera mantener el equilibrio del sistema, utilizando técnicas de optimización en un sistema realimentado.

La optimización del tiempo de resolución en el desarrollo de cadenas largas de grafos direccionales para representar elementos flexibles es otro aspecto a considerar en trabajos futuros. Se

sugiere aplicar técnicas de división del grafo y posteriormente aplicar la adición del desfase, con el objetivo de reducir el tiempo necesario para obtener una solución.

Es necesario realizar un estudio detallado sobre la implementación óptima del ecosistema basado en nubes de puntos. Si bien en este trabajo se llevaron a cabo pruebas de concepto, es necesario profundizar en la mejor implementación de estos conceptos para obtener resultados más sólidos y confiables.

En este trabajo se han sentado las bases para futuros desarrollos en términos de automatización del proceso de diseño. Se planteó el uso de programación paralela para realizar procesos en cada etapa, lo cual se asemeja a los procesos estandarizados utilizados en la actualidad, pero aplicados al contexto de las nubes de puntos. Se sugiere que los trabajos futuros se centren en el desarrollo de muestras de estos procesos para el ecosistema propuesto, con el objetivo de consolidar y validar su aplicabilidad en diferentes escenarios de diseño.

Para concluir los trabajos futuros deberían abordar la ampliación del análisis a tres dimensiones, el desarrollo de la propagación inversa en grafos, la optimización de cadenas de grafos direccionales para elementos flexibles, el estudio detallado de la implementación del ecosistema basado en nubes de puntos y la realización de muestras de los procesos propuestos. Estos avances permitirán un mayor refinamiento y aplicación práctica de la metodología de diseño generativo en el futuro.

A. Anexo: Librería para clasificación por frentes de Pareto

Se establece el código fuente de la librería de creación propia, para la resolución de los frentes de Pareto, bajo programación paralela en CUDA.

```
1) #ifndef _LIB_PARETO_FRONT
2) #define _LIB_PARETO_FRONT
3)
4) #define BLOCK_SIZE 32
5)
6) #include <cuda_runtime.h>
7)
8) using namespace std;
9)
10) /**
11) * @brief Inicialice sequency to obtain Pareto Front only minimize all the objective, if
    necessary maximize multiply -1
12) *
13) * @param dataGPU - memory shared CPU-GPU
14) * @param arrayFrontsGPU - array that almacene the position of the front for each
    point
15) * @param NPoints - all the points
16) * @param kObjectives - qty of functions
17) * @param posFront - number of iteration or actually pos Front
18) * @param isEnd - flag that indiciated that all point has a front assigned
19) * @return __global__
20) */
21) __global__ void InitSequency(float **dataGPU, int *arrayFrontsGPU, int NPoints, int
    kObjectives, int posFront, bool *isEnd, bool * isAssignedFront){
22)     int idx = blockDim.x * blockIdx.x + threadIdx.x;
23)     if(arrayFrontsGPU[idx]j=0){
24)         //arrayFrontsGPU[idx] = idx;
25)         isEnd[0] = false;
26)         for(int a=0;a<NPoints;a++){
```

```

27)         if(idx != a && arrayFrontsGPU[a]i=0){
28)             int count = 0;
29)             for(int b=0;b;kObjectives;b++){
30)                 if(dataGPU[a][b] i= dataGPU[idx][b]) count++;
31)             }
32)             if(count == kObjectives) arrayFrontsGPU[idx] = -1;
33)         }
34)     }
35)     if (arrayFrontsGPU[idx] != -1) {
36)         arrayFrontsGPU[idx] = posFront;
37)         isAssignedFront[0] = true;
38)     }
39)     else arrayFrontsGPU[idx] = 0;
40) }
41) }
42)
43) __global__ void CompleteFront(float** dataGPU, int* arrayFrontsGPU, int NPoints, int
kObjectives, int posFront, bool* isEnd) {
44)     int idx = blockDim.x * blockIdx.x + threadIdx.x;
45)     if (arrayFrontsGPU[idx] == 0) {
46)         isEnd[0] = false;
47)         arrayFrontsGPU[idx] = posFront;
48)     }
49) }
50)
51) int* ObtainArrayFront(float **dataCPU, int NPoints, int kObjectives){
53)     /// arrayFrontsGPU -i-1 NoDominant ; 0 Initialice; n posFront
54)     int* arrayFrontsGPU;
55)     cudaMallocManaged(&arrayFrontsGPU,NPoints*sizeof(int));
56)     for(int a=0;a;iNPoints;a++) arrayFrontsGPU[a] =0;
57)
58)     float** dataGPU;
59)     cudaMallocManaged(&dataGPU,NPoints*sizeof(float*));
60)     for(int a=0;a ;iNPoints; a++){
61)         cudaMallocManaged(&dataGPU[a],kObjectives*sizeof(float));
62)         for(int b=0;b;kObjectives;b++){
63)             dataGPU[a][b] = dataCPU[a][b];
64)         }
65)     }
66)     bool* isEnd;

```

```

67)   cudaMallocManaged(&isEnd, sizeof(bool));
68)   isEnd[0] = false;
69)
70)   bool* isAssignedFront;
71)   cudaMallocManaged(&isAssignedFront, sizeof(bool));
72)   isAssignedFront[0] = false;
73)
74)   // inicialice front sequency
75)   int posFront = 0;
76)
77)   while(isEnd[0] == false && posFront<NPoints){
78)       posFront++;
79)
80)       isEnd[0] = true;
81)       isAssignedFront[0] = false;
82)       int sizeArray = NPoints;
83)       int threadsPerBlock = (sizeArray<BLOCK_SIZE*BLOCK_SIZE)?sizeArray:BLOCK_SIZE*BLOCK_SIZE;
// max 1024
84)       int blocksPerGrid = floor((((sizeArray) + threadsPerBlock - 1) / threadsPerBlock);
85)       InitSequency<<blocksPerGrid, threadsPerBlock>>(dataGPU, arrayFrontsGPU, NPoints, kObjectives, posFront, isEnd, isAssignedFront);
86)       cudaDeviceSynchronize();
87)       if (isAssignedFront[0] == false) {
88)           CompleteFront<<blocksPerGrid, threadsPerBlock >>(dataGPU, arrayFrontsGPU, NPoints, kObjectives, posFront, isEnd);
89)           cudaDeviceSynchronize();
90)       }
91)       return arrayFrontsGPU;
92) }
93)
94)
95) #endif
96)

```

B. Anexo: Algoritmo resolución de caminos en la estructura basada en primitivas

El algoritmo resuelve los caminos para la resolución de los grafos direccionales, por lo cual establece la conexión de los MUCOs y los aloja en la memoria dinámica de la GPU para reducir las soluciones según los casos específicos, y pasar de una programación secuencial a una programación en paralelo.

Se comienza generando una tabla de MUCOs adyacentes que representan la conectividad de los nodos en un grafo direccional como se aprecia en la figura **B-1**.

| | | | | |
|----|-----|-----|----|----|
| M0 | E1 | | | |
| M1 | E2 | E3 | | |
| M2 | | | | |
| M3 | E4 | E5 | E6 | E7 |
| M4 | E8 | | | |
| M5 | E9 | E10 | | |
| M6 | | | | |
| M7 | E11 | | | |
| M8 | | | | |

Figura B-1.: Tabla de MUCOs adyacentes.

Fuente: Creación Propia

Con la tabla de adyacencia se procede a la asignación de espacio dinámico en la memoria y a la realización paso a paso de la configuración inicial que se observa en la figura **B-2**. La primera celda de cada fila contiene el tamaño de las columnas del sistema que corresponde a la suma de los enlaces (edges) y nodos (MUCO) de la estructura de datos. Desde la segunda columna de la primer fila se empieza a almacenar la cantidad de caminos que genera la estructura de datos en un numero entero negativo.

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -S1 | -S2 | -S3 | -S4 | -S5 | -S6 | -S7 | -S8 | -S9 | -S10 | -S11 | -S12 | -S13 | -S14 |

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| | | | | | | | | | | | | | | |
| | M0 | M1 | M1 | M3 | M3 | M3 | M3 | M4 | M5 | M5 | M7 | | | |

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|-----|-----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | | | |
| | M0 | M1 | M1 | M3 | M3 | M3 | M3 | M4 | M5 | M5 | M7 | | | |

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|-----|-----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | | | |
| | M0 | M1 | M1 | M3 | M3 | M3 | M3 | M4 | M5 | M5 | M7 | M2 | M6 | M8 |

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|-----|-----|----|----|----|
| -14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | -1 | -1 | -1 |
| | M0 | M1 | M1 | M3 | M3 | M3 | M3 | M4 | M5 | M5 | M7 | M2 | M6 | M8 |

Figura B-2.: Alojamiento de memoria dinámica en la GPU.

Fuente: Creación Propia

En la segunda fila de cada uno de los siete arreglos secuenciales se almacena el tamaño de cada camino, nuevamente como un entero negativo. Como se observa en el tercer arreglo

todos los tamaños en un principio se establecen de tamaño dos. El tamaño es para representar todas las conexiones de la tabla de adyacencia de la figura **B-1**, en el cual la letra M hace referencia al MUCO y la letra E al edge o enlace. Los MUCOs que no cuentan con alguna conexión a otro MUCO se marcaran con -1 y es lo que se observa de color verde en el séptimo arreglo.

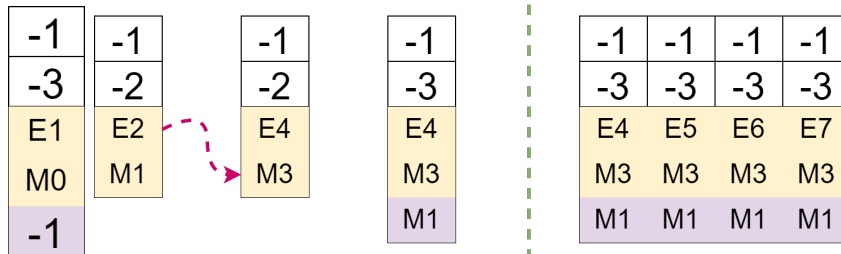


Figura B-3.: Posibilidad 1.
Fuente: Creación Propia

Al establecer el arreglo inicial se presentan las siguientes operaciones por cada columna para completar el camino que resuelve un grafo direccional. En la figura **B-3** se observa el caso en que se completa un camino se agrega una nueva celda a la columna y se termina con -1, al mismo tiempo que se cambia el tamaño del camino. En el caso que un MUCO raíz como lo sería M1 indique en su enlace una conexión con otro MUCO, como lo indica la conexión E2 con M3, todos los caminos que tengan el MUCO siguiente M3 tomara la información del camino previamente calculado y lo agregara a su camino.

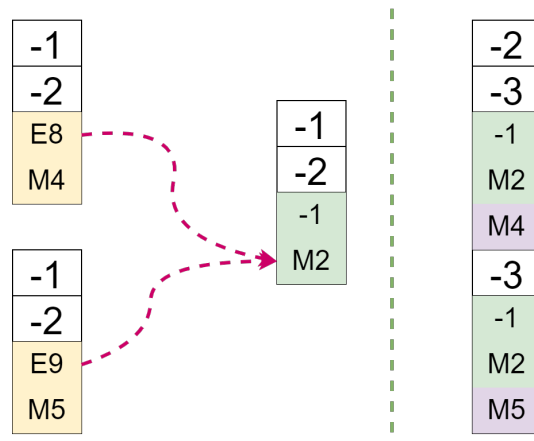


Figura B-4.: Posibilidad 2.
Fuente: Creación Propia

En el caso que un MUCO tenga conexión con dos MUCOs raíces como se observa en la figura **B-4**, se crean dos caminos en contenga cada uno la información del camino del MUCO previo.

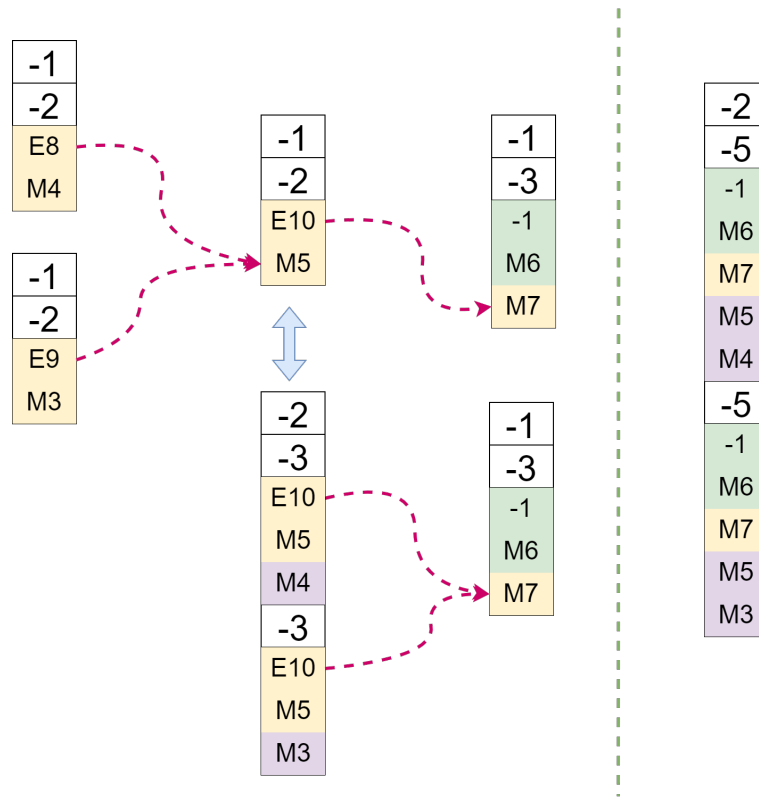


Figura B-5.: Posibilidad 3.

Fuente: Creación Propia

En un caso extendido como se muestra en la figura **B-5** se aplica la condición explicada anteriormente, pero con ambos caminos previos.

C. Anexo: Código G generativo para fresado en Y diámetro 6 mm

| | | |
|-----------------------------------|--------------|--------------|
| G00Z-20 | G01 X2Y9F80 | G01 X2Y11F80 |
| G18 G21(ZP XP PLANE, (FRESADO) | G01 Z0 F80 | G01 Z0 F80 |
| T2222 | G01 C144F80 | G01 C288F80 |
| M38 (ACTIVACION EJE C) | G01 X1Y9F80 | G01 X1Y11F80 |
| M113 S7000 | G01 Z0 F80 | G01 Z0 F80 |
| G98 (PER MINUTE FEED) | G01 C144F80 | G01 C288F80 |
| G00 C0 | G01 X1Y8F80 | G01 X1Y11F80 |
| Y20 | G01 Z0 F80 | G01 Z1 F80 |
| Z0 | G01 C144F80 | G01 C288F80 |
| G01 Y20F80 | G01 X1Y9F80 | G01 X1Y11F80 |
| (INICIO TRAYECTORIA) | G01 Z0 F80 | G01 Z2 F80 |
| G01 X0Y12F80 | G01 C144F80 | G01 C324F80 |
| G01 Z0 F80 | G01 X2Y9F80 | G01 X1Y11F80 |
| G01 C0F80 | G01 Z0 F80 | G01 Z3 F80 |
| G01 X1Y12F80 | G01 C180F80 | G01 C0F80 |
| G01 Z0 F80 | G01 X3Y9F80 | G01 X1Y11F80 |
| G01 C36F80 | G01 Z0 F80 | G01 Z4 F80 |
| G01 X1Y11F80 | G01 C216F80 | G01 C36F80 |
| G01 Z0 F80 | G01 X3Y10F80 | G01 X1Y11F80 |
| G01 C72F80 | G01 Z0 F80 | G01 Z5 F80 |
| G01 X2Y11F80 | G01 C252F80 | G01 C72F80 |
| G01 Z0 F80 | G01 X4Y10F80 | G01 X0Y11F80 |
| G01 C108F80 | G01 Z0 F80 | G01 Z5 F80 |
| G01 X2Y10F80 | G01 C288F80 | G01 C72F80 |
| G01 Z0 F80 | G01 X3Y10F80 | G01 X0Y11F80 |
| G01 C144F80 | G01 Z0 F80 | G01 Z6 F80 |
| G01 X2Y10F80 | G01 C288F80 | G01 C72F80 |
| G01 Z0 F80 | G01 X2Y10F80 | G01 X0Y11F80 |
| G01 C144F80 | G01 Z0 F80 | G01 Z5 F80 |
| | G01 C288F80 | G01 C72F80 |

| | | |
|---------------|--------------|---------------|
| G01 X0Y11F80 | G01 Z0 F80 | G01 C180F80 |
| G01 Z4 F80 | G01 C0F80 | G01 X0Y10F80 |
| G01 C108F80 | G01 X2Y8F80 | G01 Z5 F80 |
| G01 X0Y11F80 | G01 Z0 F80 | G01 C180F80 |
| G01 Z3 F80 | G01 C0F80 | G01 X0Y10F80 |
| G01 C144F80 | G01 X3Y8F80 | G01 Z6 F80 |
| G01 X0Y11F80 | G01 Z0 F80 | G01 C180F80 |
| G01 Z2 F80 | G01 C0F80 | G01 X0Y10F80 |
| G01 C180F80 | G01 X3Y9F80 | G01 Z5 F80 |
| G01 X0Y11F80 | G01 Z0 F80 | G01 C180F80 |
| G01 Z1 F80 | G01 C0F80 | G01 X0Y10F80 |
| G01 C216F80 | G01 X4Y9F80 | G01 Z4 F80 |
| G01 X-1Y11F80 | G01 Z0 F80 | G01 C216F80 |
| G01 Z1 F80 | G01 C0F80 | G01 X0Y10F80 |
| G01 C216F80 | G01 X3Y9F80 | G01 Z3 F80 |
| G01 X-1Y11F80 | G01 Z0 F80 | G01 C252F80 |
| G01 Z0 F80 | G01 C36F80 | G01 X0Y10F80 |
| G01 C216F80 | G01 X2Y9F80 | G01 Z2 F80 |
| G01 X-1Y10F80 | G01 Z0 F80 | G01 C288F80 |
| G01 Z0 F80 | G01 C36F80 | G01 X0Y10F80 |
| G01 C216F80 | G01 X2Y10F80 | G01 Z1 F80 |
| G01 X0Y10F80 | G01 Z0 F80 | G01 C324F80 |
| G01 Z0 F80 | G01 C36F80 | G01 X-1Y10F80 |
| G01 C252F80 | G01 X1Y10F80 | G01 Z1 F80 |
| G01 X0Y9F80 | G01 Z0 F80 | G01 C324F80 |
| G01 Z0 F80 | G01 C36F80 | G01 X-1Y10F80 |
| G01 C288F80 | G01 X1Y10F80 | G01 Z0 F80 |
| G01 X1Y9F80 | G01 Z1 F80 | G01 C324F80 |
| G01 Z0 F80 | G01 C36F80 | G01 X-1Y10F80 |
| G01 C324F80 | G01 X1Y10F80 | G01 Z1 F80 |
| G01 X1Y8F80 | G01 Z2 F80 | G01 C324F80 |
| G01 Z0 F80 | G01 C72F80 | G01 X-1Y10F80 |
| G01 C0F80 | G01 X1Y10F80 | G01 Z2 F80 |
| G01 X1Y8F80 | G01 Z3 F80 | G01 C0F80 |
| G01 Z0 F80 | G01 C108F80 | G01 X-1Y10F80 |
| G01 C0F80 | G01 X1Y10F80 | G01 Z3 F80 |
| G01 X1Y7F80 | G01 Z4 F80 | G01 C36F80 |
| G01 Z0 F80 | G01 C144F80 | G01 X-1Y10F80 |
| G01 C0F80 | G01 X1Y10F80 | G01 Z4 F80 |
| G01 X1Y8F80 | G01 Z5 F80 | G01 C72F80 |

| | | |
|---------------|---------------|-------------------------|
| G01 X-1Y10F80 | G01 X-3Y10F80 | G01 X-5Y10F80 |
| G01 Z5 F80 | G01 Z2 F80 | G01 Z1 F80 |
| G01 C108F80 | G01 C288F80 | G01 C180F80 |
| G01 X-2Y10F80 | G01 X-3Y10F80 | G01 X-5Y10F80 |
| G01 Z5 F80 | G01 Z3 F80 | G01 Z0 F80 |
| G01 C108F80 | G01 C324F80 | G01 C180F80 |
| G01 X-2Y10F80 | G01 X-3Y10F80 | G01 Y15 |
| G01 Z6 F80 | G01 Z4 F80 | G00 Y20 |
| G01 C108F80 | G01 C0F80 | X0 |
| G01 X-2Y10F80 | G01 X-3Y10F80 | Z-20 |
| G01 Z5 F80 | G01 Z5 F80 | M115(APAGAR HERRA- |
| G01 C108F80 | G01 C36F80 | MIENTAS) |
| G01 X-2Y10F80 | G01 X-4Y10F80 | M39(CANCELACION EJE |
| G01 Z4 F80 | G01 Z5 F80 | C) |
| G01 C144F80 | G01 C36F80 | M39 |
| G01 X-2Y10F80 | G01 X-4Y10F80 | (TRONZADO FINAL) |
| G01 Z3 F80 | G01 Z6 F80 | G00 Z-20 |
| G01 C180F80 | G01 C36F80 | G18 G21 G99 (ZP XP PLA- |
| G01 X-2Y10F80 | G01 X-4Y10F80 | NE, , PER REVOLUTION |
| G01 Z2 F80 | G01 Z5 F80 | FEED) |
| G01 C216F80 | G01 C36F80 | T0101 |
| G01 X-2Y10F80 | G01 X-4Y10F80 | G00 Z10 |
| G01 Z1 F80 | G01 Z4 F80 | G00 X30 |
| G01 C252F80 | G01 C72F80 | G99 |
| G01 X-3Y10F80 | G01 X-4Y10F80 | M03 S1500 |
| G01 Z1 F80 | G01 Z3 F80 | G01 X-0.5 F0.05 |
| G01 C252F80 | G01 C108F80 | G00 X30 |
| G01 X-3Y10F80 | G01 X-4Y10F80 | G50 Z0 |
| G01 Z0 F80 | G01 Z2 F80 | G00 Z-20 |
| G01 C252F80 | G01 C144F80 | M05 |
| G01 X-3Y10F80 | G01 X-4Y10F80 | (FINALIZAR PROGRAMA) |
| G01 Z1 F80 | G01 Z1 F80 | M30 |
| G01 C252F80 | G01 C180F80 | |

D. Anexo: Código G generativo para fresado en Z diámetro 6mm

| | | |
|------------------------------------|----------------|----------------|
| G00Z-20 | G01 C36F80 | G01 C0F80 |
| G18 G21(ZP XP PLANE,) (FRESADO) | G01 X-6Y0F80 | G01 X-11Y-2F80 |
| T2828 | G01 Z-1 F80 | G01 Z2 F80 |
| M38 (ACTIVACION EJE C) | G01 C72F80 | G01 C36F80 |
| M113 S7000 | G01 X-7Y0F80 | G01 X-11Y-3F80 |
| G98 (PER MINUTE FEED) | G01 Z-1 F80 | G01 Z2 F80 |
| G00 C0 | G01 C108F80 | G01 C72F80 |
| Y20 | G01 X-7Y0F80 | G01 X-11Y-3F80 |
| Z0 | G01 Z0 F80 | G01 Z3 F80 |
| G01 Y20F80 | G01 C144F80 | G01 C108F80 |
| (INICIO TRAYECTORIA) | G01 X-8Y0F80 | G01 X-11Y-4F80 |
| G00 Z-2 | G01 Z0 F80 | G01 Z3 F80 |
| G01 X-1Y0F80 | G01 C180F80 | G01 C144F80 |
| G01 Z-2 F80 | G01 X-9Y0F80 | G01 X-11Y-5F80 |
| G01 C0F80 | G01 Z0 F80 | G01 Z3 F80 |
| G01 X-2Y0F80 | G01 C216F80 | G01 C180F80 |
| G01 Z-2 F80 | G01 X-9Y0F80 | G01 X-11Y-5F80 |
| G01 C0F80 | G01 Z1 F80 | G01 Z4 F80 |
| G01 X-3Y0F80 | G01 C252F80 | G01 C216F80 |
| G01 Z-2 F80 | G01 X-10Y0F80 | G01 X-11Y-6F80 |
| G01 C0F80 | G01 Z1 F80 | G01 Z4 F80 |
| G01 X-4Y0F80 | G01 C288F80 | G01 C252F80 |
| G01 Z-2 F80 | G01 X-11Y0F80 | G01 X-10Y-6F80 |
| G01 C0F80 | G01 Z1 F80 | G01 Z3 F80 |
| G01 X-5Y0F80 | G01 C324F80 | G01 C252F80 |
| G01 Z-2 F80 | G01 X-11Y0F80 | G01 X-10Y-5F80 |
| G01 C0F80 | G01 Z2 F80 | G01 Z3 F80 |
| G01 X-5Y0F80 | G01 C324F80 | G01 C288F80 |
| G01 Z-1 F80 | G01 X-11Y-1F80 | G01 X-9Y-5F80 |
| | G01 Z2 F80 | G01 Z3 F80 |

| | | |
|---------------|----------------|----------------|
| G01 C324F80 | G01 X-9Y-5F80 | G01 Z3 F80 |
| G01 X-9Y-4F80 | G01 Z3 F80 | G01 C144F80 |
| G01 Z3 F80 | G01 C72F80 | G01 X-6Y-2F80 |
| G01 C0F80 | G01 X-10Y-5F80 | G01 Z3 F80 |
| G01 X-8Y-4F80 | G01 Z4 F80 | G01 C144F80 |
| G01 Z3 F80 | G01 C108F80 | G01 X-7Y-2F80 |
| G01 C36F80 | G01 X-10Y-6F80 | G01 Z3 F80 |
| G01 X-8Y-3F80 | G01 Z4 F80 | G01 C180F80 |
| G01 Z3 F80 | G01 C144F80 | G01 X-7Y-3F80 |
| G01 C72F80 | G01 X-11Y-6F80 | G01 Z3 F80 |
| G01 X-7Y-3F80 | G01 Z4 F80 | G01 C216F80 |
| G01 Z3 F80 | G01 C180F80 | G01 X-8Y-3F80 |
| G01 C108F80 | G01 X-10Y-6F80 | G01 Z3 F80 |
| G01 X-7Y-2F80 | G01 Z3 F80 | G01 C252F80 |
| G01 Z3 F80 | G01 C180F80 | G01 X-8Y-4F80 |
| G01 C144F80 | G01 X-10Y-5F80 | G01 Z3 F80 |
| G01 X-6Y-2F80 | G01 Z3 F80 | G01 C288F80 |
| G01 Z3 F80 | G01 C216F80 | G01 X-9Y-4F80 |
| G01 C180F80 | G01 X-9Y-5F80 | G01 Z3 F80 |
| G01 X-6Y-1F80 | G01 Z3 F80 | G01 C324F80 |
| G01 Z3 F80 | G01 C252F80 | G01 X-9Y-5F80 |
| G01 C216F80 | G01 X-9Y-4F80 | G01 Z3 F80 |
| G01 X-6Y-2F80 | G01 Z3 F80 | G01 C0F80 |
| G01 Z3 F80 | G01 C288F80 | G01 X-10Y-5F80 |
| G01 C216F80 | G01 X-8Y-4F80 | G01 Z4 F80 |
| G01 X-7Y-2F80 | G01 Z3 F80 | G01 C36F80 |
| G01 Z3 F80 | G01 C324F80 | G01 X-10Y-6F80 |
| G01 C252F80 | G01 X-8Y-3F80 | G01 Z4 F80 |
| G01 X-7Y-3F80 | G01 Z3 F80 | G01 C72F80 |
| G01 Z3 F80 | G01 C0F80 | G01 X-11Y-6F80 |
| G01 C288F80 | G01 X-7Y-3F80 | G01 Z4 F80 |
| G01 X-8Y-3F80 | G01 Z3 F80 | G01 C108F80 |
| G01 Z3 F80 | G01 C36F80 | G01 X-10Y-6F80 |
| G01 C324F80 | G01 X-7Y-2F80 | G01 Z3 F80 |
| G01 X-8Y-4F80 | G01 Z3 F80 | G01 C108F80 |
| G01 Z3 F80 | G01 C72F80 | G01 X-10Y-5F80 |
| G01 C0F80 | G01 X-6Y-2F80 | G01 Z3 F80 |
| G01 X-9Y-4F80 | G01 Z3 F80 | G01 C144F80 |
| G01 Z3 F80 | G01 C108F80 | G01 X-9Y-5F80 |
| G01 C36F80 | G01 X-6Y-1F80 | G01 Z3 F80 |

| | | |
|---------------|----------------|-------------------------|
| G01 C180F80 | G01 Z3 F80 | G01 Y15 |
| G01 X-9Y-4F80 | G01 C108F80 | G00 Y20 |
| G01 Z3 F80 | G01 X-7Y-3F80 | X0 |
| G01 C216F80 | G01 Z3 F80 | Z-20 |
| G01 X-8Y-4F80 | G01 C144F80 | M115(APAGAR HERRA- |
| G01 Z3 F80 | G01 X-8Y-3F80 | MIENTAS) |
| G01 C252F80 | G01 Z3 F80 | M39(CANCELACION EJE |
| G01 X-8Y-3F80 | G01 C180F80 | C) |
| G01 Z3 F80 | G01 X-8Y-4F80 | M39 |
| G01 C288F80 | G01 Z3 F80 | (TRONZADO FINAL) |
| G01 X-7Y-3F80 | G01 C216F80 | G00 Z-20 |
| G01 Z3 F80 | G01 X-9Y-4F80 | G18 G21 G99 (ZP XP PLA- |
| G01 C324F80 | G01 Z3 F80 | NE, , PER REVOLUTION |
| G01 X-7Y-2F80 | G01 C252F80 | FEED) |
| G01 Z3 F80 | G01 X-9Y-5F80 | T0101 |
| G01 C0F80 | G01 Z3 F80 | G00 Z10 |
| G01 X-6Y-2F80 | G01 C288F80 | G00 X30 |
| G01 Z3 F80 | G01 X-10Y-5F80 | G99 |
| G01 C36F80 | G01 Z4 F80 | M03 S1500 |
| G01 X-6Y-1F80 | G01 C324F80 | G01 X-0.5 F0.05 |
| G01 Z3 F80 | G01 X-10Y-6F80 | G00 X30 |
| G01 C72F80 | G01 Z4 F80 | G50 Z0 |
| G01 X-6Y-2F80 | G01 C0F80 | G00 Z-20 |
| G01 Z3 F80 | G01 X-11Y-6F80 | M05 |
| G01 C72F80 | G01 Z4 F80 | (FINALIZAR PROGRAMA) |
| G01 X-7Y-2F80 | G01 C36F80 | M30 |

E. Aplicación en LabFabEx-UN del diseño generativo secuencial

El esquema de trabajo de validación en el LabFabEx-UN se presenta en la replicación funcional de los algoritmos planteados para el desarrollo de Grippers en el laboratorio. Se registran el planteamiento de procesos para llevar a cabo la validación, que siguen el esquema de aplicación que se presenta en la figura E-1.

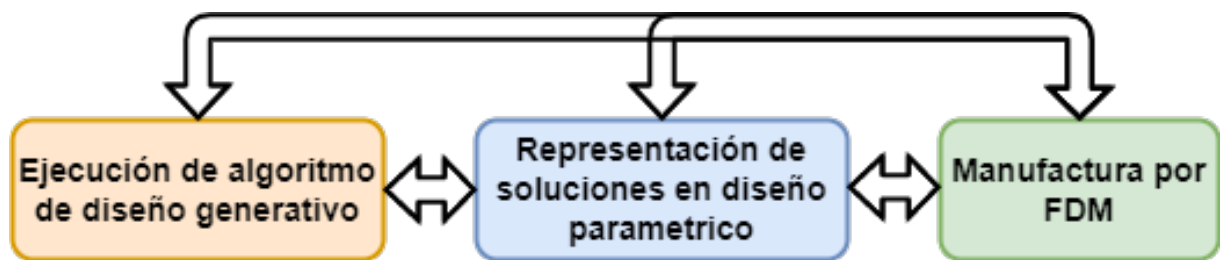


Figura E-1.: Esquema de aplicación en el LabFabEx-UN.

Fuente: Creación propia

El proceso de validación fue realizado por cuatro estudiantes de últimos semestres de la carrera de mecatrónica de la Universidad Nacional de Colombia, con la finalidad de cumplir con los requerimientos necesarios para ejecutar e integrar todas las etapas. En este primer proceso el algoritmo generativo a desarrollar se basa en la estructura de datos basado en el modelamiento matemático, que implica un desarrollo de programación secuencial que se puede ejecutar en cualquier equipo del Laboratorio.

Los parámetros para la configuración del algoritmo se observan en la figura E-2, en el cual el punto objetivo de referencia es el punto al cual la cadena de eslabones debe aproximarse con una inclinación de 45° en el ultimo eslabón. La variación de las generaciones o la población del algoritmo NSGAI que permite modificar la estructura de datos para evaluarla y del algoritmo genético para generar la variación de los ángulos se refleja en el tiempo de procesamiento y en la convergencia de las soluciones.

```
##### Configuración #####
# Definir los paremtros del MSGAII
sizePopulation = 20 #par 40
generations = 10 #200
maxLinks = 5 #10

# Definir los parametros del GA 4 angles
sizePopulationGA = 30 #par
generationsGA = 6
factorParentsSelect=2 #sizePopulationGA/factorParentsSelect

# Punto objetivo
pointRef = [10,40]

# Limites del espacio de busqueda
# largo , ancho , angulo eslabon
limits = [[10,60],[10,30],[0,360]]
posInit = [0,0]
# Los valores decimales los convierte a decimales
#valbit = 256 #8bit
valbit = 65536 #16bit
#####
```

Figura E-2.: Parámetros de configuración del algoritmo generativo secuencial.

Fuente: Creación propia

En la figura E-3 se observa un ejemplo de como se pueden representar gráficamente las soluciones en cantidad de eslabones, que en este caso empieza en soluciones de 2 eslabones hasta 4 eslabones, Las características de longitud. Esta representación ayuda al diseñador a comprender las soluciones que se generan en la consola y que estan expresadas en matrices para N soluciones de tamaño $N \times (Eslabones * 3)$. En estas matrices se destacan valores de ancho del eslabón y el ángulo para cumplir con la aproximación del ultimo eslabón a 45° que no se representan gráficamente en las soluciones.

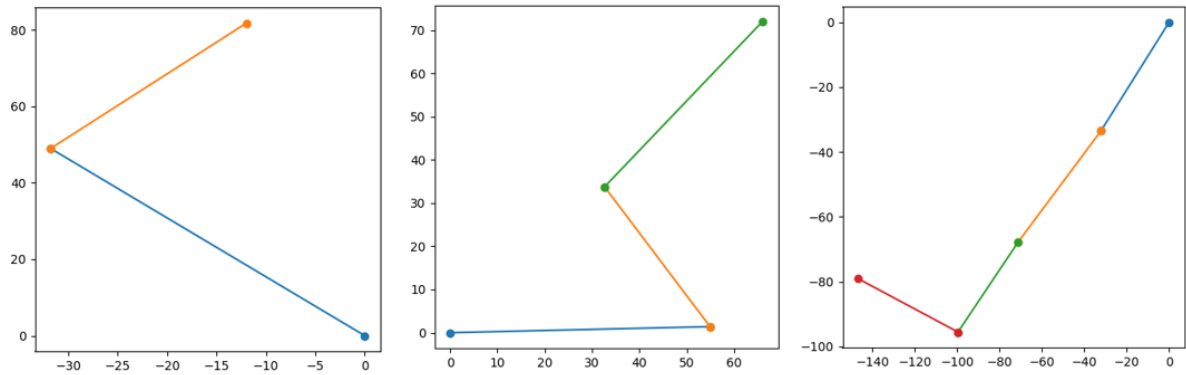


Figura E-3.: Ejemplificación de las soluciones generadas por el algoritmo.
Fuente: Creación propia

Las soluciones seleccionadas se encuentran contempladas en la tabla E-1. En esta se destaca que la solución seleccionada por el estudiante Fabian contempla tres eslabones, también la solución que selecciona Alejandro implica cambiar mas parámetros de ancho en la figura, que en una aproximación inicial se plantea mantener el ancho de mm.

Tabla E-1.: Soluciones seleccionadas.

| Estudiante | | Eslabón 1 [mm] | Eslabón 2 [mm] | Eslabón 3 [mm] |
|------------|-------|----------------|----------------|----------------|
| Jhoan | largo | 37.5 | 32.3 | |
| | ancho | 15 | 15 | |
| Fabian | largo | 34.4 | 30.1 | 19.3 |
| | ancho | 15 | 15 | 15 |
| Camilo | largo | 51 | 18.2 | |
| | ancho | 15 | 15 | |
| Alejandro | largo | 50 | 43 | |
| | ancho | 25 | 19 | |

Luego de obtener las soluciones del algoritmo de diseño generativo se procede a la segunda etapa del esquema que consiste representar la solución seleccionada en fusión 360. En este paso se provee de un modelo de eslabón parametrizado y cada estudiante debe incluir en el ensamblaje una cantidad igual a la cantidad de eslabones que indique la solución seleccionada. En la figura E-4 se observa la respuesta a la integración según el parámetro de numero de eslabón que responde a un ancho mayor para asegurar el acople con el eslabón previo.

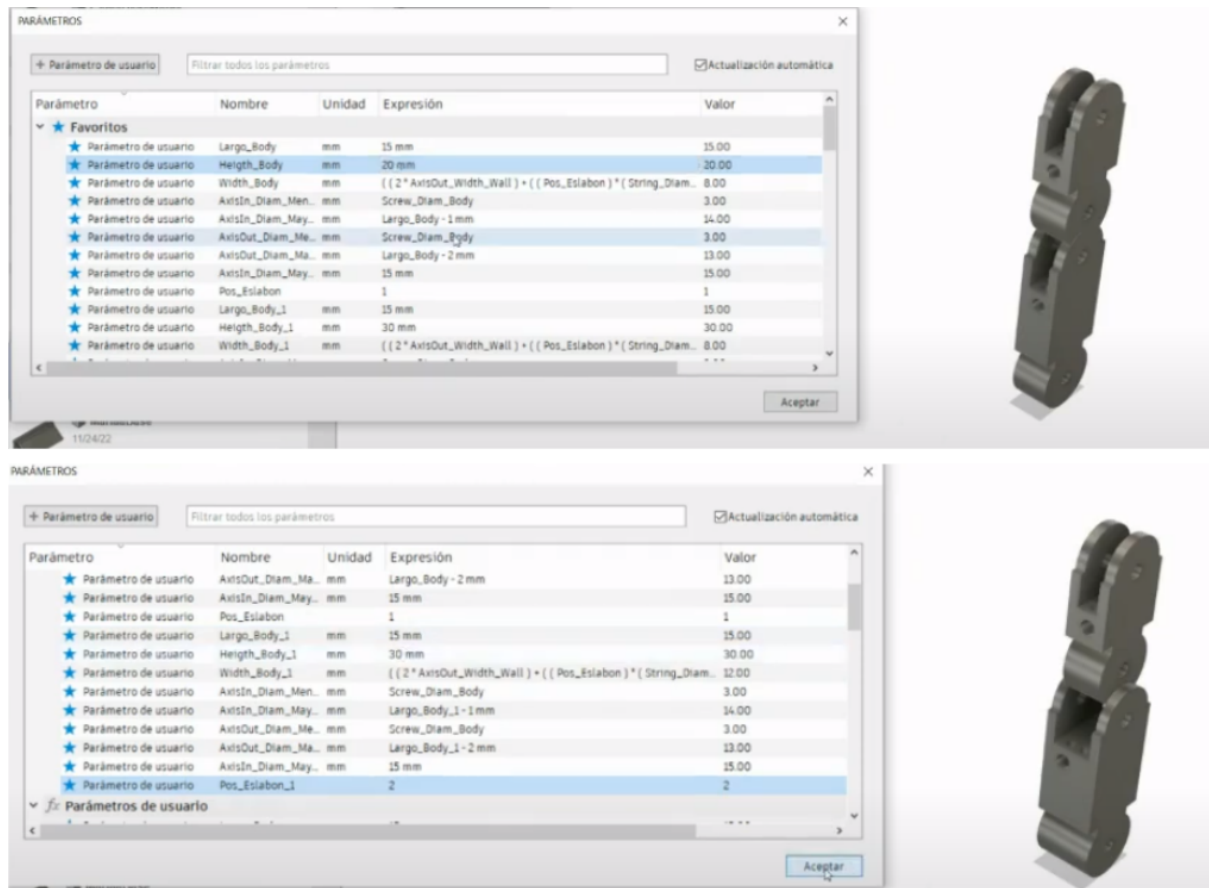


Figura E-4.: Modificación de parámetros en Fusion360.

Fuente: Creación propia

Para la última etapa del esquema planteado referente a la manufactura, se hace uso del modelado CAD para la generación de archivos STL que es el insumo para el software de CURA que genera el código G para el proceso de FDM. Las soluciones son escaladas a un 50 % con la finalidad de ser impresas todas al mismo tiempo en la maquinaria del laboratorio y responder a disminuir tiempo de fabricación de las cadenas de eslabones como se observa en la figura E-5.



Figura E-5.: Plancha de impresión 3D.

Fuente: Creación propia

La manufactura por adición material debe pasar por un proceso de retirar el material soporte generado por la maquinaria para cumplir con la geometría de la pieza, por tal motivo es el único proceso adicional que debe hacerse antes del ensamblaje, por lo tanto en la figura **E-6** se observa los eslabones luego del proceso de remoción del material soporte.



Figura E-6.: Eslabones impresos 3D.

Fuente: Creación propia

Con la idea de completar el proceso de ensamblaje por medio de impresión 3d, se procede

a realizar mediciones de los agujeros para los ejes de cada uno de las cadenas de eslabones. Por medio de la maquina Zoller se toman el siguiente registro fotografico del espacio para los ejes en los eslabones como se aprecia en las figuras **E-7**, **E-8**, **E-9** y **E-10**.

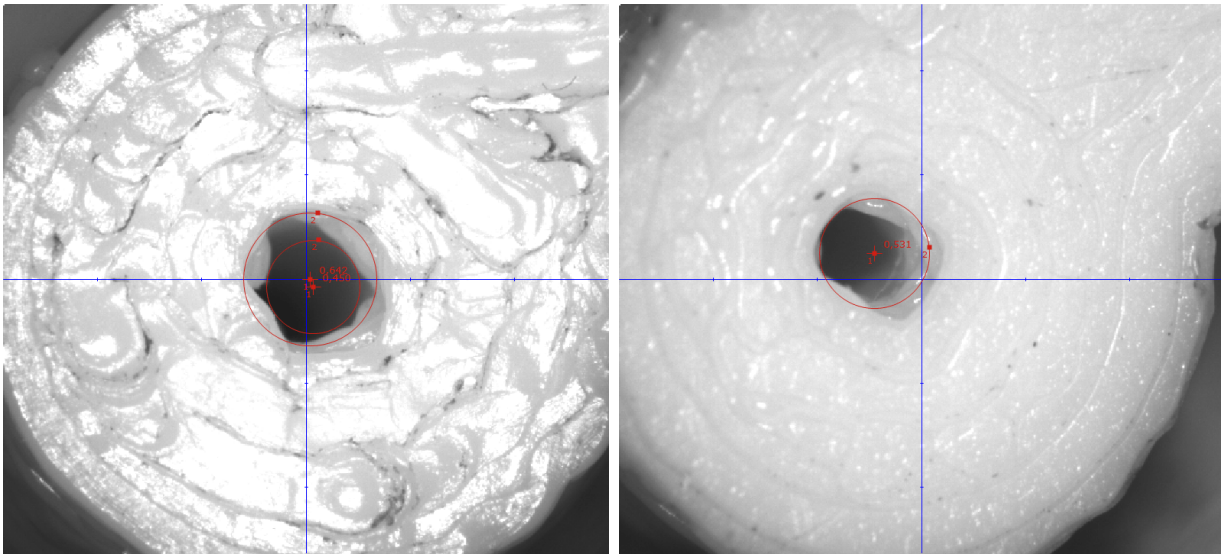


Figura E-7.: Medición radios en dos probetas- Estudiante: Jhoan.
Fuente: Creación propia

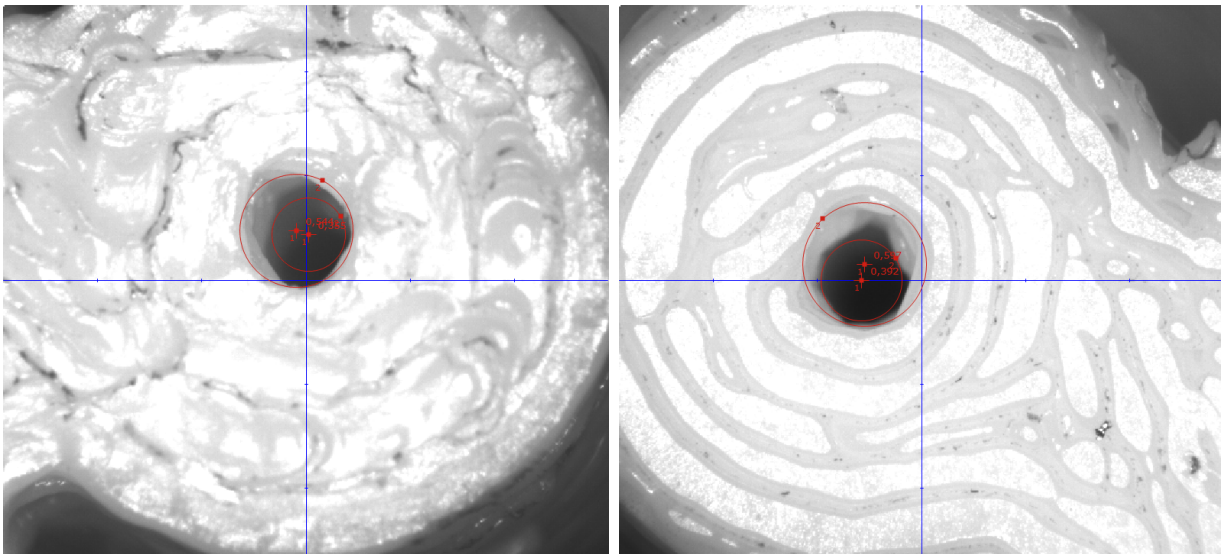


Figura E-8.: Medición radios en dos probetas- Estudiante: Fabian.
Fuente: Creación propia

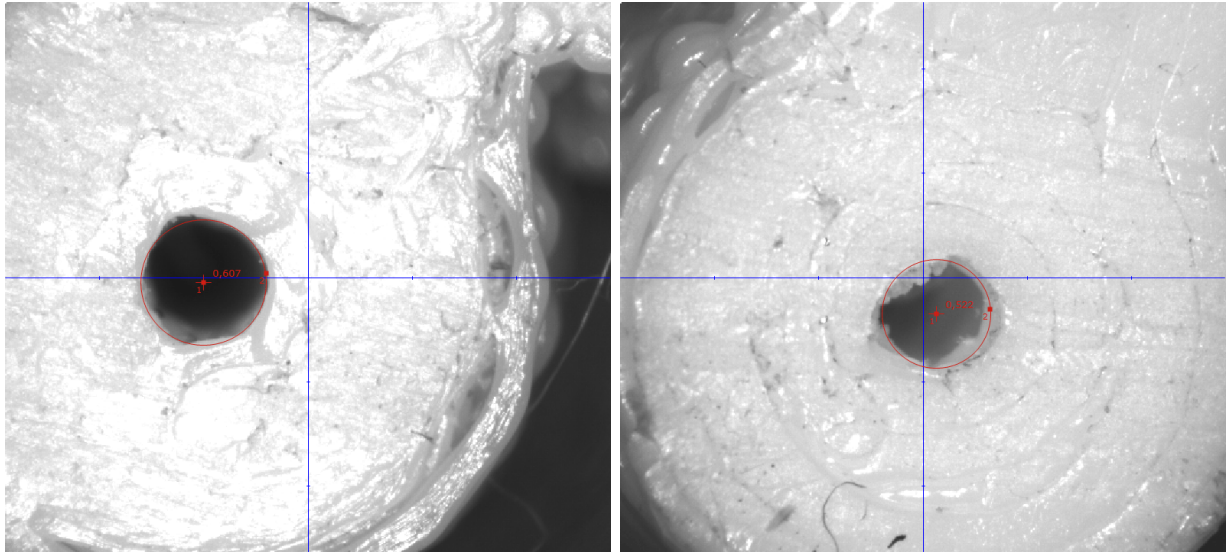


Figura E-9.: Medición radios en dos probetas- Estudiante: Camilo.
Fuente: Creación propia

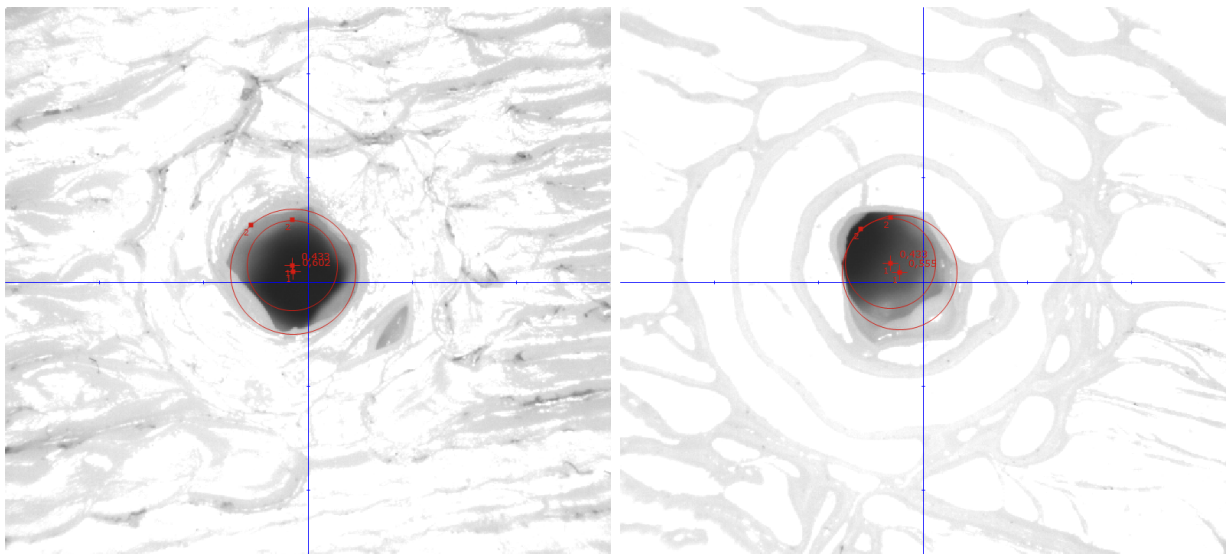


Figura E-10.: Medición radios en dos probetas- Estudiante: Camilo.
Fuente: Creación propia

Con el registro fotográfico se aclara en la tabla **E-2** el diámetro efectivo que deben tener los ejes para permitir un correcto ensamblaje de los eslabones. Como se observa cada estudiante utiliza diferentes valores de tolerancia para evidenciar como el modelo paramétrico puede ajustarse a distinta maquinaria.

En la tabla **E-3** se observa los valores obtenidos por la maquina Zoller y el calculo del

Tabla E-2.: Diámetros efectivos para los ejes.

| Estudiante | Diámetro efectivo [mm] | Tolerancia [mm] | Diámetro Final Escalamiento 50 % [mm] |
|------------|------------------------|-----------------|---------------------------------------|
| Jhoan | 3 | 0.35 | 1.675 |
| Fabian | 3 | 0.45 | 1.725 |
| Camilo | 3 | 0.4 | 1.7 |
| Alejandro | 3 | 0.45 | 1.725 |

Error para las dos probetas respecto al diámetro efectivo calculado. Los errores calculados se encuentran en un rango de 0.391mm a 0.656mm.

Tabla E-3.: Mediciones en maquina Zoller.

| Estudiante | Radio menor P1 [mm] | Radio mayor P1 [mm] | Error P1 [mm] | Radio menor P2 [mm] | Radio Mayor P2 [mm] | Error P2 [mm] |
|------------|---------------------|---------------------|---------------|---------------------|---------------------|---------------|
| Jhoan | | 0.531 | 0.613 | 0.450 | 0.642 | 0.391 |
| Fabian | 0.355 | 0.544 | 0.637 | 0.392 | 0.597 | 0.531 |
| Camilo | | 0.522 | 0.656 | | 0.607 | 0.486 |
| Alejandro | 0.433 | 0.555 | 0.615 | 0.433 | 0.602 | 0.521 |

Con lo anterior en mente se realiza la impresión de seis probetas para los ejes como se aprecia en la figura E-11. Las tolerancia van de 0.7 mm hasta 0.2mm de izquierda a derecha. Debido a que se espera por el error que los ejes que se ajusten se encuentren entre 0.4 y 0.7 mm.

**Figura E-11.:** Probetas de tolerancia 0.2mm a 0.7 mm.

Fuente: Creación propia

Con estas probetas se desarrolla la ensamblabilidad de las cadenas de eslabones como se aprecia en la figura **E-12**. Este ejemplo consiste en el ensamblaje del estudiante Jhoan en el cual usa pines con tolerancia de 0,6mm y 0,7mm. Como resultado se observa que ambos cumplen con las función pero con el pin de 0,6mm se presenta mayor resistencia a la rotación.

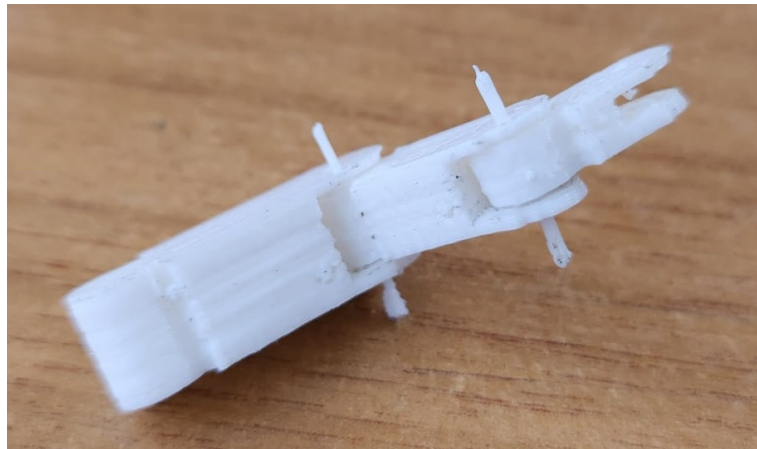


Figura E-12.: Ensamblaje de pines con tolerancia de 0.6 y 0.7 mm.

Fuente: Creación propia

El mismo comportamiento se presenta en los ensamblajes de los demás estudiantes que se muestran en la figura **E-13**. Debido al diámetro de las características del agujero pasante y los pines, se observa que su geometría de manufactura no asegura cilindros perfectos, por lo cual estas irregulares entran en contacto ejerciendo mayor fuerza de fricción al rotar los eslabones, por lo cual queda a decisión del diseñador la selección de los pines según la necesidad específica que desee se cumpla.



Figura E-13.: Ensamblaje de pines con tolerancia de 0.6 y 0.7 mm de todos los estudiantes.

Fuente: Creación propia



Figura E-14.: Estudiantes participantes.

Fuente: Creación propia

F. Protocolo despliegue del Gripper en el LabFabEx-UN

En esta guía se desarrollan los conceptos y procesos necesarios para llevar a cabo un ejercicio de integración del Gripper, diseñado por medio del algoritmo de diseño generativo, con el robot SCARA que se encuentra integrado con la plataforma PRIA-UN.

Se parte de tener manufacturado y ensamblado el Gripper como se aprecia en la siguiente figura **F-1**, se hace uso de un servidor Raspberry PI 4, para generar la integración de un nuevo robot con PRIA-UN.

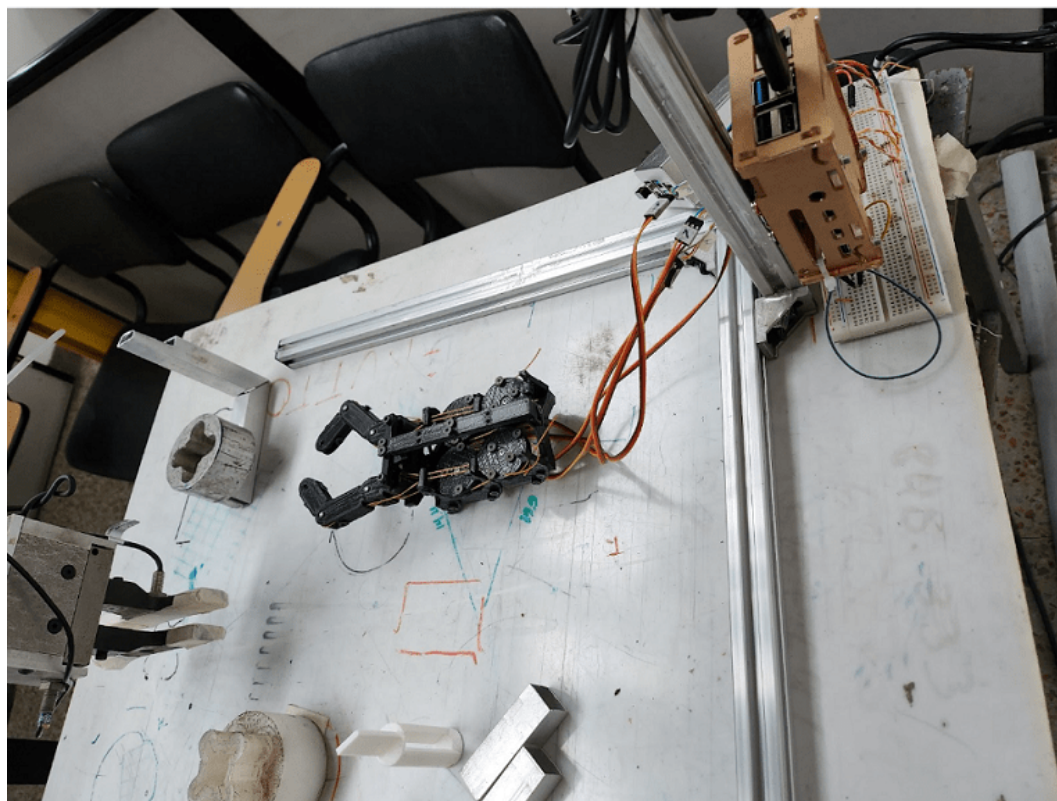


Figura F-1.: Gripper ensamblado en el espacio de trabajo.
Fuente: Creación propia

Por medio de la plataforma PRIA-UN se lleva a cabo la secuencia para la apertura del efector final del robot tipo SCARA del laboratorio por medio de la interfaz de uso NodeRED, con el siguiente bloque.

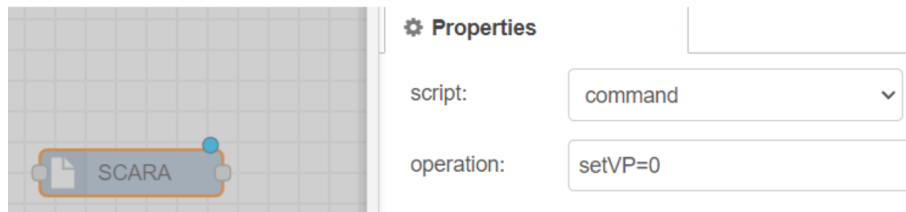


Figura F-2.: Comando robot SCARA apertura del efector final.

Fuente: Creación propia

Para la sujeción del gripper en el robot SCARA es necesario ubicarlo en el efector final, al efectuar la apertura del efector final del SCARA y generando el agarre del efector final con el cuerpo del gripper como se muestra a continuación.

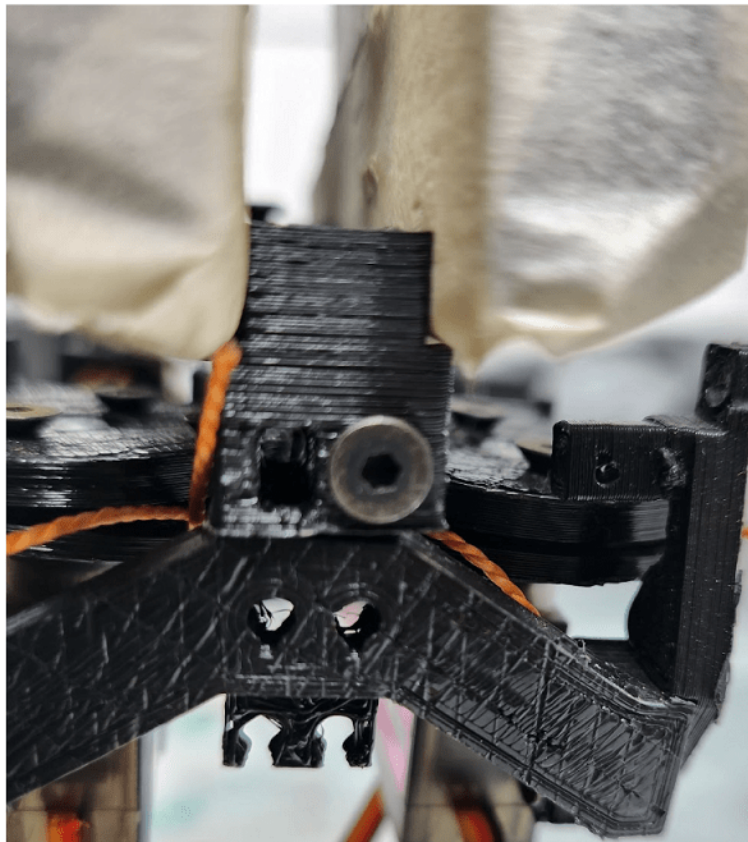


Figura F-3.: Vista trasera sujeción del Gripper.

Fuente: Creación propia

Se aclara que el Gripper tiene una pestaña en la parte trasera donde indica donde debe hacer contacto la superficie del Gripper, en este proceso puede ser necesaria la intervención de dos personas para asegurar el correcto ajuste del Gripper. Se adiciona como se ubica el Gripper en una vista lateral.

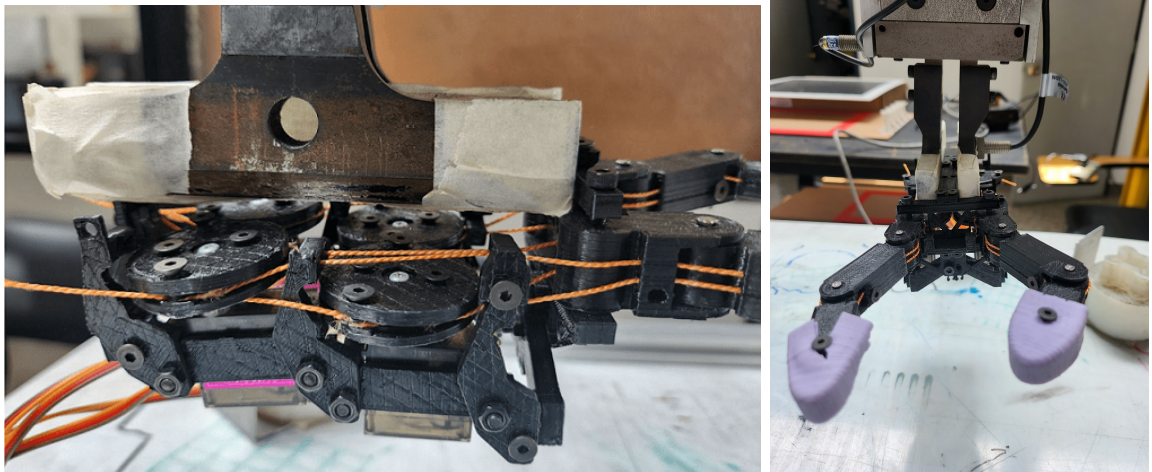


Figura F-4.: Vista lateral y frontal sujeción del Gripper.
Fuente: Creación propia

Para la integración del robot con PRIA-UN se remite la siguiente arquitectura que contiene la implementación con Docker y como reduce la complejidad de la configuración del robot con la base de datos Firebase que controla el sistema MES.

Para conectar el Gripper con PRIA es el mismo mecanismo que los demás robots integrados, es necesario entrar a la consola del servidor y habilitarlo al sistema.

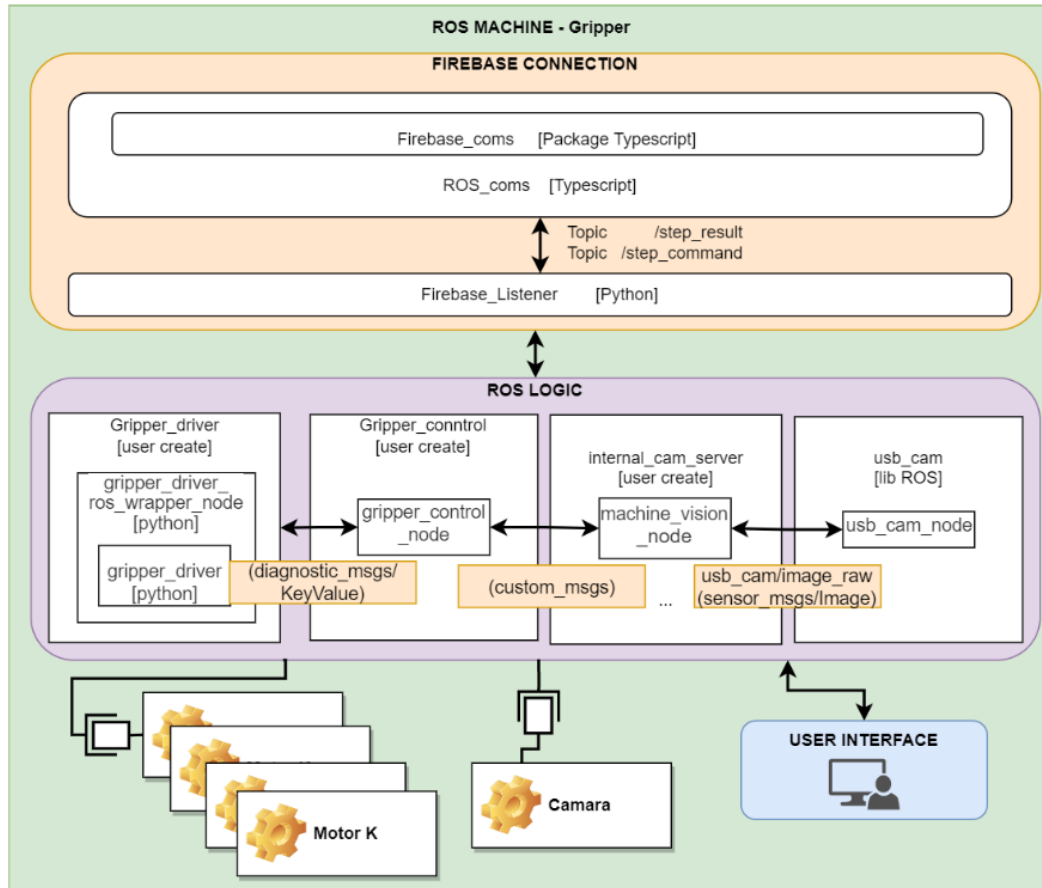


Figura F-5.: Arquitectura de conexión con PRIA-UN en RPI4.

Fuente: Creación propia

Se accede al servidor portátil Raspberry Pi 4, en este caso por VNC.

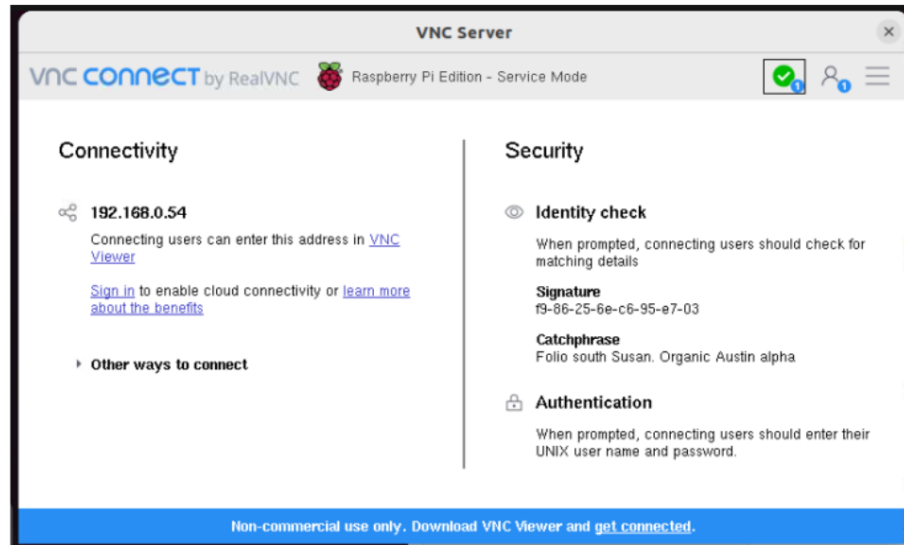


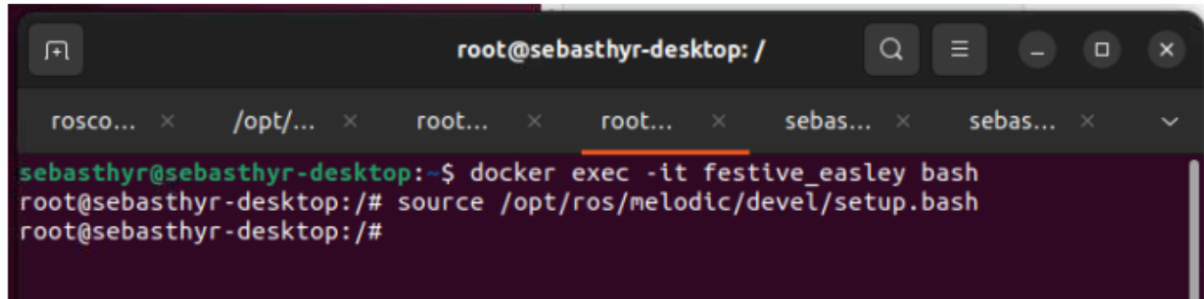
Figura F-6.: Conexión VNC.
Fuente: Creación propia

Se abre una ventana de consola y se arranca la imagen de docker que contiene el programa.

```
sebasthyr@sebasthyr-desktop: ~  
sebasthyr@sebasthyr-desktop:~$ xhost +local:docker  
non-network local connections being added to access control list  
sebasthyr@sebasthyr-desktop:~$ docker start festive_easley  
festive_easley  
sebasthyr@sebasthyr-desktop:~$
```

Figura F-7.: Comenzar a ejecutar en RPI la imagen de docker.
Fuente: Creación propia

Se recomienda abrir nuevas ventanas e ingresar a la consola de la imagen de docker por medio de los siguientes comandos y se habilita el bash de ROS para que lea todos los comandos especiales.

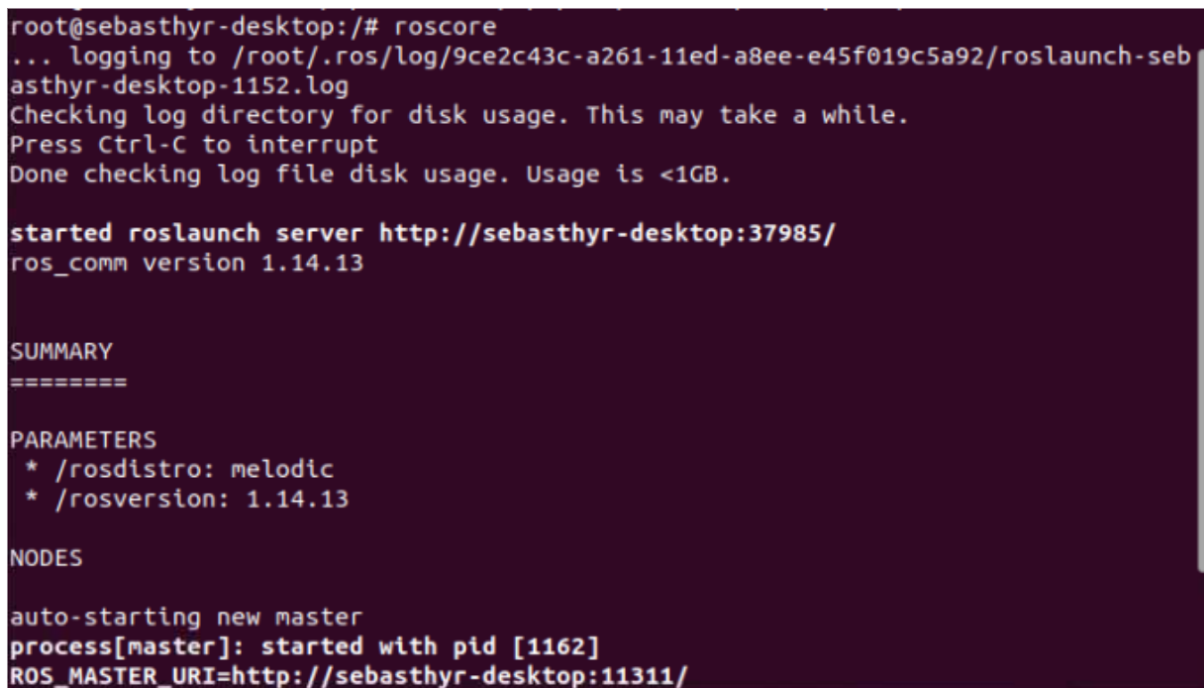
A terminal window titled 'root@sebasthyr-desktop: /' with several tabs open. The terminal shows the following commands and output:

```
sebasthyr@sebasthyr-desktop:~$ docker exec -it festive_easley bash
root@sebasthyr-desktop:/# source /opt/ros/melodic/devel/setup.bash
root@sebasthyr-desktop:/#
```

Figura F-8.: Configurar el entorno de ROS en docker.

Fuente: Creación propia

Luego de tener habilitadas varias ventanas con la imagen de docker, se inicializa el maestro de ROS.

A terminal window showing the output of the 'roscore' command. The output includes logging information, disk usage checks, and the successful start of the ROS master.

```
root@sebasthyr-desktop:/# roscore
... logging to /root/.ros/log/9ce2c43c-a261-11ed-a8ee-e45f019c5a92/roslaunch-seb
asthyr-desktop-1152.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://sebasthyr-desktop:37985/
ros_comm version 1.14.13

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES

auto-starting new master
process[master]: started with pid [1162]
ROS_MASTER_URI=http://sebasthyr-desktop:11311/
```

Figura F-9.: Ejecuta el maestro de ROS.

Fuente: Creación propia

Después de tener el maestro funcionando se corre el Launch del Gripper y se debe observar la siguiente salida

```
root@sebasthyr-desktop:/# roslaunch generic_pria generic_pria.launch
... logging to /root/.ros/log/9ce2c43c-a261-11ed-a8ee-e45f019c5a92/roslaunch-seb
asthyr-desktop-1180.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://sebasthyr-desktop:43059/

SUMMARY
=====

PARAMETERS
* /rosapi/params_glob: [*]
* /rosapi/services_glob: [*]
* /rosapi/topics_glob: [*]
* /rosbridge_websocket/address: 0.0.0.0
* /rosbridge_websocket/authenticate: False
* /rosbridge_websocket/bson_only_mode: False
* /rosbridge_websocket/delay_between_messages: 0
* /rosbridge_websocket/fragment_timeout: 600
* /rosbridge_websocket/max_message_size: None
```

Figura F-10.: Ejecución Roslaunch.

Fuente: Creación propia

Con lo anterior se activa el robot en el sistema con el siguiente comando.

```
root@sebasthyr-desktop:/# curl -X POST localhost:3000/connect
{"status":200,"text":"OK"}root@sebasthyr-desktop:/#
```

Figura F-11.: Conexión al sistema de PRIA-UN del robot.

Fuente: Creación propia

Observamos la respuesta http pero también podemos verificar en la ventana que se ejecuto el launcher que hay una respuesta del robot.

```
2023-02-01 18:53:40+0000 [-] [
ients total.
Robot ROBOT1 connected
```

Figura F-12.: Verificación conectividad del Gripper con PRIA-UN.

Fuente: Creación propia

Procedemos a enviar órdenes desde PRIA-UN donde se aclare dos valores

- Pos : Un número de 0 a 4 que indica cual motor se desea mover, la posición 4 es una rutina completa diseñada.
- Angle : Un valor numérico que se remite al motor en grados

En la base de datos se puede verificar los procesos que se crean. Igualmente se pueden acceder a los tópicos para supervisar la información que recibe el gripper de el numero de motor, el ángulo y ancho de banda PWM de posición del motor.

```
sebasthyr@sebasthyr-desktop:~$ docker exec -it festive_easley bash
root@sebasthyr-desktop:/# source /opt/ros/melodic/devel/setup.bash
root@sebasthyr-desktop:/#
root@sebasthyr-desktop:/#
root@sebasthyr-desktop:/# rosrun usb_cam usb_cam_node
```

Figura F-13.: Ejecutar tópico de ROS para transmitir imágenes.

Fuente: Creación propia

Para establecer la conexión con el servidor de visión se ejecuta el nodo de ROS usb cam node. Este nodo permite obtener las imágenes RGB de la cámara web y transmitirla a otro tópicos de ROS, en especial la transmisión por websocket de los mensajes entre nodos.

```
sebasthyr@sebasthyr-desktop:~$ docker exec -it festive_easley bash
root@sebasthyr-desktop:/# cd /home/testClientAngular/webSocketApp/
root@sebasthyr-desktop:/home/testClientAngular/webSocketApp# ng serve --host 0.0
.0.0
```

Figura F-14.: Ejecutar el servidor de visión en la RPI4.

Fuente: Creación propia

Se requiere habilitar el servidor de visión para poder acceder externamente a la información de la cámara, por tal motivo se debe ejecutar el front end de Angular para este caso.

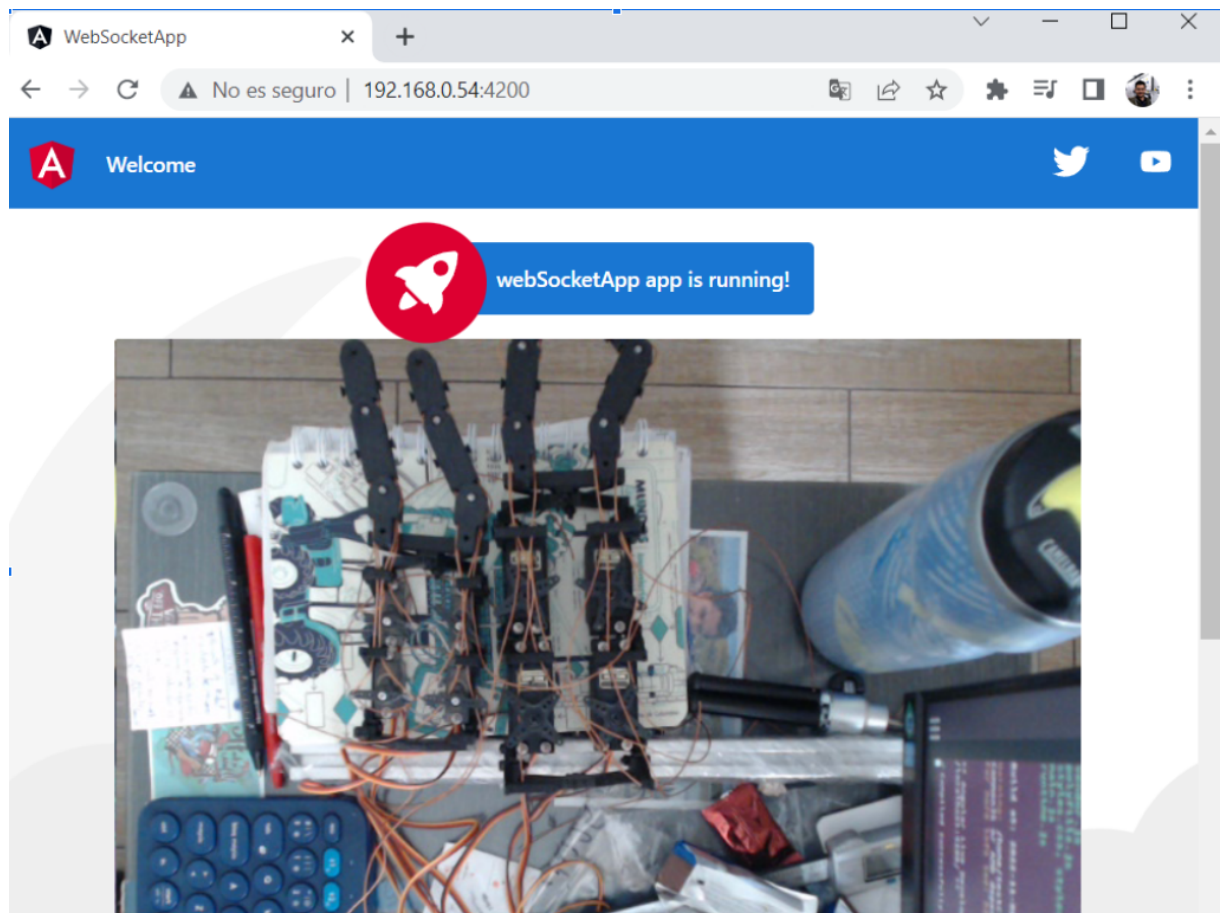


Figura F-15.: Visualización externa del servidor de visión.
Fuente: Creación propia

Finalmente se observa la transmisión de vídeo por medio de websocket para transmitir la información en tiempo real.

G. Anexo: Desarrollo método tradicional Gripper: Problema de diseño

G.1. Planteamiento del problema

G.1.1. ¿Quién?

Empresas o individuos que tengan o requieran un brazo robótico para interactuar con objetos de diferentes formas y materiales, sin tener que cambiar su Gripper para interactuar con estos objetos. Aclarando que no importa el tipo de arquitectura del brazo puede ser antropomórfico, Scara, delta, entre otros.

G.1.2. ¿Qué y para Qué?

Se plantea el desarrollo de un Gripper de doble accionamiento planar en 2 dimensiones, que responda a diferentes características de orientación final en la última falange de cada dedo, para poder ofrecer funciones de pinza y como valor agregado la capacidad de orientar el objeto manipulado. Sirve para realizar operaciones ensamble cuando la arquitectura del robot es de 3gdl y no tiene la capacidad de orientar su efector final.

G.1.3. ¿Por qué?

En la actualidad los Gripper de doble accionamiento responden únicamente a la acción de abrir y cerrar sus dedos (pinza), esta interacción permite únicamente el realizar operaciones de "Pick and Place", sin embargo, operaciones de ensamblaje no son viables debido a que la orientación del objetivo tomado no puede ser corregida por el Gripper. Por lo tanto un Gripper que garantice un mayor grado de libertad en cada uno de sus dedos permite cambiar o corregir la orientación.

G.1.4. ¿Dónde y Cuándo?

Para responder a un mercado local que no produce Grippers y todos son exportados genéricos. Se ofrecería servicios de Grippers personalizados que incluye la instalación y la puesta en funcionamiento del Gripper. Basa su funcionalidad en componentes de modularidad y

diseño paramétrico, para disminuir los tiempos de integración y garantizar ciclos de vida útil mayores.

G.1.5. ¿Costos?

El valor agregado del Gripper se centra en permitir orientar los objetos cuando ya los tiene sujetos en sus pinzas, esta capacidad requiere que donde se manufacture pueda responder a creación de partes discretas. Por tal motivo hay que analizar como responde el mercado local frente a este tipo de manufactura o optar por fabricar en mercados internacionales e importar.

G.2. Planteamiento de subproblemas

G.2.1. Defina algunas características esperadas de su producto y el entorno de operación:

El Gripper se acopla a la base de su efector final, esto indica que el área de trabajo esta descrito por el área de trabajo del brazo robótico. Los objetos para manipular son piezas manufacturadas por un torno suizo y una impresora 3d, lo que implica dimensiones de 5-35mm de diámetro y un largo de hasta 100mm. El Gripper debe tener la fuerza suficiente en las dos falanges finales de los dedos para levantar estas piezas. Con el fin de cambiar la orientación de los objetos debe poder cambiar la orientación de la superficie de contacto de las ultimas falanges de los dedos, asegurando que este haciendo contacto en todo momento. La precisión de la operación depende de la precisión del brazo robótico y del Gripper, en el cual el brazo robótico representa el sistema con mayor precisión que puede ayudar a corregir la precisión del Gripper.

G.2.2. ¿Cuál es la función positiva primaria de su producto?

Mediante la llegada de las ultimas falanges de un sistema de pinzas sujetar y orientar objetos a una posición objetivo.

G.2.3. ¿Anticipa efectos negativos para su producto?

- Según la precisión que se desea en el Gripper puede llegar a ser necesario mucha sensorica para garantizar que la ultima falange llegue con la orientación deseada.
- Para responder a las diferentes geometrías y el concepto de modularidad se debe manejar conceptos que se puedan replicar o la manufactura se vuelve compleja por la necesidad de hacer diferentes familias de productos.

- Para garantizar la fuerza de sujeción por lo general se deben usar más motores o motores de mayor tamaño, lo que puede generar problemas de peso y de tamaño que no cumplan las características del producto.

G.3. Identificación de contradicciones

Se identifican como principales contradicciones las siguientes:

- Precisión – Dificultad de detectar y medir
- Fácil de manufacturar – Complejidad del dispositivo
- Grado de Automatización – Dificultad de medición

G.4. Requerimientos del cliente

Los requerimientos se especifican en la siguiente tabla **G-1** según una entrevista al cliente, que en este caso corresponde al líder del LabFabEx-UN, donde se especifican los pesos de importancia de cada requerimiento.

Tabla G-1.: Requerimientos del cliente.

| Requerimientos | Puntaje | Tipo |
|--|---------|---------------|
| Manipular mas de 5 familias de productos | 5 | Funcionalidad |
| Orientar los objetos para ensamblaje | 6 | Funcionalidad |
| Manipular objetos con superficies lisas | 3 | Funcionalidad |
| Manipular objetos con diametro máximo de 45 mm | 3 | Funcionalidad |
| Manipular objetos con altura máxima de 100mm | 2 | Funcionalidad |
| Manipular objetos con peso máximo de 1kg | 3 | Funcionalidad |
| Identificar colisiones | 1 | Seguridad |
| Ciclo de trabajo del Gripper 10h/diarias | 3 | Manufactura |
| Tiempo de vida 5 años | 3 | Manufactura |
| Permitir actualizaciones | 6 | Económico |
| Poco mantenimiento | 3 | Manufactura |
| Reciclable | 4 | Ambiental |
| Resistir ciclos de apertura y cerrado $> 10^6$ | 4 | Manufactura |
| Integración con diferentes brazos | 4 | Funcionalidad |
| Manipular objetos con forma no simétricas | 4 | Funcionalidad |
| Operar sin intervención humano | 6 | Ergonómico |

G.5. Especificaciones de ingeniería

Los valores de las especificaciones de ingeniería se obtienen basados en el análisis de la competencia, dado que existe una estandarización en las especificaciones que deben brindar. Los atributos relevantes del estudio de mercado de la competencia proveen las magnitudes que se deben tomar. Se adicionan algunas especificaciones propias a solventar el problema de diseño.

Tabla G-2.: Especificaciones técnicas, sus valores, tipo y valor de dificultad.

| Especificaciones | Dificultad | Tipo | Valor |
|--|------------|-----------|-------------|
| Fuerza de agarre | 4 | Maximizar | 40N |
| Cantidad de sensores de fuerza | 4 | Minimizar | 4 |
| Apertura | 5 | Maximizar | 160mm |
| Cierre | 2 | Minimizar | 40mm |
| fuerza de los motores | 1 | Minimizar | 20N |
| Tiempo de respuesta | 3 | Minimizar | 20ms |
| Resistencia a la fatiga | 2 | Maximizar | $1kgf/cm^2$ |
| Cantidad de elementos rígidos | 5 | Minimizar | 4 |
| Cantidad de sensores de desplazamiento | 4 | Minimizar | 4 |
| Peso del Gripper | 3 | Minimizar | 2kg |
| Cantidad de motores | 3 | Minimizar | 4 |

G.6. Desarrollo y análisis de resultados del QFD

En el anexo B se encuentra el desarrollo de la casa de la calidad. Se resaltan las especificaciones mejor rankeadas que en orden son las siguientes:

1. Cantidad de elementos rígidos
2. Cantidad de sensores de desplazamiento
3. Cantidad de motores
4. Apertura
5. Resistencia a la fatiga

Se observa que las tres primeras especificaciones tiene en común conceptos de modularidad en términos de especificaciones de diseño, que se aplican a requerimientos como permitir orientaciones, permita actualizaciones, reciclables, entre otros. Después se encuentra una especificación en términos de la funcionalidad del Gripper y la resistencia a la fatiga se

orienta a especificaciones de diseño respecto a la operación propia de abrir y cerrar, que se evalúa como una posible falla por fatiga cuando los ciclos alcanzan el orden mayores 10^6 .

Respecto al análisis de los requerimientos con los productos de la competencia se destaca que los requerimientos de orientación respecto al objeto tomado, proporcionada por el valor agregado rankea mejor la solución. También se observa que el concepto de ofrecer el Gripper como servicio tiene repercusión en como nuestra propuesta puede garantizar características que en los comerciales son fijas.

H. Anexo: Desarrollo método tradicional Gripper: Diseño conceptual

En esta sección se desarrolla el diseño conceptual dominante que califique de la mejor manera para cumplir con los requerimientos del cliente, mientras satisface las especificaciones de diseño. En orden de cumplir el objetivo se empieza por identificar las funciones que debe cumplir el producto, después se proponen conceptos que puedan satisfacer las funciones definidas, se integran en conceptos más generales y son evaluadas para identificar el concepto global dominante.

H.1. Análisis funcional

El análisis funcional responde a la pregunta ¿qué debe hacer el producto y cómo desarrollará dichas funciones?. Al conocer los requerimientos del cliente y estar planteados como especificaciones de ingeniería se identifican las funcionalidades del producto, obteniendo de manera inicial el diagrama de caja negra. Las entradas están relacionadas con materiales, energía e información, que genera salidas del mismo tipo. Por último se plantean las subfunciones que permitirían llevar a cabo la función principal del diagrama de caja negra, y se detalla en mayor detalle en la construcción del diagrama de caja gris.

Las funciones y subfunciones desarrolladas son las que se usarán en todo el proceso de generación de conceptos y evaluación, por eso las funciones se descomponen en subfunciones que permiten dar un mayor detalle a las operaciones que debe realizar el producto. Adicionalmente se observa cómo interactúan entre ellas y establecer las prioridades o funciones que pueden ser más importantes de satisfacer.

H.1.1. Diagrama de caja gris

Se realiza el análisis de caja negra en donde se identifican los conceptos de energía, materiales e información. Respecto a la energía el Gripper se conecta al último eslabón del brazo robótico y puede proveer energía eléctrica y neumática fija y continua, lo que significa que no necesita procesos adicionales de almacenamiento de energía. La salida en términos de energía se da específicamente por la pérdida de energía al ser convertida en energía mecánica, por lo que puede generar ruido y calor principalmente.

El análisis de materiales que entran y salen responden a la operación propia del Gripper de manipular objetos, y en este caso específico los objetos que describe el cliente se expresan en piezas maquinadas por equipos industriales como el torno suizo, el centro de mecanizado e impresoras 3d. Por consiguiente la salida de material responde a los mismo objetos sin deformaciones, pero cumpla con requerimientos de pose definidos al iniciar el proceso.

Respecto a la información, el sistema se encuentra comunicado con un software de ejecución de manufactura MES, que disminuye la interacción humano maquina al permitir automatizar operaciones comunicando la maquinaria de la celda de manufactura. Adicionalmente se identifican información adicional que debe ingresar al Gripper al tratarse de un sistema mecatrónico para el control y realimentación del mismo, como la posición y orientación objetivo a la cual debe llegar el objeto al final. La salida de información se representa en información que sea necesaria para automatizar la operación por medio del software MES e información que necesite el operario para identificar el estado del Gripper.

En la figura H-1 se observa el diagrama de caja gris, se observa como se comunican las funciones de energía con los materiales y la información para satisfacer el objetivo principal.

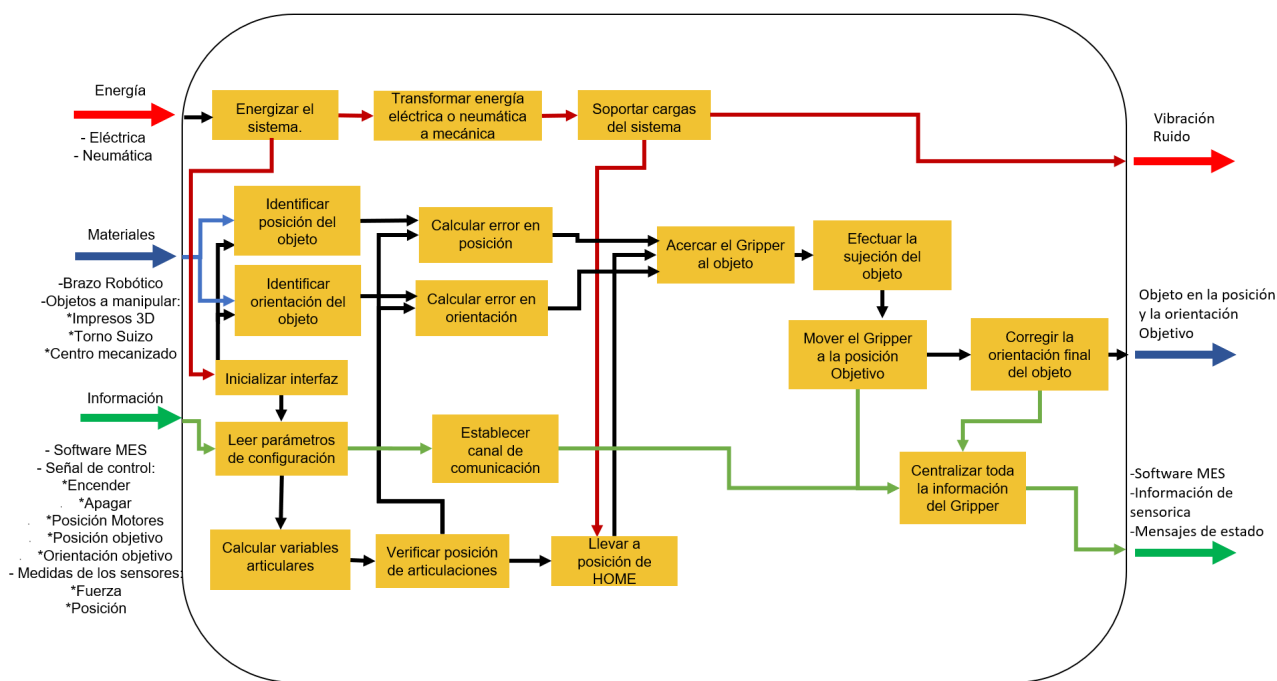


Figura H-1.: Diagrama de caja gris del Gripper.

H.1.2. Descomposición funcional

La descomposición funcional se realiza respecto al diagrama de caja gris, en el cual se identifican 5 funciones principales que pueden agrupar las subfunciones descritas. En la figura

H-2 se observan las funciones principales en color azul y las subfunciones en color amarillo.

En orden de izquierda a derecha se pueden clasificar como:

1. Subfunciones de suministro de energía.
2. Subfunciones de inicialización de sensorica y sistemas de control.
3. Subfunciones de sensorica y electrónica.
4. subfunciones de control.
5. Subfunciones de movimientos orientados a secuencias mecánicas.

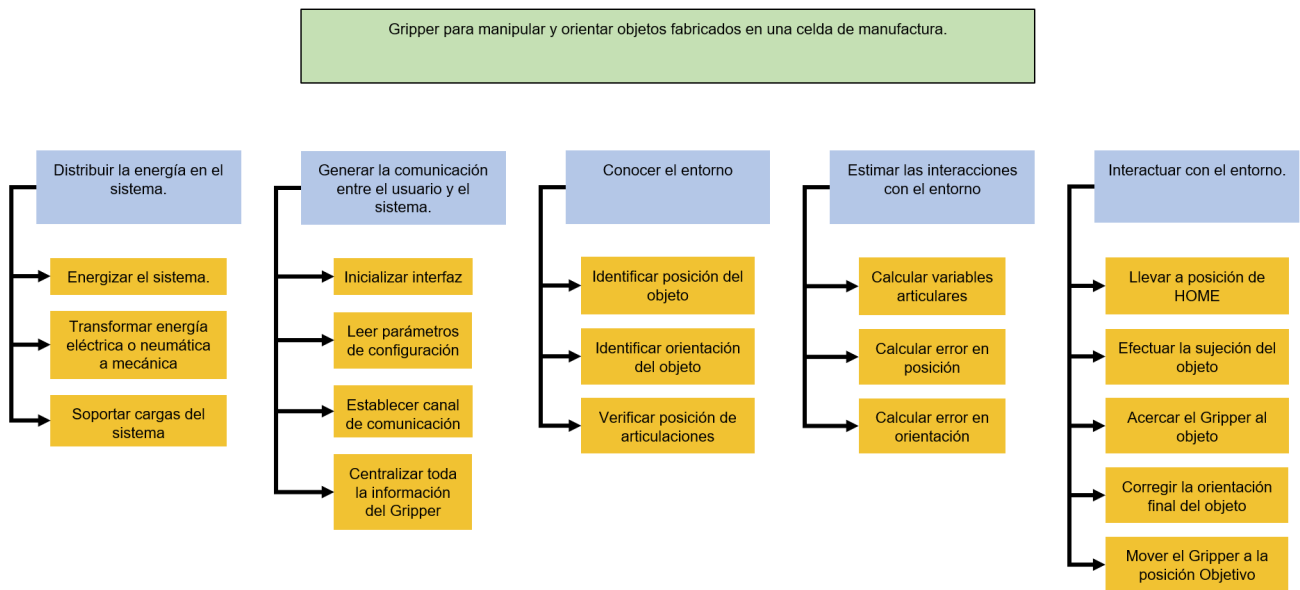


Figura H-2.: Descomposición funcional del Gripper.

H.2. Técnica de inventiva Triz

Durante el primer capítulo se planteó el problema y los subproblemas de diseño, en la última etapa se plantean parámetros que se contradicen y que se deben analizar para garantizar el mejor funcionamiento. Con lo anterior se aplica el método consolidado y revisado de TRIZ para identificar principios inventivos de solución. Se realiza para las tres contradicciones principales que afectan al problema de diseño.

H.2.1. Sub-problema 1

El planteamiento de un Gripper que permita modificar la orientación del objetivo, requiere conocer la posición de cada eslabón en la estructura de la pinza y así calcular la cadena cinemática, por lo tanto, entre mayor precisión tenga el Gripper en términos de sensores directos e indirectos, será menos difícil de detectar y medir el objeto con el que se tenga contacto, donde entra en efecto la primera contradicción.

A continuación se describen los parámetros técnicos en términos del método Triz:

- Parámetro técnico (#1) que desea mejorar: Precisión de la medición (#28)
- Parámetro técnico (#2) que empeora: Dificultad de detectar y medir (#37)

Con los parámetros técnicos definidos y la aplicación de la técnica de inventiva TRIZ se identifican los siguientes principios inventivos que pueden emplearse:

1. Copia (#26)
2. Mediador, intermediario (#24)
3. Cambiar el color o las propiedades ópticas (#32)
4. Reemplazo del Sistema mecánico (#28)

Se identifican que el concepto de gemelo digital ("Digital Twin") y el uso de cámaras son las principales soluciones para llevar a cabo la medición de los eslabones del Gripper. El cual plantea mediciones indirectas y como hacer el seguimiento y procesamiento de múltiples sensores. Aunque las mediciones indirectas se pueden realizar con cámaras y mejorar la detección en visión de máquina con el cambio de color de las piezas, también se puede medir indirectamente al medir como se desplace los sensores. El reemplazo del sistema mecánico se puede aplicar al último eslabón con sensores de presión o hápticos para identificar el contacto con el objeto y tener una medición directa.

H.2.2. Sub-problema 2

Para poder responder a la manufactura discreta es esencial que se manufacturen la menor cantidad de familias de productos, debido a que ellas cambian parámetros de manufactura que no requieren un "seteo" de máquina y por tal motivo se puede considerar una manufactura secuencial para estas piezas. Sin embargo, la naturaleza de poder ofertar mayores grados de libertad en un Gripper de doble accionamiento implica que la complejidad del Gripper aumente. En contra parte si se tienen que fabricar pocas piezas pero todas distintas geometrías hace la manufactura más difícil mientras el sistema en sí representa una complejidad menor por la cantidad de piezas.

A continuación se describen los parámetros técnicos en términos del método Triz:

- Parámetro técnico (#1) que desea mejorar: Fácil de manufacturar (#32)
- Parámetro técnico (#2) que empeora: Complejidad del dispositivo (#36)

Con los parámetros técnicos definidos y la aplicación de la técnica de inventiva TRIZ se identifican los siguientes principios inventivos que pueden emplearse:

1. Objetos baratos de corta vida (#27)
2. Copia (#26)
3. Segmentación (#1)

Los principios inventivos guían al termino de Modularidad, donde se plantea una cadena cinemática de eslabones bajo un enfoque paramétrico que permita manufacturar estas como una familia de productos. Se plantea bajo un esquema generativo donde solo cambian parámetros de tamaño a la forma del diseño y que no requieren procesos de ajuste de máquina para manufacturar como si fuera en serie.

H.2.3. Sub-problema 3

Se busca la menor interacción del operario por lo que el sistema representa un alto grado de automatización y también representa una menor dificultad en la medición. Dado que el dispositivo tiene alto grado de automatización puede realizar operaciones de sensorica mas fácilmente y con mayor frecuencia que si contara con un grado menor de automatización donde los recursos de computo son limitados a unas pocas tareas. Al igual el grado de automatización limita los tiempos de espera de instrucciones de un operario humano y puede realizar múltiples tareas al tiempo, ejemplo: el calculo de sus articulaciones y demás se hace de forma inmediata.

A continuación se describen los parámetros técnicos en términos del método Triz:

- Parámetro técnico (#1) que desea mejorar: Grado de automatización (#38)
- Parámetro técnico (#2) que empeora: Dificultad medición (#37)

Con los parámetros técnicos definidos y la aplicación de la técnica de inventiva TRIZ se identifican los siguientes principios inventivos que pueden emplearse:

1. Rechazo y regeneración, descarte y recuperación (#34)
2. Objetos baratos y desechables (#27)
3. Autoservicio, auto organización (#25)

Un alto grado de automatización permite medir y controlar sistemas que incluyan sistemas como resorte, que ayudan a hacer uso de energía cinética para volver a la posición de estabilidad o de home, o ayudar en las regiones donde la fuerza proporcionada por el motor es menor. También se menciona que estos elementos, como los resortes, se puedan rechazar cuando ya no sirvan o que las piezas se puedan volver a fundir para reemplazarlas o que se pueda hacer cambio de motores en términos de modularidad.

H.3. Generación de conceptos

Se establecen conceptos como primitivas para satisfacer las subfunciones que representan la descomposición mayor de la operación del Gripper. En este orden de ideas los conceptos son generados por dos técnicas principalmente uno basado en la propuesta de Diseño Generativo Realimentado y la otra por una lluvia de ideas de conceptos como primitivas.

H.3.1. Técnicas de creatividad

Diseño Generativo Realimentado

Libros guías de metodologías diseño de producto contemporáneas como presenta Ullman, 2010 [12], en el diseño mecánico o el “Diseño Axiomático” presentado por Farid et al., 2016 [13], hacen énfasis en que la labor de creación de conceptos es dependiente de la experiencia del diseñador y de sus conocimientos, donde no se incluyen la información de productos creados por otros diseñadores en términos de proceso de manufactura y medición del ciclo de vida del producto.

Lo anterior plantea que los paradigmas actuales responden a características de realimentación que no aportan a los procesos de creación de producto, sino a conceptos de mejoramiento como lo expresa Norton, 2011 [14], con el “Diseño Detallado”. Por lo cual se observa una oportunidad de generar una propuesta en la etapa de diseño, aplicable a los diferentes modelos contemporáneos, que realimente la etapa de diseño de producto con enfoque a la creación de productos.

La finalidad es crear un sistema que pueda heredar características propias de un diseño para ser usado y ajustado a la etapa de diseño de un nuevo producto. La propuesta incluye como principal herramienta la innovación y la retroalimentación patrones funcionales de los productos fabricados previamente, en este caso particular eslabones rígidos y tensores con sistemas de poleas para la creación de un “griper planar 2D”. El objetivo de este “gripper” es la manipulación de piezas creadas en torno suizo, centro de mecanizado e impresoras 3D del laboratorio LabFabEX- UN.

El aporte del modelo de soporte al diseño, como el Diseño Generativo mostrados por trabajos de Li et al., 2018 [32], Khan et al., 2019 [23], Gupinar et al., 2019 [90], o la Optimización

topológica, muestran la viabilidad de construir a partir de componentes específicos identificando sus características principales, sin embargo, en la creación de un nuevo producto algunas características no están definidas, o no se pueden definir por la dependencia con otro componente.

Se identifica que la realimentación en la etapa de diseño a una pieza fabricada solo brinda información para el mejoramiento de esta. No es posible utilizar la información de sus componentes parciales para ser heredada de manera univoca en otro diseño en un proceso de creación. En el desarrollo del trabajo investigativo se lleva un desarrollado de un motor de soporte al diseño. Bajo el desarrollo de funciones multiobjetivo para satisfacer los requerimientos de productos, se realiza un proceso de realimentación de conceptos (eslabones, tensores y poleas) en un espacio planar, para obtener un conjunto de conceptos que satisfacen de la mejor manera la definición de dichos requerimientos.

Lluvia de ideas como primitivas.

Una primitiva se puede describir como un nivel de abstracción en el cual al combinarlas entre si, generen un sistema mas complejo, por lo anterior y buscando aplicar el conceptos de familias de productos los conceptos se describen en unidades o ideas de bajo nivel que al combinarse generen sistemas mas completos que satisfagan la función.

H.3.2. Mapeo función-concepto

Al aplicar las técnicas de creatividad y la técnica de inventiva Triz para cada una de las subfunciones del sistema se observa comportamientos donde un solo concepto satisface varias funciones, que la combinación de varios conceptos satisfagan uno o varias funciones, que varios conceptos satisfagan varias funciones, entre otros. Los diagramas de cada concepto se encuentran contenidos en el anexo ??, donde se pueden apreciar en mayor detalle.

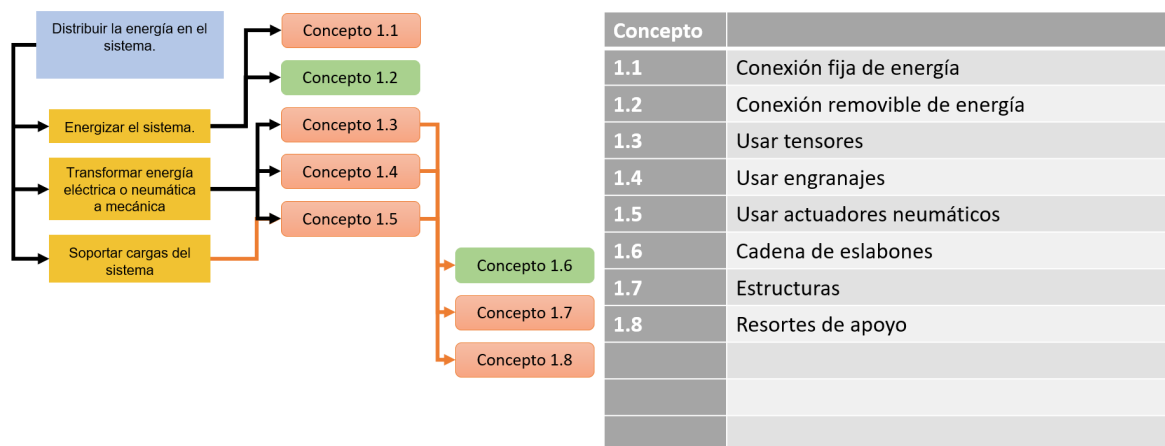


Figura H-3.: Mapeo función 1-concepto.

La figura **H-3** muestra los conceptos que permiten hacer la transformación de la energía eléctrica o neumática en mecánica para realizar las operaciones del Gripper. En la figura **H-4** representa concepto de las interfaces que pueden darle la autonomía de inicializar los distintos procesos de electrónica y control del Gripper.

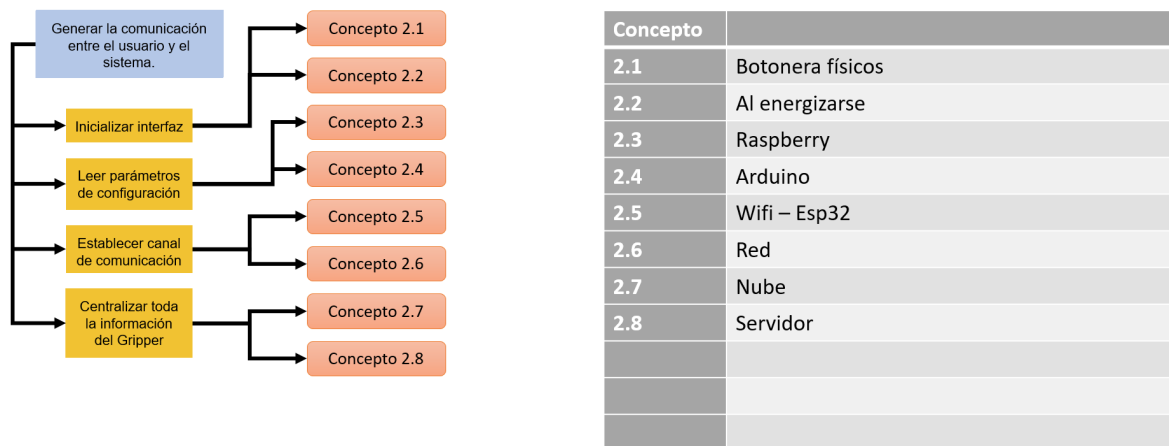


Figura H-4.: Mapeo función 2-concepto.

En el mapeo de la figura **H-5** se identifica principalmente la sensorica que puede servir en el mecanismo para realizar mediciones directas o indirectas de la posición y fuerza de las pinzas del Gripper. En la ultima figura **H-6** se observa que hereda conceptos de la primera función dado que la manipulación de los objetos se da por los movimiento mecánicos que realiza la pinza, en este orden de ideas se muestra como la primera función esta correlacionada con la función 5.

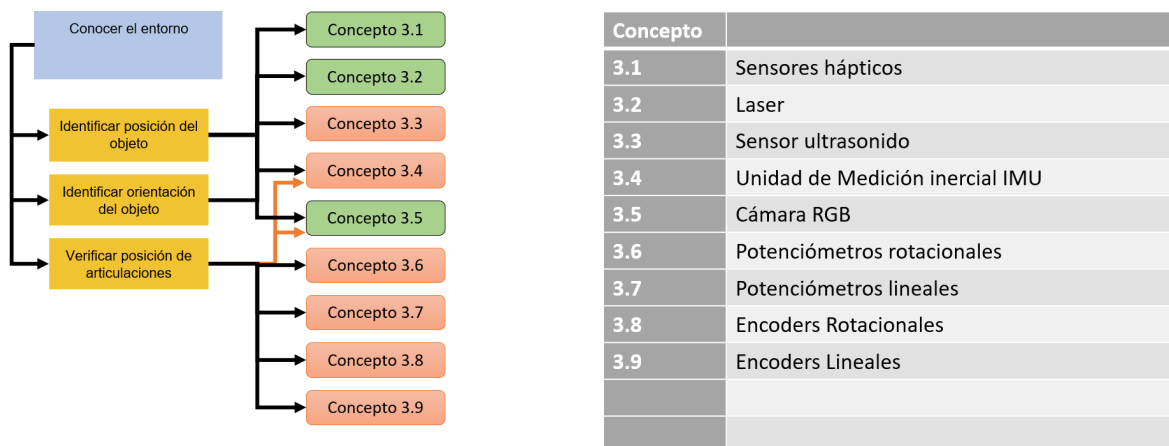


Figura H-5.: Mapeo función 3-concepto.

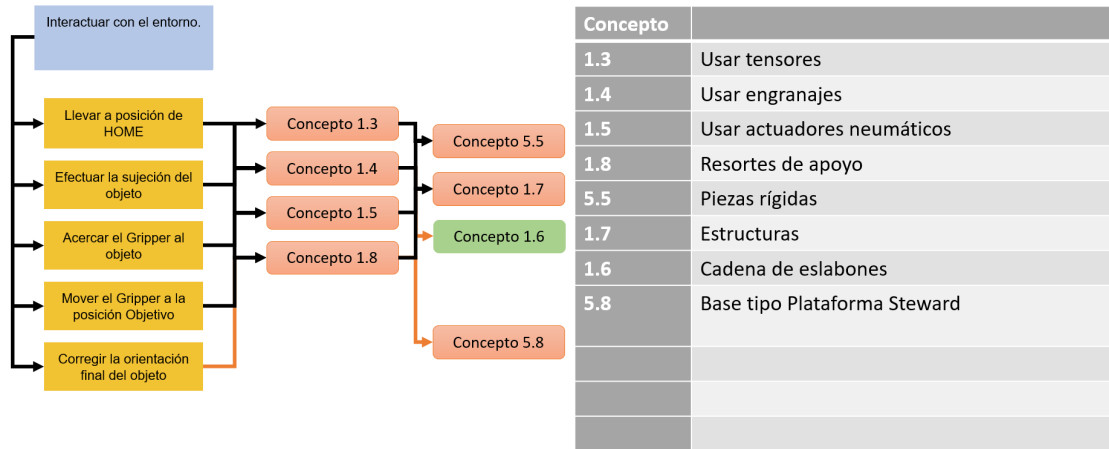


Figura H-6.: Mapeo función 5-concepto.

H.4. Integración de conceptos

La combinación de conceptos para generar un sistema solido que cumpla todas las funciones objetivos se realiza mediante arboles de clasificación y una tabla de combinación de conceptos.

H.4.1. Árbol de clasificación de conceptos

Los arboles de clasificación que estan contenidos en el anexo ?? agrupan los conceptos por áreas de conocimiento, estableciendo que los conceptos se desarrollen en las áreas del conocimiento en que mejor se desempeñen y tenga conocimiento el equipo de trabajo, aunque este no debe ser un factor para excluir debido a que un problema puede ser fácilmente resuelto por un área y muy complicado en otra, lo que genera son etapas de consulta o asesoramiento de personal capacitado para la evaluación de la propuesta.

| Línea de funciones: | | | | |
|---|--|---|--|--|
| Función #1 Identificar orientación del objeto. | Función #2 Calcular error en orientación. | Función #3 Acercar el Gripper al objeto. | Función #4 Efectuar la sujeción del objeto. | Función #5 Corregir la orientación final del objeto |
| Concepto 3.5 | Concepto 4.1 | Concepto 1.3 Concepto 1.6 | Concepto 1.3 Concepto 1.6 | Concepto 1.3 Concepto 1.6 |
| Concepto 3.1 | Concepto 4.2 | Concepto 1.5 Concepto 1.6 | Concepto 1.5 Concepto 1.6 | Concepto 1.5 Concepto 1.6 |
| Concepto 3.2 | Concepto 4.3 | Concepto 1.4 Concepto 5.5 | Concepto 1.4 Concepto 5.5 | Concepto 5.8 |

Figura H-7.: Tabla de combinación de conceptos en la línea funcional principal.

H.4.2. Tabla de combinación de conceptos

Por este método se puede obtener una gran cantidad de conceptos globales que pueden ser evaluados. En la figura H-7 se observa la línea principal de funciones, donde se agrupan las funciones de mayor nivel técnico en el desarrollo del Gripper. Se observan como combinación de conceptos pueden satisfacer mas de una función y poder identificar que los verdes estan ligados a las técnicas de inventiva Triz por lo cual genera ventajas en la solución de contradicciones que los conceptos únicamente generado por lluvia de ideas.

H.5. Evaluación y selección de Conceptos

En la figura H-8 se observan los tres conceptos globales fuertes que se evalúan para definir el que se llevara a la etapa de ingeniería de detalle. La evaluación se realiza por medio de 4 etapas sucesivas de eliminación para mantener los conceptos que cumplen las funciones planteadas, el desarrollo de las tablas se encuentran en detalle en el anexo ?? y corresponden a las siguientes etapas: 1) Juicios de factibilidad. 2) Revisión de la tecnología. 3) Evaluación de Funciones. 4) Matriz de decisión (PUGH).

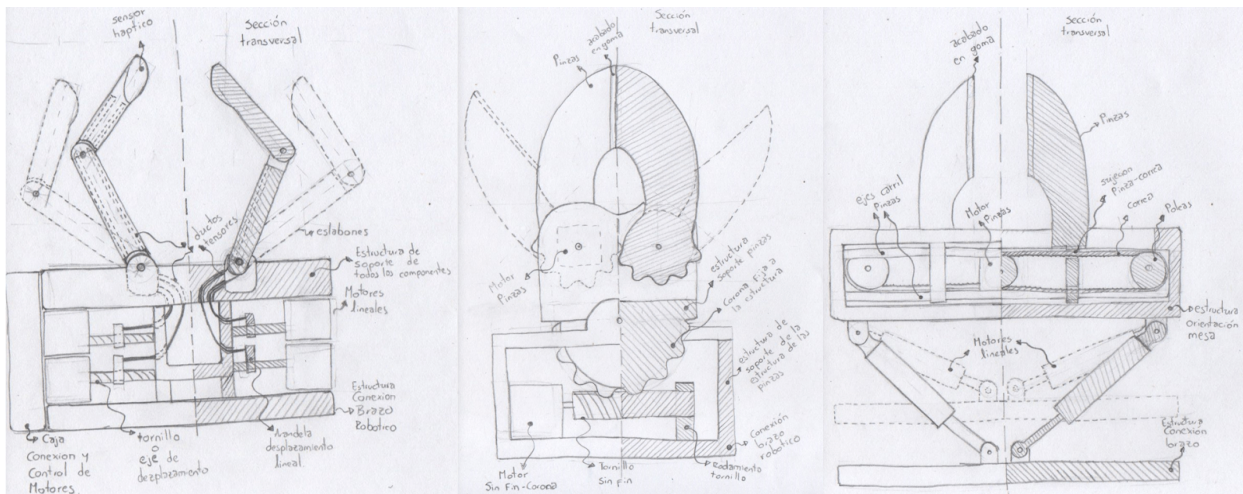


Figura H-8.: Conceptos Globales resultantes. (a.) Eslabones rígidos y tensores. (b.) Pinzas por engranajes. (c.) Pinzas por correas dentadas.

H.6. Transmisión efectiva del Concepto Global Dominante (CGD)

El concepto global dominante se observa en la figura H-9, se observan dos cadenas cinemáticas de dos eslabones cada una. Cada eslabón en la cadena cinemática tiene una actuación por tensores, por lo cual, cada eslabón es independiente del otro. Los tensores son tensionados

por motores lineales, los cuales cuentan con su propio sensor lineal para realizar mediciones indirectas de la ubicación de los eslabones. En el ultimo eslabón se plantea complementarlo con un sensor haptico para tomar medidas directas de contacto y fuerza del objeto.

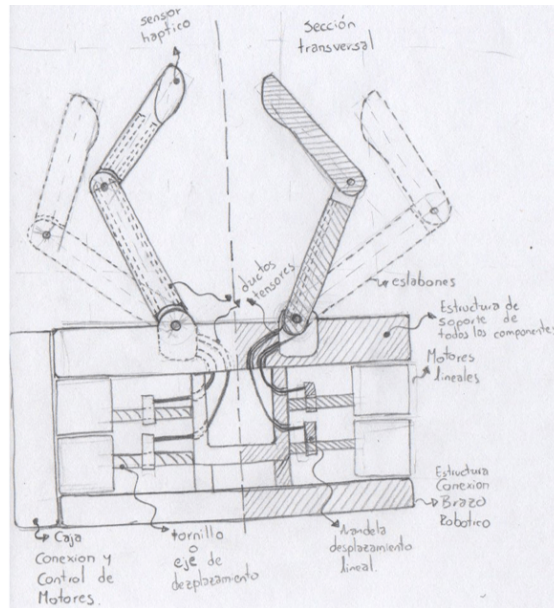


Figura H-9.: Conceptos Globales dominante: Sistema de eslabones rígidos y actuación por sensores.

I. Anexo: Desarrollo primera iteración

Gripper: Diseño detallado

En este capítulo se condensa todas las consideraciones obtenidas en el proceso iterativo del proceso de diseño detallado, en el cual las secciones propuestas se establecen como un proceso secuencial para el desarrollo final, sin embargo, se aclara que es un proceso paralelo donde el desarrollo de un concepto implica consecuencias en el desarrollo de otros conceptos de las secciones previas.

La distribución de este capítulo comienza por la generación conceptual del diseño detallado, luego el desarrollo en diseño industrial del producto, el planteamiento y desarrollo de los modelos del Gripper, proseguido de una evaluación del diseño por factores y su respectiva transmisión efectiva del Gripper.

I.1. Generación del diseño detallado

Para el desarrollo del diseño detallado seguimos el siguiente esquema planteado donde se analizan el producto desde etapas de conceptualización, identificación de materiales y técnicas disponibles, la interacción de los diferentes subsistemas y su integración, entre otros. Este análisis de diferentes aspectos proporciona una guía para definir características que debe cumplir el producto y en vez de ser un proceso secuencial, es un proceso multi-direccional en el cual se desarrolla la sinergia del sistema.

Se resaltan factores relevantes que se usaron como criterio de selección o descarte de conceptos, y resaltando que su selección depende de un proceso global iterativo.

I.1.1. Diseño a nivel del sistema

Se realiza la interacción de subsistemas basados en sus interacciones fundamentales, las cuales pueden indicar relaciones físicas o de energía y comunicaciones. En la figura **I-1** se observa el diagrama de estas interacciones. Se destaca como la estructura tiene una interacción con la mayoría de subsistemas, cumpliendo su funcionalidad de dar soporte a todos los componentes del Gripper. También se destaca como el subsistema de la cadena cinemática es el subsistema que interactúa de distinta manera con los demás subsistemas, lo que indica que como el

principal propósito del Gripper depende de las cadenas cinemáticas los otros subsistemas proveen las características que este necesita para su buen comportamiento.

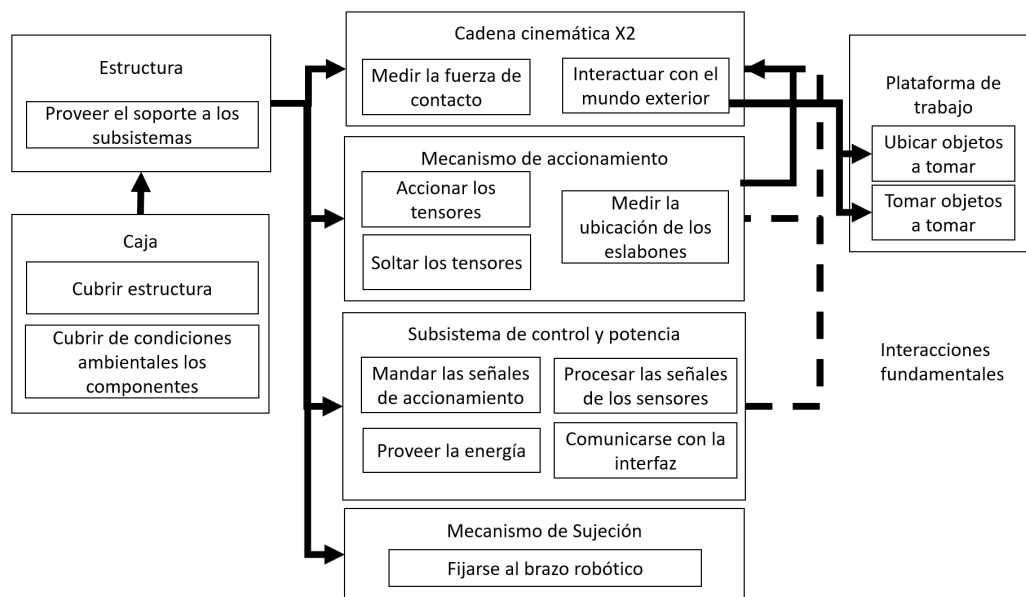


Figura I-1.: Interacciones fundamentales.

Sin embargo las interacciones fundamentales implican unas interacciones incidentales propias de la funcionalidad de los subsistemas y se deben identificar para evitar que estas interacciones perjudiquen el buen funcionamiento del Gripper. En la figura **I-2** se observan estas interacciones en detalle. Se resalta que las interacciones identificadas son variadas y en niveles altas pueden ser perjudiciales para el Gripper como el operario.

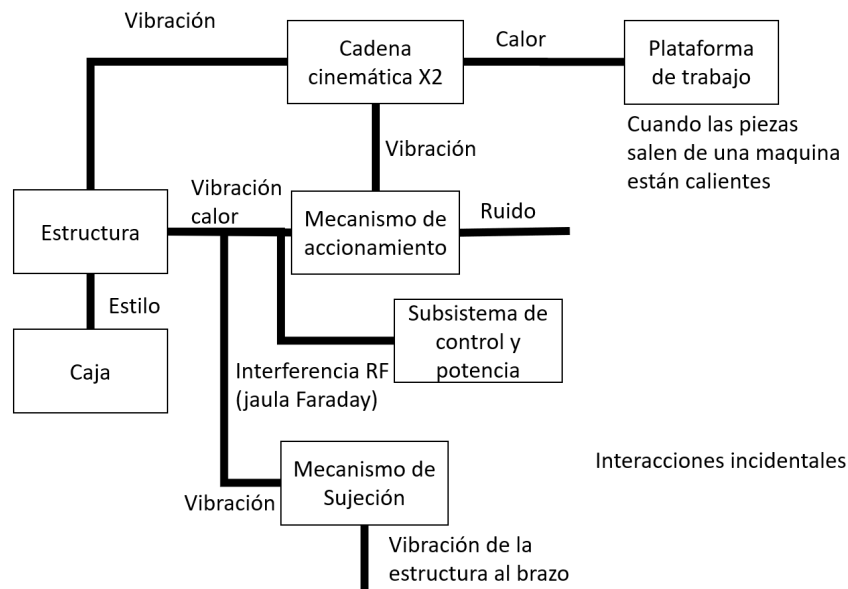


Figura I-2.: Interacciones Incidentales.

I.1.2. Definición de la arquitectura

Con la definición de los subsistemas y sus interacciones se plantea como sería su arquitectura en términos espaciales como se muestra en la figura **I-3**. Se observa como espacialmente la caja provee tanto diseño como protección a los subsistemas de apoyo a las cadenas cinemáticas. Al igual se observa como la estructura provee soporte a los subsistemas y para el mecanismo de sujeción junto a las cadenas cinemáticas el soporte solo interactúa con las interfaces de estos sistemas.

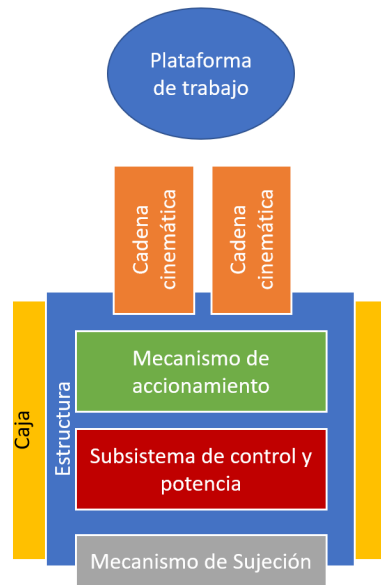


Figura I-3.: Interacción de los subsistemas.

I.1.3. Componentes disponibles y normalizados

Los elementos normalizados son componentes discretos que bajo el enfoque de diseño paramétrico pueden ser modificados instantáneamente. El modelado responde a los cambios para entrar a una etapa de evaluación y permitir el proceso iterativo de diseño. Por lo anterior la selección detallada de los componentes se realiza en una etapa posterior de desarrollo de diseño detallado, pero se mencionan cuáles son los componentes que pueden interactuar en el diseño.

- Motores Lineales: Existe una muy variada gama de motores lineales, sin embargo, la selección depende de las cargas que se calculen en el mecanismo y además también se puede considerar usar un motor rotacional y crear el sistema mecánico de transmisión. Lo que si se define es un recorrido de 20mm que es la distancia necesaria para que la cadena cinemática genere el movimiento.
- Pines (pasadores): Para las juntas rotacionales se establece el uso de pines estandarizados, aunque pueden fabricarse en el torno suizo. Estos pines se deben calcular para soportar los esfuerzos cortantes y por ser material metálico con un acabado superficial liso puede cumplir la función de rodamiento.
- Rodamientos: Aunque se plantea que las juntas no incluyan rodamientos por los precios y el tamaño, al evaluar el sistema rotacional respecto a un pin(eje), se hace la respectiva consulta para identificar y tener los valores dimensionales para implementarlos en el

diseño. Igualmente, la selección de estos componentes depende del cálculo de esfuerzos en un modelo más detallado.

- Separadores tornillo: Por el concepto de modularidad y la implicación de que la estructura se construya con la menor necesidad de manufactura por no tener requerimientos de alta prioridad. Se establecen componentes mecánicos estandarizados como separadores para realizar el diseño de la estructura.
- Tarjetas de desarrollo: Permiten realizar las operaciones de cómputo y de comunicación necesarias para alcanzar un alto nivel de automatización. Entre los principales exponentes se encuentra la Raspberry pi, Arduino, nodeMCU, Launchpad MSP430, entre otros.
- Electrónica: Hace referencia a circuitos embebidos para cumplir e interconectar los diferentes sistemas electrónicos en el sistema principal. Se destacan la funcionalidad para regular el voltaje, para el acoplamiento de las señales de los sensores, para regular y amplificar, entre otros.

I.1.4. Selección de materiales y técnicas de producción

Una de las características de este proyecto es que se plantea para desarrollar en el LabFabEX-UN que puede simular condiciones de automatización para la manufactura discreta, por tal motivo se cuenta con maquinaria a disposición como es el Torno suizo, un centro de mecanizado de 5 ejes, una impresora FDM y una impresora 3D industrial.

Aunque la definición detallada de cual método de fabricación para la realización del producto se puede llevar a cabo en cualquiera de las tecnologías mencionadas, se identifica la oportunidad para la fabricación de un prototipo por medio de la manufactura por adición de material. El resultado final puede ser piezas compuesta por secciones manufacturas por remoción de material y por secciones en impresión 3D, sin embargo, no quita la implicación de ser piezas uniformes y desarrollados únicamente por un método de manufactura.

Se dispone de los siguientes materiales y técnicas para llegar a plantear un diseño detallado:

- ABS o PLA para impresora 3d.
- Varilla de Nylon para torno suizo.
- Varillas normalizadas de aluminio para torno suizo.
- Bloques de aluminio para centro de mecanizado.
- Bloque de Nylon para centro de mecanizado.
- Laminas de aluminio para corte láser o chorro de agua.

I.1.5. Restricciones espaciales

Respecto al entorno, los objetos a manipular se encuentran en una zona de trabajo donde no tendrán contacto con otros elementos ni con otros humanos, por lo cual será un ambiente controlado. En este ambiente controlado también se consideran condiciones de luz y la ubicación de al menos una cámara.

Respecto al espacio, el tamaño de los eslabones de las cadenas cinemáticas del Gripper se calcula por medio del motor de diseño generativo y se calculan en detalle en el modelado de la siguiente sección. Los componentes estandarizados como los motores y la tarjeta de desarrollo junto a la electrónica adicional ocupan un espacio definido discreto y son seleccionados según los cálculos que deben soportar en el modelado.

Respecto a la admisión y evacuación del material, se realiza en entornos controlados que corresponden a espacios definidos de Pick and Place, se define estaciones en la salida de las maquinarias compuestas por plataformas con condiciones apropiadas para reconocimiento por cámaras y que el único objeto sobre las plataformas corresponde al objeto a manipular.

I.2. Planteamiento y desarrollo de los modelos

I.2.1. Modelos Gráficos

El análisis del movimiento de las cadenas cinemáticas por medio de tensores como se observa en la figura **I-4**. Se resalta que en la figura **I-4(a)** correspondiente a la parte superior el tensor es libre y no tiene condiciones en el comienzo del eslabón. Esta condición muestra como cambia la dirección de la fuerza ejercida y como afecta en el momento cuando se encuentra en el punto mas bajo, debido a que la dirección de la fuerza esta muy cercano al eje de rotación y su momento tiende a 0. El análisis implica un movimiento nulo o muy pequeño del eslabón o la necesidad de aplicar una fuerza mayor.

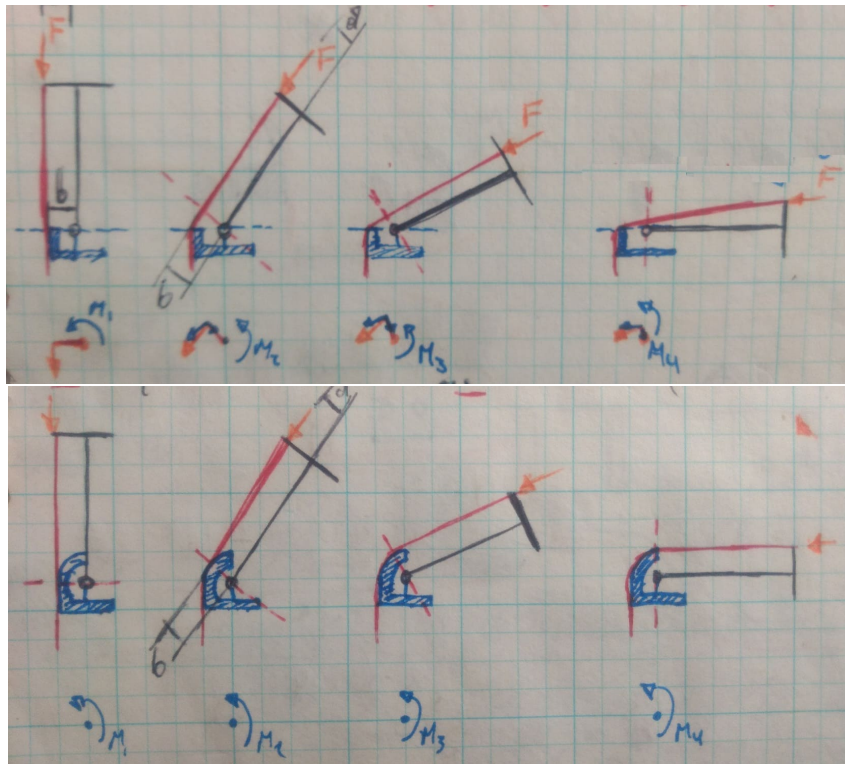


Figura I-4.: Movimiento de eslabones por medio de tensores. a) Simulando el movimiento sin condiciones espaciales en el eje de rotación. b) Simulando condiciones de polea en el eje de rotación.

En la figura I-4(b) de la parte inferior se establecen condiciones de polea para que el tensor se pueda deslizar y cumpla la funcionalidad de transmitir la fuerza conservando distancias al eje de rotación. Al igual el radio que se crea en la polea permite que entre secciones del tensor el cambio angular sea mas pequeño y no genere fuerzas de fricción que impidan el movimiento del tensor.

El análisis referente a cuales son las condiciones de polea y por que es necesario implementarlas fue el resultado de la observación de prototipos previos y se observa en la figura I-5. En esta figura se detalla el comportamiento de una sección del tensor frente a una rotación en el cual la sección de rotación presenta un corte de 90° y en la otra la superficie simula una curvatura circular como las poleas.

El análisis se centra en la deformación que se genera en el tensor respecto a la superficie de contacto, en el cual su deflexión genera una zona de compresión definida por la región azul mientras en el extremo que no tiene contacto se genera una fuerza de tensión. Esta deformación genera e la imagen de la izquierda que el diámetro del tensor disminuya y crea una mayor superficie de contacto al igual que otros fenómenos que hacen que el contacto del tensor y al superficie no sea lisa y genera estancamientos en su desplazamiento. Mientras al

simular una polea parcial los efectos de la deflexión no genera mayores deformaciones en el tensor y presentara únicamente una fuerza de rozamiento entre los dos objetos.

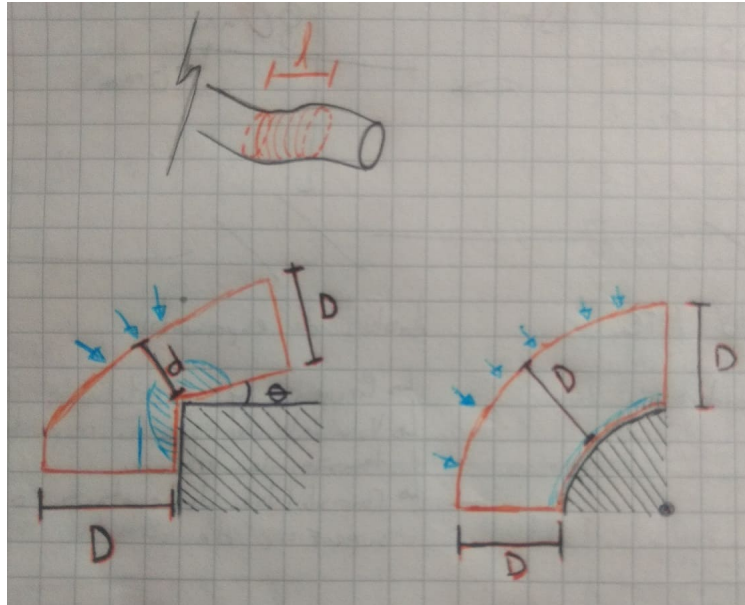


Figura I-5.: Condiciones de polea en el eje de rotación.

Con lo anterior en la figura **I-6** se amplia el análisis considerando don tensores que permiten el movimiento en ambas direcciones, por lo cual los conductos por donde pasan los tensores deben generar estas condiciones de polea en sus extremos para permitir el buen funcionamiento del eslabón.

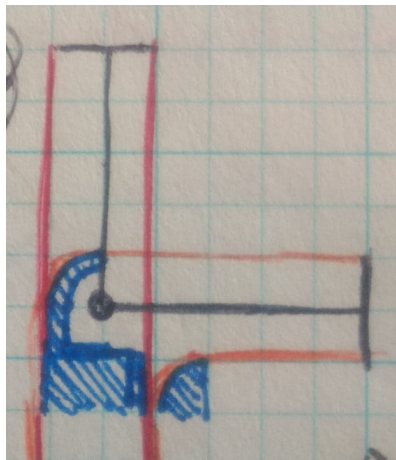


Figura I-6.: Condiciones de polea en la aplicación bidireccional de fuerzas tensoras.

Los análisis previos indican el movimiento de los eslabones sin ningún peso extra, por tal motivo en la figura ?? se observa la condición crítica para el Gripper en el cual todo la fuerza

del peso del objeto a manipular se ejerce mayormente sobre un eslabón. Al identificar esta condición todos los análisis numéricos y de simulación se realizan bajo esta condición en el cual una cadena cinemática debe soportar toda la carga.

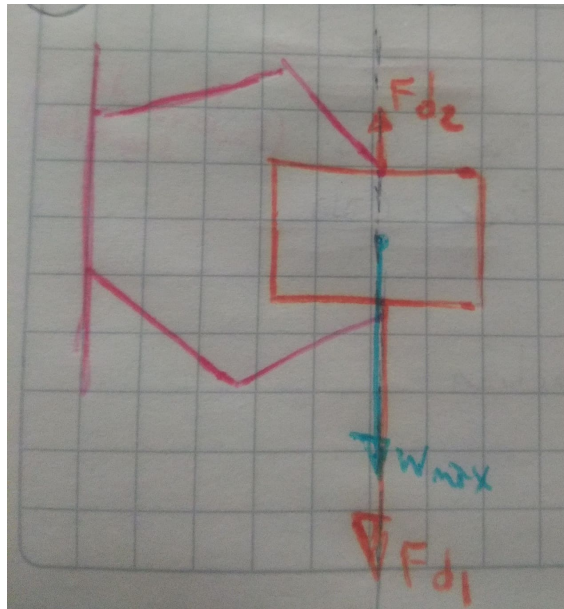


Figura I-7.: Condición crítica al manipular objetos.

En la siguiente figura **I-8(a)** se observa como la fuerza del objeto es aplicado al ultimo eslabón en el caso critico estudiado, y para obtener dicha fuerza normal se presenta la geometría que debe tener la huella del ultimo eslabón en la figura **I-8(b)**, el cual permite cambiar la orientación del ultimo eslabón sin perder el contacto óptimo con el objeto.

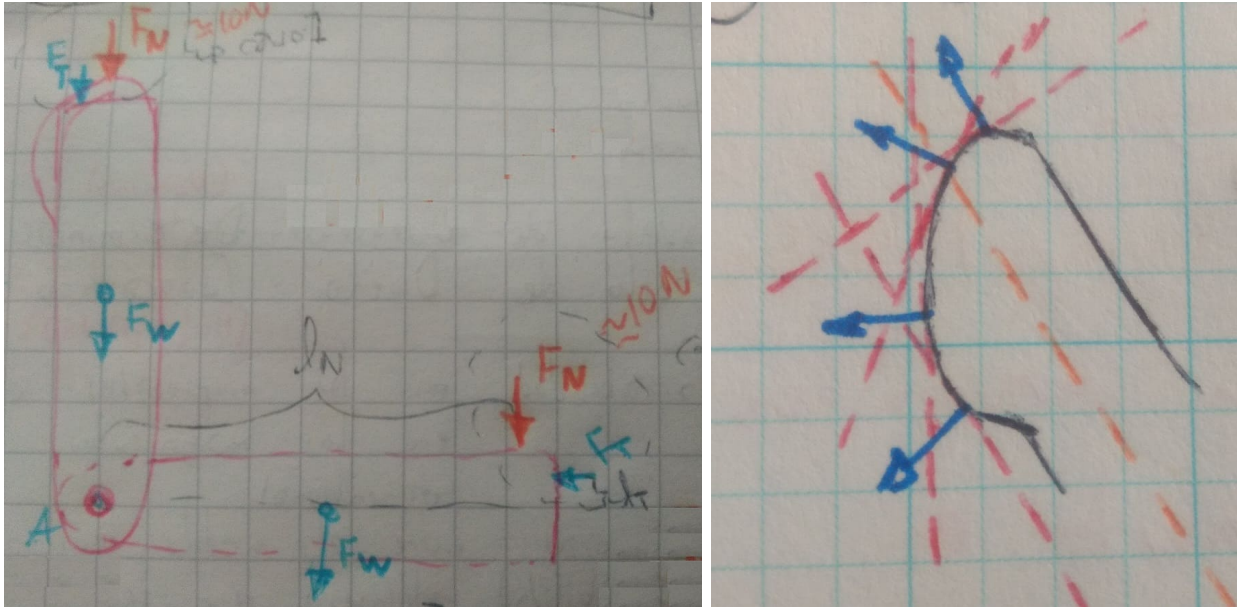


Figura I-8.: Fuerza de contacto con el objeto a manipular. a) La fuerza normal que ejerce el objeto sobre el ultimo eslabón. b) Geometría que debe tener el ultimo eslabón para que la fuerza que el contacto con el objeto se de en diferentes orientaciones.

I.2.2. Modelos Analíticos

Se desarrollo el calculo espacial para cada cadena cinemática como se observa en la figura I-9. Se gráfica angularmente según las limitaciones físicas del Gripper de 90° para el eslabón superior y de aproximado 75° para el eslabón inferior. También se observa como la cantidad de posiciones que se grafiquen ayudan a obtener un mejor calculo del espacio de trabajo.

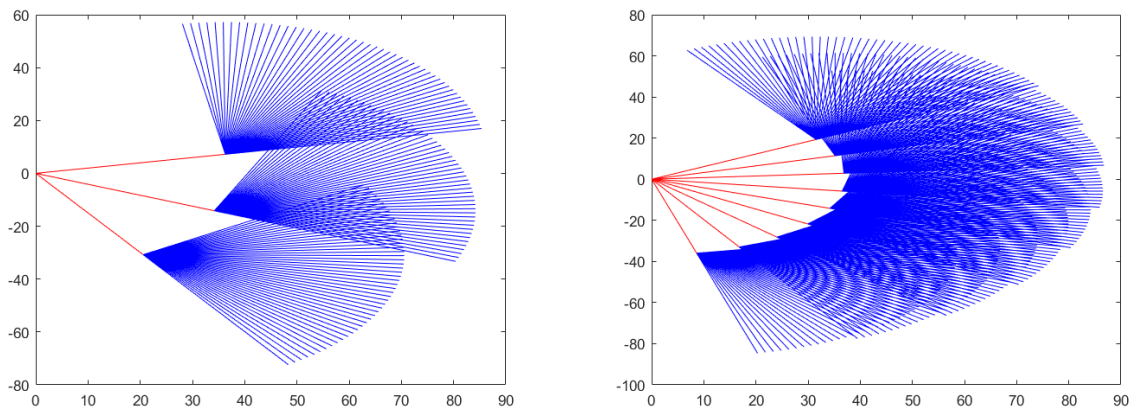


Figura I-9.: Espacio de trabajo para una cadena cinemática. a) Tres posiciones del eslabón inferior. b) Diez posiciones del eslabón inferior.

En la figura **I-10** se aprecia el espacio de trabajo para las dos cadenas cinemáticas. en donde se detalla que el espacio de trabajo supera los 160mm de apertura que es uno de los requerimientos establecidos del Gripper, también cumple con un cierre de 0mm como otro requerimiento propuesto. El cálculo del largo de los eslabones, la separación entre las cadenas cinemáticas y el ángulo de trabajo para cada eslabón se desarrolló en el motor de diseño generativo basado en algoritmos genéticos que proveen el mejor resultado para los requerimiento multiobjetivo propuestos.

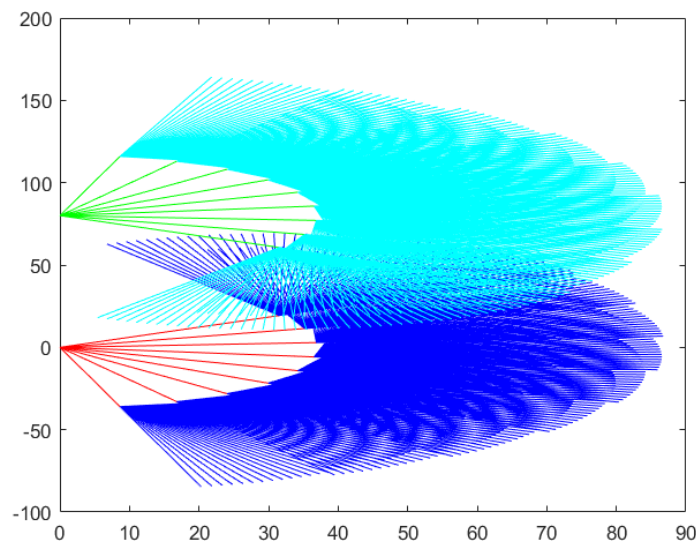


Figura I-10.: Espacio de trabajo del Gripper.

I.2.3. Modelos físicos y prototipos

I.2.4. Prototipos

En el desarrollo del motor de diseño generativo, se tuvo en cuenta los valores y comportamiento de prototipos realizados previamente de cadenas cinemáticas por medio de tensores. En la figura **I-11(a)** se aprecia un prototipo de una cadena cinemática de tres eslabones accionada por un tensor. El accionamiento se realiza por una polea que mientras gira en una dirección solo ejerce tensión en un sentido y cuando cambia de dirección genera la tensión en sentido contrario, por lo que solo es necesario un motor para generar el accionamiento en ambas direcciones.

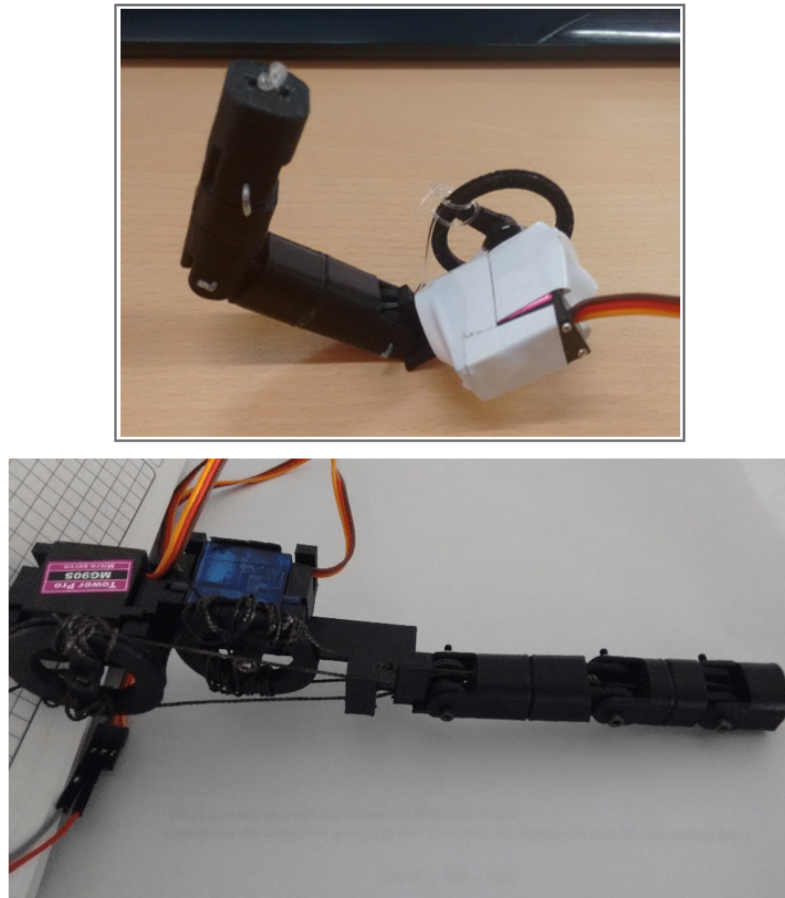


Figura I-11.: Prototipos de cadenas cinemáticas accionada por tensores. a) Accionamiento por un tensor en tres eslabones. b) Accionamiento de por dos tensores en tres eslabones.

El prototipo de la figura **I-11(b)** se observa una cadena cinemática de tres eslabones con doble accionamiento, lo que se desarrolla que el eslabón inferior y el intermedio estaba ligado su posición directamente con un accionamiento, mientras que el eslabón final respondía a un movimiento indirecto de la combinación en posición de los eslabones anteriores. El accionamiento del ultimo eslabón se planteo para garantizar cambios de orientación que ayuden a mantener una buen contacto con el objeto, sin embargo, esta combinación no respondió en todos los casos y por eso se platea la huella en la ultima falange que solvente este problema.

I.2.5. Modelos físicos

Estos modelos se plantean bajo entornos de simulación que permiten observar como se comporta el Gripper bajo diferentes condiciones dinámicas y estáticas de movimiento. La primera posibilidad de simular y calcular la fuerza necesaria de los tensores se observa en la figura

I-12, en el cual se observa que el análisis se aplica en la cadena cinemática inferior.

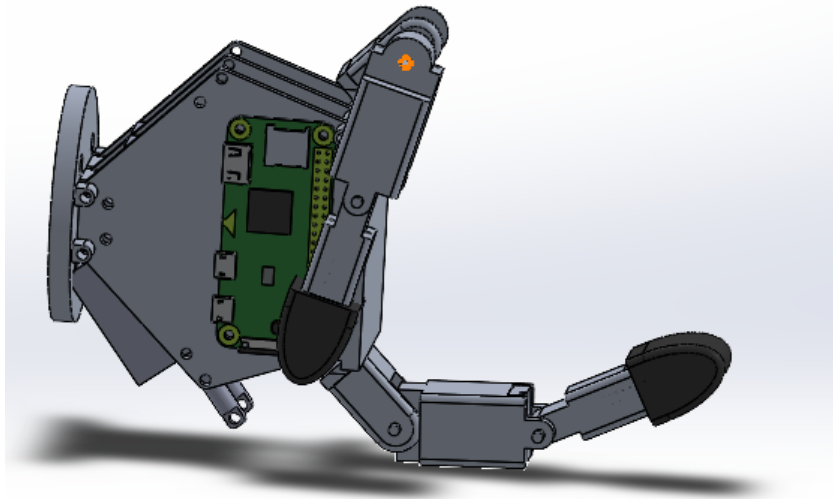


Figura I-12.: Simulación de tensores para la cadena cinemática inferior en laso abierto.

Las condiciones de esta simulación corresponden al posicionamiento estático de los eslabones como se aprecia en la figura I-12. Para este caso se aplica la fuerza de los tensores en el final de cada eslabón simulando la fuerza de los tensores para generar una rotación. El posicionamiento se realiza en laso abierto lo que implica que para cada eslabón se genera una curva de fuerzas calculadas para que logre el objetivo y se mide el torque en el eje de rotación de cada eslabón como se aprecia en la figura I-13.

Este primer enfoque genera buenos resultados, sin embargo, el generar las curvas de fuerzas para posiciones dinámicas implica tener un buen modelamiento matemático de la cadena cinemática por lo cual empieza a ser limitado los ambientes y las variables que se pueden manipulara para observar el comportamiento en distintos puntos críticos de funcionamiento.

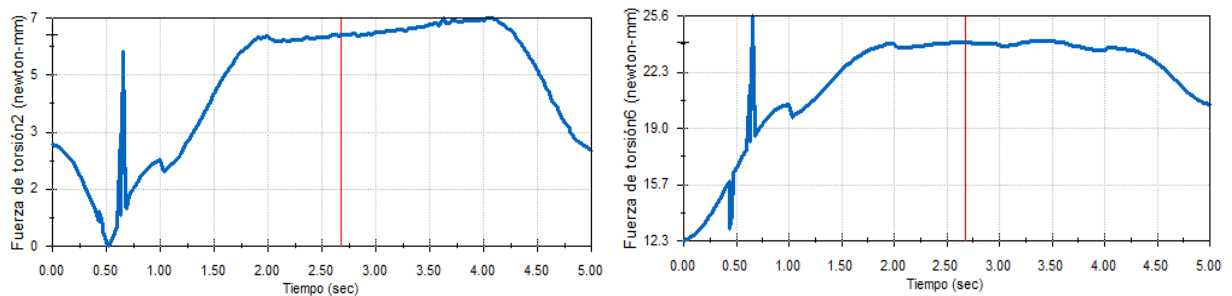


Figura I-13.: Momentos en los ejes de rotación de cada eslabón. a) Eslabón superior. b) Eslabón inferior.

El segundo enfoque se da al generar un objeto el cual manipular y observar para posiciones estáticas de las cadenas cinemáticas el torque que se genera en sus ejes de rotación, como se aprecia en la figura I-14.

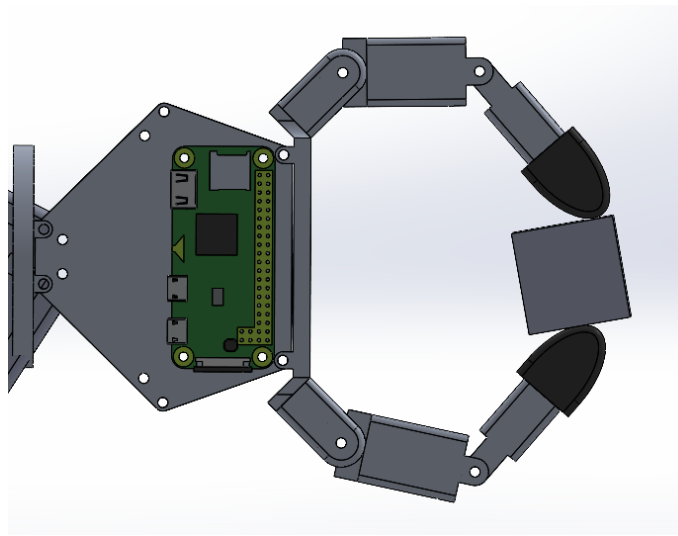


Figura I-14.: Simulación de la interacción de contacto de las cadenas cinemáticas con un objeto de prueba.

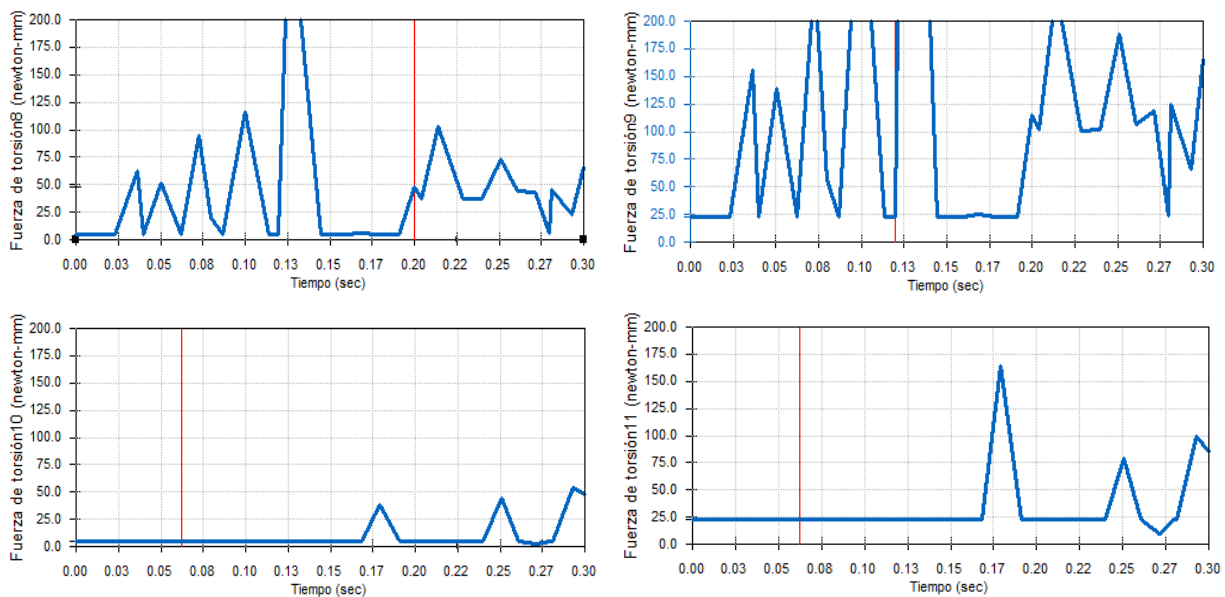


Figura I-15.: Resultados de torques de la interacción de contacto de las cadenas cinemáticas con un objeto de prueba. a) Cadena inferior eslabón superior. b) Cadena inferior eslabón inferior. c) Cadena superior eslabón superior. d) Cadena superior eslabón inferior.

Como se aprecia en la figura **I-15** los resultados en cuestión de torques genera ruido al calcular los contactos de los eslabones superiores con la pieza fabricada. Por tal motivo este enfoque no se usa.

El tercer y ultimo enfoque implica simular en los ejes de rotación servomotores. Con este enfoque tenemos un sistema preciso en seguimiento de trayectorias por lo que podemos calcular el torque en puntos críticos de estudio, el sistema se observa en la figura **I-16**.

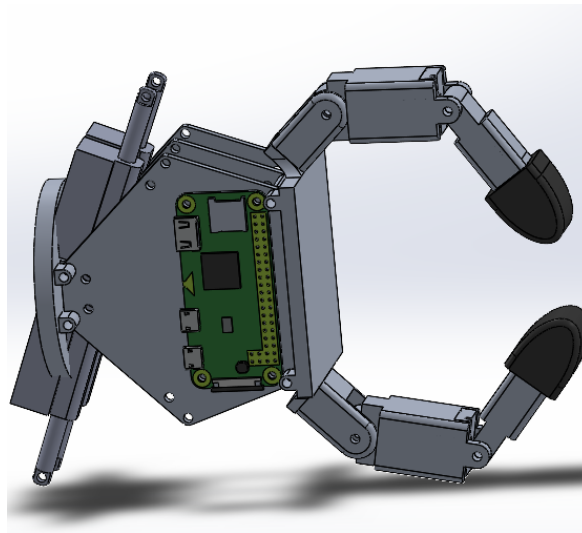


Figura I-16.: Simulación de motores en los ejes de rotación.

En este enfoque podemos medir el torque necesario en cada motor para el seguimiento de trayectoria y luego poder convertir este torque a la fuerza ejercida en el tensor al dividir por la distancia de aplicación de la fuerza en cada eslabón. También se añade la fuerza normal que ejerce la pieza a tomar siempre en dirección de la gravedad para cualquier punto de la trayectoria, sus resultados se observan en la figura **I-17**. Por los resultados obtenidos anteriormente se selecciona este enfoque para el dimensionamiento de los motores lineales que se usaran para los tensores.

La distancia de aplicación de la fuerza del tensor respecto al eje de rotación del eslabón inferior es 6mm, por lo que obtenemos que el motor lineal inferior en la posición crítica debe ser de 48N. La distancia para el eslabón superior es de 4mm lo que indica que el motor debe de ser de mínimo 20.25N.

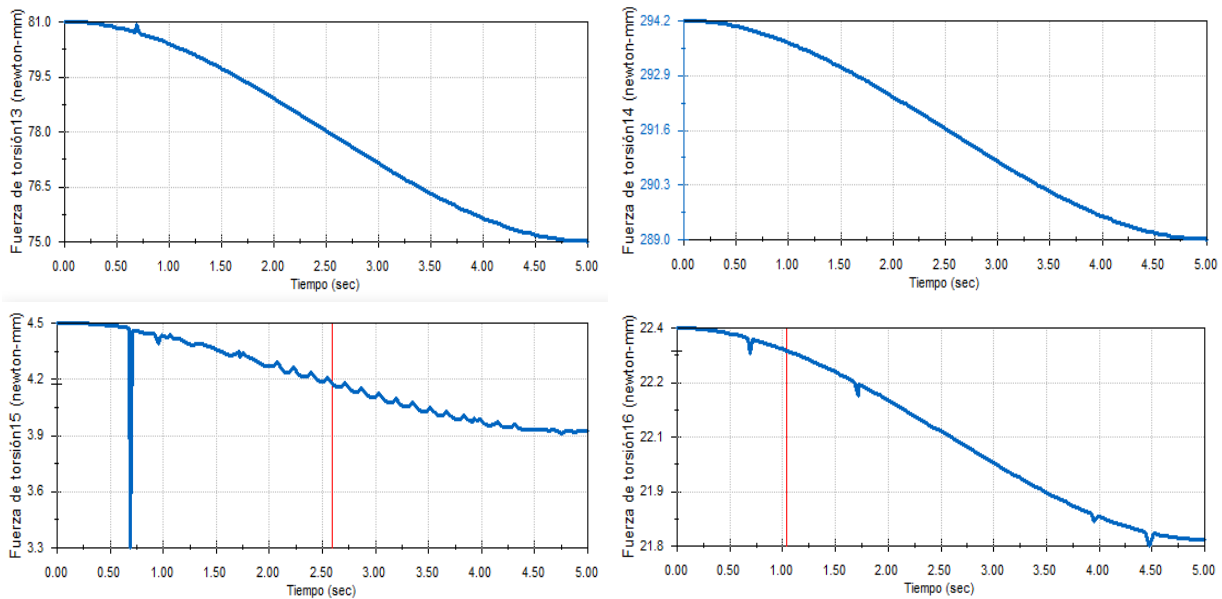


Figura I-17.: Resultados de torques de simulación de motores en los ejes de rotación. a) Cadena inferior eslabón superior. b) Cadena inferior eslabón inferior. c) Cadena superior eslabón superior. d) Cadena superior eslabón inferior.

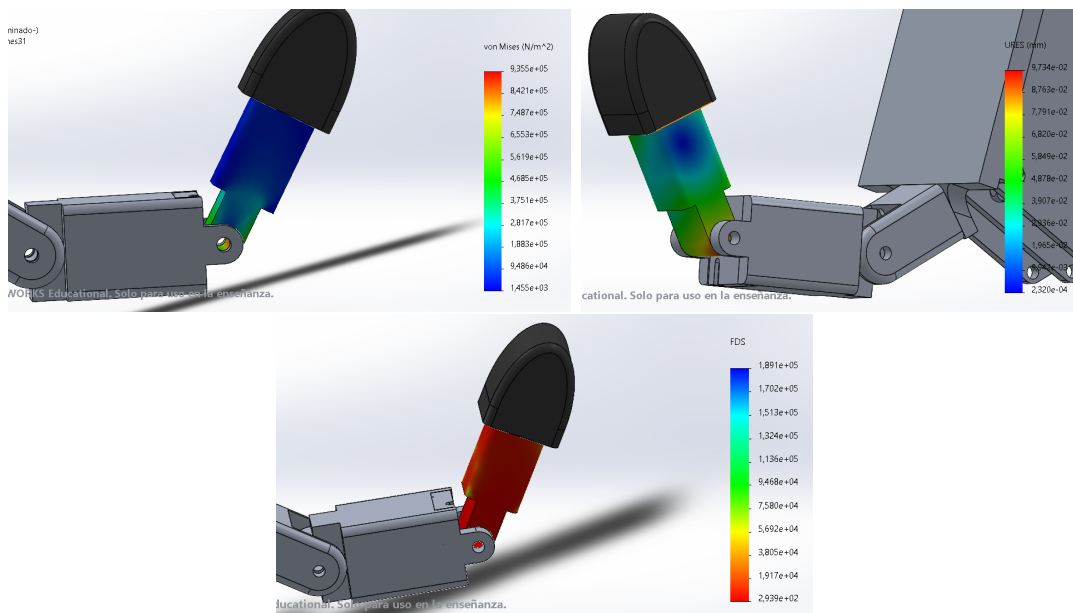


Figura I-18.: Calculo de elementos finitos en el eslabón superior. a) Von Mises. b) Deformaciones. c) Factor de seguridad.

Adicionalmente se realiza el análisis de Von Mises, el calculo de deformaciones y el calculo del factor de seguridad por medio de técnicas de elementos finitas basados en fuerzas dinámicas

calculadas anteriormente. En la figura I-18 se observa la aplicación de estos tres cálculos para el eslabón superior en el punto crítico. En la figura I-19 se aprecia el calculo de elementos finitos para el eje de rotación.

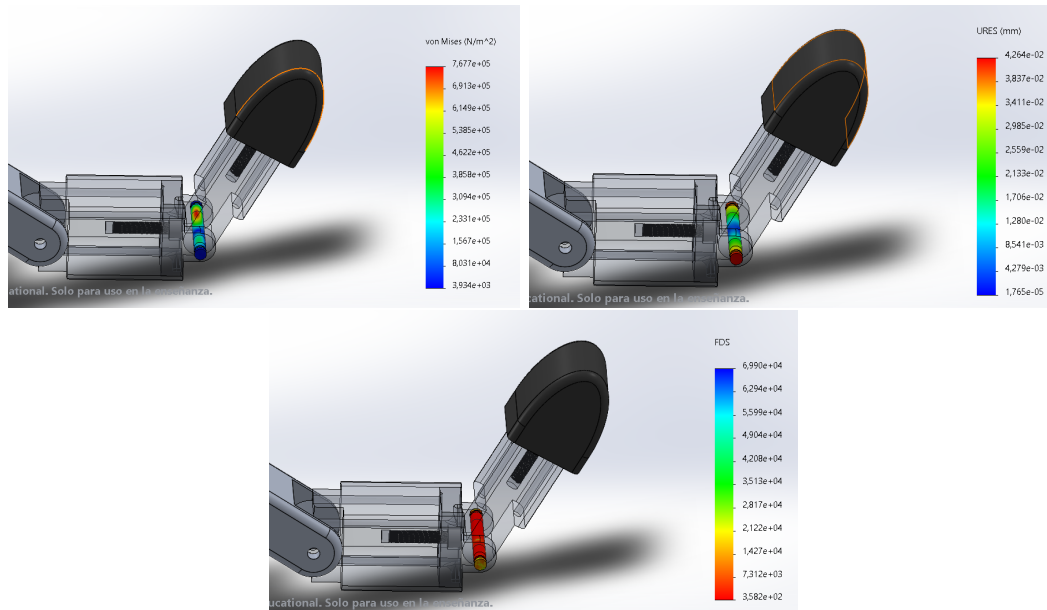


Figura I-19.: Calculo de elementos finitos en el eje de rotación. a) Von Mises. b) Deformaciones. c) Factor de seguridad.

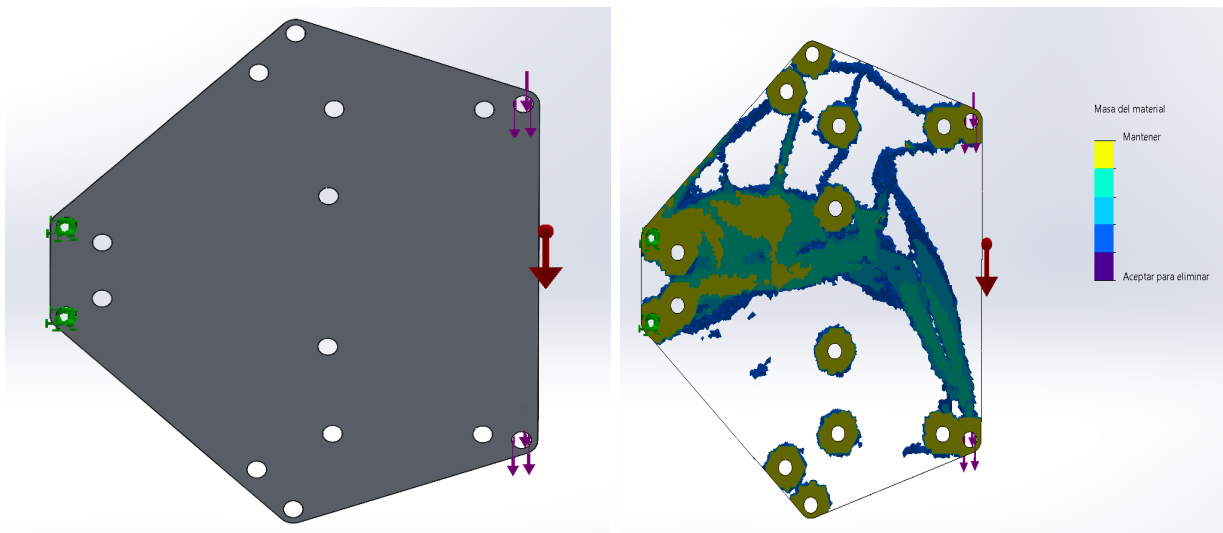


Figura I-20.: Desarrollo del proceso de optimización topológica.

En el desarrollo de la optimización topológica se realiza en las laminas de aluminio que mantienen fijas los elementos y que da soporte a todos los subsistemas del Gripper. En la

figura **I-20** se observa que el proceso se realiza para el peso en un solo sentido, pero el Gripper debe considerar que en distintos momentos esa fuerza es dinámica y cambiara el sentido.

Con el resultado topológico obtenido en la figura **I-20** se diseña la geometría que se presenta en la figura **I-21**. Dicha geometría es simétrica para que responda al cambio de dirección de la aplicación de la fuerza generada por soportar la cadena cinemática y objeto a tomar. En la figura **I-21** también se realiza un estudio topológico para observar el comportamiento de la nueva pieza y se observa que son muy pocas las zonas que sugiere remover para llegar a una pieza mas óptima.

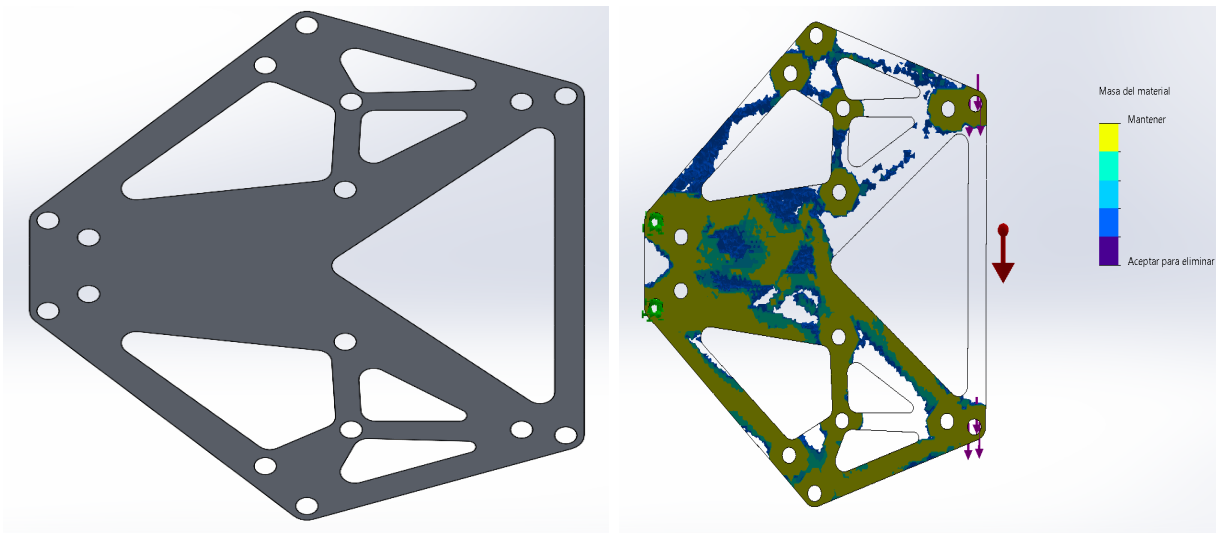


Figura I-21.: Desarrollo del proceso de optimización topológica de verificación.

J. Anexo: Desarrollo método tradicional Gripper: Diseño detallado

J.1. Transmisión efectiva del producto

El Gripper biplanar desarrollado en detalle se observa en la figura **J-1**, el cual cuenta con dos cadenas cinemáticas de dos eslabones cada una, una estructura recubierta por una protección y una base para sujetar al brazo robótico.

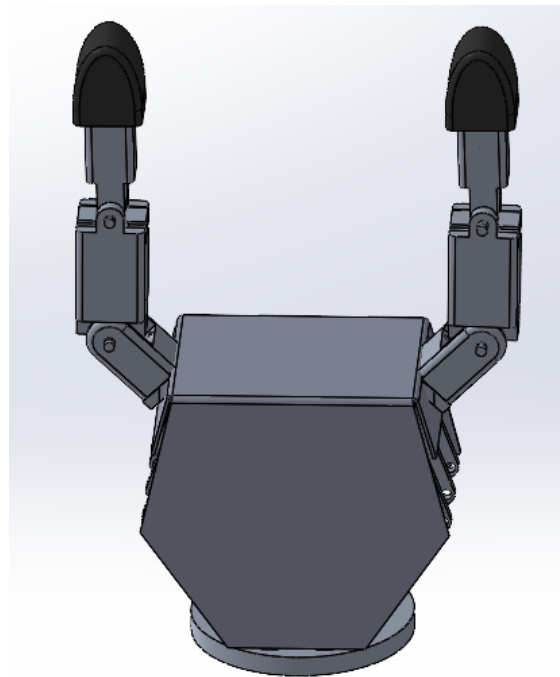


Figura J-1.: Ensamblaje final del Gripper.

Las cadenas cinemáticas se conforman de dos eslabones que son accionados por tensores individualmente por lo que tienen dos grados de libertad cada una. Este conjunto se aprecia en la figura **J-2**. Se detalla que el final del eslabón superior se encuentra con un recubrimiento siliconado que permite mejorar las condiciones de agarre del objeto a manipular, lo que implica que esta será la única parte que tendrá contacto con el objeto.

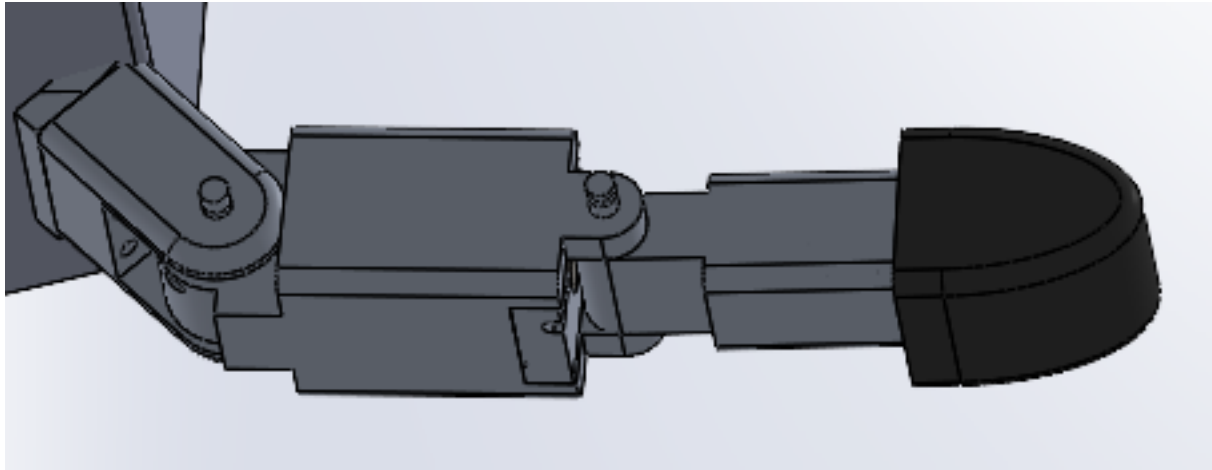


Figura J-2.: Conjunto de la cadena cinemática.

El conjunto de la estructura se observa en la figura **J-3**, el cual esta recubierto por una protección que protege los diferentes elementos de control, potencia y electrónica que son necesarios para accionar las cadenas cinemáticas. Al igual este recubrimiento cumple una funcionalidad estética que provee geometrías mas finas y continuas que provee una visión de unidad al Gripper, dando la sensación de robustez al equipo.

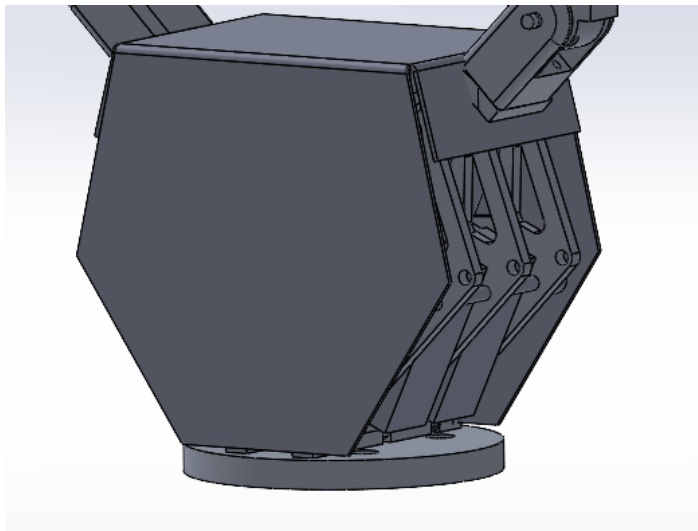


Figura J-3.: Conjunto de la estructura

Los componentes que estan fijos en la estructura se pueden observar en la figura **J-4**. En esta vistas se puede apreciar que la electrónica se encuentra en las secciones exteriores de la estructura fijas mediante tornillería y que los motores, que seria la parte robusta, se encuentra en el interior de la estructura y alineadas con los tensores de las cadenas cinemáticas.

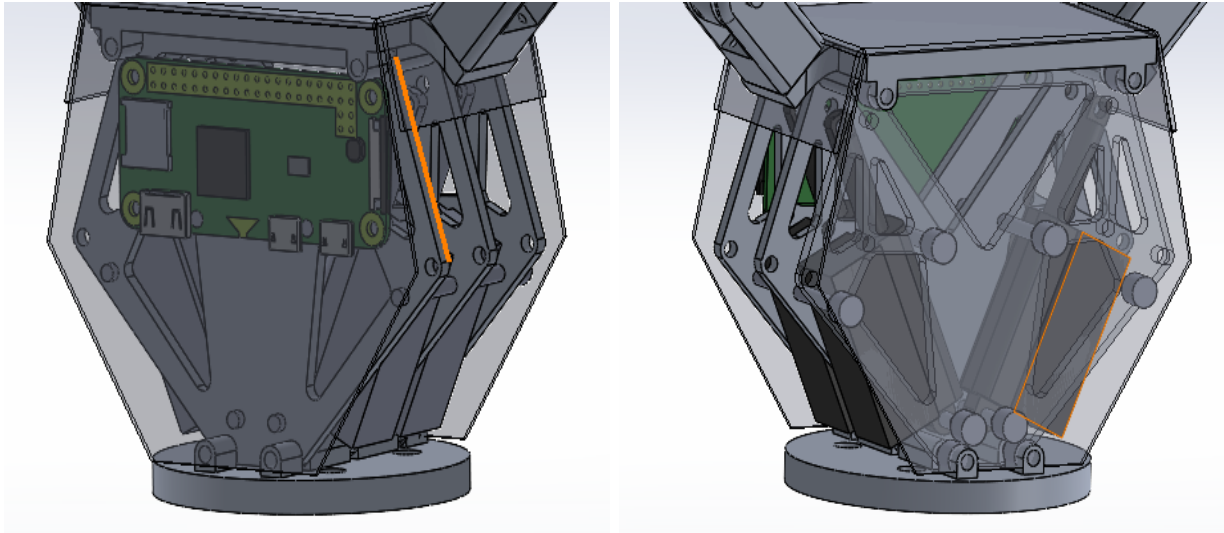


Figura J-4.: Secciones del conjunto de la estructura. a) Electrónica, control y potencia. b) Motores lineales.

Por ultimo la interfaz que conecta el Gripper con el brazo robótico se observa en la figura **J-5**. Esta interfaz es modificable según el brazo robótico al que se integre, y la mayoría de estos integra un sistema de sujeción fija por tornillería. Dado que esta pieza es separable de la estructura no genera mayores costo la integración con otro brazo robótico.

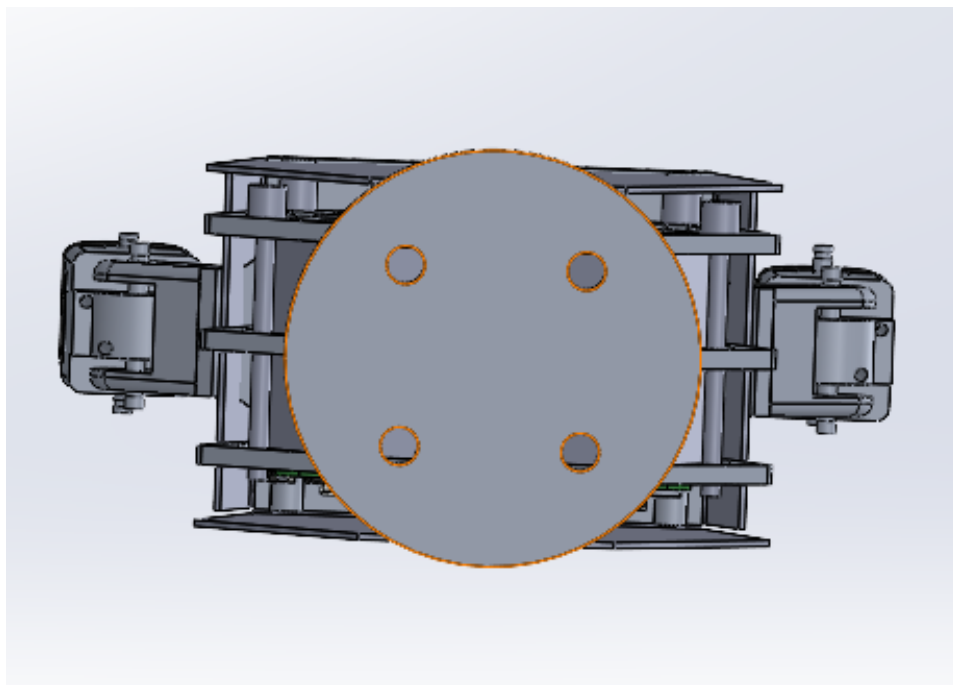


Figura J-5.: Interfaz de sujeción con el brazo robótico.

K. Anexo: gripper_driver.py

```
1) import RPi.GPIO as GPIO
2) import time
3)
4) class GripperDriver:
5)     def __init__(self, max_angle=180):
6)         self.max_angle = max_angle
7)         self.qty_motors = 4
8)         self.angle_motors = [0.0, 0.0, 0.0, 0.0]
9)         self.duty_cycle_motors = [0.0, 0.0, 0.0, 0.0]
10)        self.pin_rpb = [12, 32, 33, 35]
11)        self.servos = []
12)        self.start()
13)
14)    def start(self):
15)        # Set GPIO numbering mode
16)        GPIO.setmode(GPIO.BOARD)
17)        for pos in range(0,self.qty_motors):
18)            GPIO.setup(self.pin_rpb[pos],GPIO.OUT)
19)            servo = GPIO.PWM(self.pin_rpb[pos],50)
20)            self.servos.append(servo)
21)            self.servos[pos].start(0)
22)
23)    def get_angle_motor(self,pos):
24)        return self.angle_motors[pos]
25)
26)    def get_duty_cycle_motor(self,pos):
27)        return self.duty_cycle_motors[pos]
28)    def stop(self):
29)        for pos in range(0,self.qty_motors):
30)            self.servos[pos].stop()
31)            GPIO.cleanup()
32)            self.servos = []
33)
```

```
34)     def set_angle_motor(self,pos,angle):
35)         # pos empieza en cero
36)         if angle <= self.max_angle:
37)             self.angle_motors[pos] = angle
38)         else:
39)             self.angle_motors[pos] = self.max_angle
40)
41)         self.duty_cycle_motors[pos] = ((self.angle_motors[pos]*10)/180)+2
42)         self.servos[pos].ChangeDutyCycle(self.duty_cycle_motors[pos])
43)
44)     def get_status_motor(self,pos):
45)         return {
46)             'Motor': pos,
47)             'Angle': self.angle_motors[pos],
48)             'DutyCycle': self.duty_cycle_motors[pos]
49)         }
```


L. Anexo: gripper_driver_ros_wrapper.py

```
1) import rospy
2) from gripper_driver import GripperDriver
3) from std_msgs.msg import Float32
4) from std_srvs.srv import Trigger
5) from diagnostic_msgs.msg import DiagnosticStatus
6) from diagnostic_msgs.msg import KeyValue
7)
8) class GripperDriverRosWrapper:
9)     def __init__(self):
10)         max_angle_ros = rospy.get_param( "max_angle", 180)
11)         publish_angle_motor_frequency = rospy.get_param( - publish_angle_motor_frequency",
5.0)
12)         publish_duty_cycle_frequency = rospy.get_param( - publish_duty_cycle_frequency",
5.0)
13)         publish_motor_status_frequency = rospy.get_param( - publish_motor_status_frequency",
1.0)
14)
15)         self.gripper = GripperDriver(max_angle=max_angle_ros)
16)         rospy.Service( "stop_gripper", Trigger, self.callback_stop)
17)
18)         rospy.Subscriber( ".angle_command", KeyValue, self.callback_angle_command)
19)
20)         self.angle_motor_pub = rospy.Publisher( "current_angle_motor", Float32,
queue_size=10)
21)         self.duty_cycle_pub = rospy.Publisher( "current_duty_cycle", Float32, queue_size=10)
22)         self.motor_status_pub = rospy.Publisher( "motor_status", DiagnosticStatus,
queue_size=1)
23)
24)         rospy.Timer(rospy.Duration(1.0/publish_angle_motor_frequency ), self.publish_current_angle)
25)         rospy.Timer(rospy.Duration(1.0/publish_duty_cycle_frequency), self.publish_current_duty)
26)         rospy.Timer(rospy.Duration(1.0/publish_motor_status_frequency), self.publish_motor_status)
27)
28)         def stop(self):
```

```
29)         self.gripper.stop()
30)
31)     def callback_stop(self, req):
32)         self.stop()
33)         return {"success": True, "message": "Gripper has been stopped"}
34)
35)     def callback_angle_command(self, msg):
36)         self.gripper.set_angle_motor(int(msg.key),float(msg.value))
37)
38)     def publish_current_angle(self, event=None):
39)         for pos in range(4):
40)             self.angle_motor_pub.publish(self.gripper.get_angle_motor(pos))
41)
42)     def publish_current_duty(self, event=None):
43)         for pos in range(4):
44)             self.duty_cycle_pub.publish(self.gripper.get_duty_cycle_motor(pos))
45)
46)     def publish_motor_status(self, event=None):
47)         for pos in range(4):
48)             status = self.gripper.get_status_motor(pos)
49)             data_list = []
50)             for key in status:
51)                 data_list.append(KeyValue(key, str(status[key])))
52)             msg = DiagnosticStatus()
53)             msg.values = data_list
54)             self.motor_status_pub.publish(msg)
55)
56) if __name__ == "__main__":
57)
58)     rospy.init_node("gripper_driver")
59)
60)     gripper_driver_wrapper = GripperDriverRosWrapper()
61)
62)     rospy.on_shutdown(gripper_driver_wrapper.stop)
63)     rospy.loginfo("Gripper driver is now started, ready to get commands.")
64)
65)     rospy.spin()
```

M. Anexo: Manual de uso del Gripper

El despliegue del Gripper se planea según la arquitectura para cualquier servidor portátil Raspberry pi 4, por lo cual el único requerimiento es tener Docker instalado. La imagen de Docker para descargar se asocia al repositorio del LabFabEx-UN para lo cual al momento de clonar la imagen de un servidor funcional se utilizan los siguientes comandos.

- 1) `docker commit containerName tagToSave`
- 2) `docker tag generic pria UserRepository/generic pria`
- 3) `docker push UserRepository/generic pria`

Luego de clonar la imagen se puede ejecutar la imagen con los siguientes comandos.

- 4) `xhost + local : docker`
- 5) `dockerrun --net = host --env = "DISPLAY" -v/tmp/.X11 - unix : /tmp/.X11 - unix - it --privileged - d --device = /dev/video0 : /dev/video0generic_pria`

Con lo anterior el servidor ya contara con todo lo necesario para manipular el Gripper bajo el entorno de PRIA-UN o manualmente, sin la necesidad de configurar nada del sistema. Por lo tanto ahora se explicara la manipulación del nuevo robot bajo las dos alternativas.

M.1. En PRIA-UN

En PRIA-UN es necesario correr el contenedor.

- 6) `dockerstartnameContainer`

Abrir tres ventanas para ingresar al Bash de cada contenedor y establecer el bash de ROS para cada ventana.

- 7) `dockerexec - itnameContainerbash`
- 8) `source/opt/ros/melodic/devel/setup.bash`

En la primera ventana ejecutamos el servidor maestro de ROS.

9) *roscore*

En la segunda ventana ejecutamos el lanzador definido para los robots genéricos.

10) *roslaunchgenericpriagenericpria.launch*

En la tercera ventana se activa el robot según el diseño de PRIA-UN.

11) *curl -XPOSTlocalhost : 3000/connect*

Con lo anterior el Gripper está disponible para leer la secuencia que se remitan a firebase y ejecutarse como se establece en PRIA-UN.

M.2. Manualmente

Es necesario correr el contenedor.

6) *dockerstartnameContainer*

Abrir tres ventanas para ingresar al Bash de cada contenedor y establecer el bash de ROS para cada ventana.

7) *dockerexec -itnameContainerbash*

8) *source/opt/ros/melodic/devel/setup.bash*

En la primera ventana ejecutamos el servidor maestro de ROS.

9) *roscore*

En la segunda ventana se corre el código que vincula el controlador del Gripper con ROS.

10) *rosrun gripperdriver gripperdriverroswrapper.py*

En la tercera ventana publicamos el mensaje para manipular cada motor según el ángulo deseado.

11) *rostopicpub -1/anglecommanddiagnosticmsgs/KeyValue''key' :! 0,' value' :! 90''*

Con lo anterior se tiene el control manual del Gripper. Adicionalmente en otra ventana se puede observar el estado del motor con los siguientes comandos.

12) *rostopicecho/currentanglemotor*

13) *rostopicecho/currentdutycycle*

14) *rostopicecho/motorstatus*

El controlador del Gripper también tiene desplegados dos servicios para encender o apagar los motores, según lo deseado en cada caso usando los siguientes comandos.

15) *rosservicecall/stopgripper*

16) *rosservicecall/startgripper*

Bibliografía

- [1] J. S. Restrepo Mendoza and E. Córdoba Nieto, “DISEÑO PARAMÉTRICO PARA CLASIFICACIÓN DE FAMILIAS DE PRODUCTOS EN MANUFACTURA DISCRETA EN EL LABFABEXUN,” in *CONGRESO INTERNACIONAL DE INGENIERÍA MECÁNICA, MECATRÓNICA Y AUTOMATIZACIÓN - Memorias 2021*, pp. 21–21, 2021.
- [2] Martin Hankel, “RAMI4.0 – Reference Architecture Model Industry 4.0.,” 11 2016.
- [3] TheOPCFoundation, “RAMI4.0 by Martin Hankel (Bosch-Rexroth) at OPC Day Europe 2016,” 2 2017.
- [4] González, “Measurement of Areas on a Sphere Using Fibonacci and Latitude-Longitude Lattices,” *Mathematical Geosciences*, vol. 42, pp. 49–64, 1 2010.
- [5] Departamento Nacional de Planeación (DNP), Superintendencia de Industria y Comercio (SIC), Dirección Nacional de Derecho de Autor (DNDA), Instituto Colombiano Agropecuario (ICA), Organización Mundial de la Propiedad Intelectual (OMPI), and Misión permanente de Colombia ante las naciones Unidas, “Reporte sobre la información en materia de Propiedad Intelectual en Colombia,” tech. rep., 2017.
- [6] J. Mountstephens and J. Teo, “Progress and challenges in generative product design: A review of systems,” 12 2020.
- [7] Z. Jiang, H. Wen, F. Han, Y. Tang, and Y. Xiong, “Data-driven generative design for mass customization: A case study,” *Advanced Engineering Informatics*, vol. 54, 10 2022.
- [8] P. Wolniak, B. Sauthoff, D. Kloock-Schreiber, and R. Lachmayer, “AUTOMATED PRODUCT FUNCTIONALITY and DESIGN OPTIMIZATION INSTANCING A PRODUCT-SERVICE SYSTEM,” in *Proceedings of the Design Society: DESIGN Conference*, vol. 1, pp. 1405–1414, Cambridge University Press, 2020.
- [9] M. A. S. Al-Shamsi, “Review of Korean Imitation and Innovation in the Last 60 Years,” 3 2022.
- [10] B. Bartikowski, F. Fastoso, and H. Gierl, “Luxury cars Made-in-China: Consequences for brand positioning,” *Journal of Business Research*, vol. 102, pp. 288–297, 9 2019.

-
- [11] K. D. Thoben, S. A. Wiesner, and T. Wuest, “Industrie 4.0 and smart manufacturing—a review of research issues and application examples,” 2017.
- [12] D. G. Ullman, *The Mechanical Design Process*, vol. 1. 2010.
- [13] A. M. Farid and N. P. Suh, *Axiomatic Design in Large Systems*. 2016.
- [14] R. L. Norton, *Diseño de máquinas. Un enfoque integrado*. Pearson Educación, cuarta edi ed., 2011.
- [15] S. BuHamdan, A. Alwisy, and A. Bouferguene, “Generative systems in the architecture, engineering and construction industry: A systematic review and analysis,” *International Journal of Architectural Computing*, 2020.
- [16] D. Nagaraj and D. Werth, “Towards a Generalized System for Generative Engineering,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 1 2020.
- [17] S. Fox, “A preliminary methodology for generative production systems,” *Journal of Manufacturing Technology Management*, vol. 22, no. 3, pp. 348–364, 2011.
- [18] A. N. Pilagatti, G. Vecchi, E. Atzeni, L. Iuliano, and A. Salmi, “Generative Design and new designers’ role in the manufacturing industry,” in *Procedia CIRP*, vol. 112, pp. 364–369, Elsevier B.V., 2022.
- [19] C. Hyunjin, “A Study on Application of Generative Design System in Manufacturing Process,” in *IOP Conference Series: Materials Science and Engineering*, vol. 727, Institute of Physics Publishing, 1 2020.
- [20] J. Wu, M. Li, Z. Chen, W. Chen, X. Wu, and Y. Xi, “Generative Design of the Roller Seat of the Wind Turbine Blade Turnover Machine Based on Cloud Computing,” *ICMAE 2020 - 2020 11th International Conference on Mechanical and Aerospace Engineering*, pp. 212–217, 2020.
- [21] H. Li and R. Lachmayer, “Automated exploration of design solution space applying the generative design approach,” in *Proceedings of the International Conference on Engineering Design, ICED*, vol. 2019-August, pp. 1085–1094, Cambridge University Press, 2019.
- [22] S. Khan and M. J. Awan, “A generative design technique for exploring shape variations,” *Advanced Engineering Informatics*, vol. 38, no. October, pp. 712–724, 2018.
- [23] S. Khan, E. Gunpinar, and B. Sener, “GenYacht: An interactive generative design system for computer-aided yacht hull design,” *Ocean Engineering*, vol. 191, no. August, p. 106462, 2019.

- [24] E. Gunpinar and S. Gunpinar, “A shape sampling technique via particle tracing for CAD models,” *Graphical Models*, vol. 96, no. January, pp. 11–29, 2018.
- [25] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, “Deep generative design: Integration of topology optimization and generative models,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 141, 11 2019.
- [26] P. Ghannad and Y. C. Lee, “Automated modular housing design using a module configuration algorithm and a coupled generative adversarial network (CoGAN),” *Automation in Construction*, vol. 139, 7 2022.
- [27] N. A. Kallioras and N. D. Lagaros, “DzAI: Deep learning based generative design,” *Procedia Manufacturing*, vol. 44, pp. 591–598, 2020.
- [28] N. A. Kallioras and N. D. Lagaros, “Mlgen: Generative design framework based on machine learning and topology optimization,” *Applied Sciences (Switzerland)*, vol. 11, 12 2021.
- [29] J. C. García Carrero, *Planeación de trayectorias en vuelo de un manipulador industrial para el Laboratorio Fábrica Experimental UN*. PhD thesis, Universidad Nacional de Colombia, Bogotá D.C, 2017.
- [30] C. Sarmiento Fautoque, *Desarrollo Teórico- Experimental en la Geometría de Maquinado Múlti-ejes aplicando Ingeniería Inversa Mixta*. PhD thesis, Universidad Nacional de Colombia, Bogotá D.C., 2014.
- [31] V. Granadeiro, L. Pina, J. P. Duarte, J. R. Correia, and V. M. Leal, “A general indirect representation for optimization of generative design systems by genetic algorithms: Application to a shape grammar-based design system,” *Automation in Construction*, vol. 35, pp. 374–382, 2013.
- [32] H. Li and R. Lachmayer, “Generative Design Approach for Modeling Creative Designs,” *IOP Conference Series: Materials Science and Engineering*, vol. 408, no. 1, 2018.
- [33] S. S. Pibal, K. Khoss, and I. Kovacic, “Framework of an algorithm-aided BIM approach for modular residential building information models,” *International Journal of Architectural Computing*, vol. 20, pp. 777–800, 12 2022.
- [34] M. Younus, C. Peiyong, L. Hu, and F. Yuqing, “MES development and significant applications in manufacturing -A review,” in *ICETC 2010 - 2010 2nd International Conference on Education Technology and Computer*, vol. 5, 2010.
- [35] T. A. Jauhar, M. Safdar, I. Kim, and S. Han, “Web-based Product Data Visualization and Feedback between PLM and MES,” in *Proceedings - Web3D 2020: 25th ACM Conference on 3D Web Technology*, Association for Computing Machinery, Inc, 11 2020.

- [36] G. Bruno, A. Faveto, and E. Traini, “An open source framework for the storage and reuse of industrial knowledge through the integration of plm and mes,” *Management and Production Engineering Review*, vol. 11, pp. 62–73, 6 2020.
- [37] E. Traini, G. Bruno, A. Awouda, P. Chiabert, and F. Lombardi, “Integration Between PLM and MES for One-of-a-Kind Production,” in *IFIP Advances in Information and Communication Technology*, vol. 565 IFIP, pp. 356–365, Springer, 2019.
- [38] M. I. Mahmoud, H. H. Ammar, M. M. Hamdy, and M. H. Eissa, “Production operation management using Manufacturing Execution Systems (MES),” in *2015 11th International Computer Engineering Conference: Today Information Society What’s Next?, ICENCO 2015*, pp. 111–116, Institute of Electrical and Electronics Engineers Inc., 2 2016.
- [39] W. Qifeng and W. Zhangjian, “Web services-based system integration approach for manufacturing execution system,” in *Proceedings - 2011 International Conference on Internet Computing and Information Services, ICICIS 2011*, pp. 469–472, 2011.
- [40] S.-H. Jing, Q.-J. Meng, and W.-Q. Cao, “Cement Enterprise MES Key Technology Research and Application,” in *2007 International Conference on Machine Learning and Cybernetics*, pp. 277–282, IEEE, 8 2007.
- [41] W. Qu, W. Cao, and Y. C. Su, “Design and implementation of smart manufacturing execution system in solar industry,” *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [42] Y. Yue-Xina and R. Gong-Chang, “Design of Real Time Data Acquisition System Framework for Production Workshop Based on OPC Technology,” in *MATEC Web of Conferences*, vol. 128, EDP Sciences, 10 2017.
- [43] X. Zeng, “Design and implementation of production management system in aviation machining workshop based on MES,” in *Proceedings - International Conference on Control Science and Electric Power Systems, CSEPS 2021*, pp. 385–388, Institute of Electrical and Electronics Engineers Inc., 5 2021.
- [44] S. Mantravadi, C. Møller, C. LI, and R. Schnyder, “Design choices for next-generation IIoT-connected MES/MOM: An empirical study on smart factories,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, 2 2022.
- [45] J. Barata, P. R. da Cunha, A. S. Gonnagar, and M. Mendes, “Product traceability in ceramic industry 4.0: A design approach and cloud-based MES prototype,” in *Lecture Notes in Information Systems and Organisation*, vol. 26, pp. 187–204, Springer Heidelberg, 2018.

- [46] M. Ko, C. Lee, and Y. Cho, “Design and Implementation of Cloud-Based Collaborative Manufacturing Execution System in the Korean Fashion Industry,” *Applied Sciences (Switzerland)*, vol. 12, 9 2022.
- [47] X. Han, M. Li, and X. Zhang, “Design and key technology of MES for spacecraft assembly,” in *Proceedings - 2016 6th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2016*, pp. 844–847, Institute of Electrical and Electronics Engineers Inc., 12 2016.
- [48] D. F. Tosse, S. Araujo, and E. Córdoba, “Plataforma para integración de máquinas en laboratorio fábrica experimental con enfoque de Industria 4.0,” in *Innovar para educar* (Corporación Cimted© 2020, ed.), vol. 1, pp. 125–146, Medellín, Antioquia – Colombia: Corporación Centro Internacional de Marketing Territorial para la Educación y el Desarrollo, primera ed., 2020.
- [49] R. Y. Zhong, G. Q. Huang, Q. Y. Dai, K. Zhou, T. Qu, and G. J. Hu, “RFID-enabled real-time manufacturing execution system for discrete manufacturing: Software design and implementation,” in *2011 International Conference on Networking, Sensing and Control, ICNSC 2011*, pp. 311–316, 2011.
- [50] Y. Wang, M. Wang, J. Wang, and Y. Zhou, “Design and implementation of device integration framework based on agent technology in MES,” in *Procedia CIRP*, vol. 83, pp. 485–489, Elsevier B.V., 2019.
- [51] G. D’Antonio, F. Segonds, J. S. Bedolla, P. Chiabert, and N. Anwer, “A proposal of manufacturing execution system integration in design for additive manufacturing,” in *IFIP Advances in Information and Communication Technology*, vol. 467, pp. 761–770, Springer New York LLC, 2016.
- [52] M. Naedele, H. M. Chen, R. Kazman, Y. Cai, L. Xiao, and C. V. Silva, “Manufacturing execution systems: A vision for managing software development,” *Journal of Systems and Software*, vol. 101, pp. 59–68, 3 2015.
- [53] T. Masood and R. H. Weston, “Modelling framework to support decision-making in manufacturing enterprises,” *Advances in Decision Sciences*, vol. 2013, 2013.
- [54] H. Habib, R. Menhas, and O. McDermott, “Managing Engineering Change within the Paradigm of Product Lifecycle Management,” *Processes*, vol. 10, 9 2022.
- [55] M. Hayat and H. Winkler, “Exploring the Basic Features and Challenges of Traditional Product Lifecycle Management Systems,” in *IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 2022-December, pp. 762–766, IEEE Computer Society, 2022.

- [56] S. Rädler and E. Rigger, “A Survey on the Challenges Hindering the Application of Data Science, Digital Twins and Design Automation in Engineering Practice,” in *Proceedings of the Design Society*, vol. 2, pp. 1699–1708, Cambridge University Press, 5 2022.
- [57] S. Nzetchou, A. Durupt, S. Remy, and B. Eynard, “Semantic enrichment approach for low-level CAD models managed in PLM context: Literature review and research prospect,” *Computers in Industry*, vol. 135, 2 2022.
- [58] M. Lennartsson, S. André, and F. Elgh, “PLM support for design platforms in industrialized house-building,” *Construction Innovation*, 2 2021.
- [59] V. Kopei, O. Onysko, C. Barz, P. Dašić, and V. Panchuk, “Designing a Multi-Agent PLM System for Threaded Connections Using the Principle of Isomorphism of Regularities of Complex Systems,” *Machines*, vol. 11, 2 2023.
- [60] Y. Liao, F. Deschamps, E. d. F. R. Loures, and L. F. P. Ramos, “Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal,” *International Journal of Production Research*, vol. 55, no. 12, pp. 3609–3629, 2017.
- [61] J. W. Veile, D. Kiel, J. M. Müller, and K. I. Voigt, “Lessons learned from Industry 4.0 implementation in the German manufacturing industry,” *Journal of Manufacturing Technology Management*, 2019.
- [62] “Status Report Reference Architecture Model Industrie 4.0 (RAMI4.0),” tech. rep., 2015.
- [63] A. F. Cifuentes Gómez, *Implementación de sistemas de gestión de información del ciclo de vida de producto basado en el desarrollo de un molde de inyección*. PhD thesis, Universidad Nacional de Colombia, Bogotá, Colombia, 2021.
- [64] P. Andrés and C. Parra, “Modelo e-Manufacturing bajo la arquitectura Cloud Manufacturing para el Laboratorio Fábrica Experimental UN,” tech. rep., 2015.
- [65] M. K. Mohanty, P. Gahan, and S. Choudhury, “Why most of the supplier development programs fail in discrete manufacturing – findings from selected Indian discrete manufacturing industries,” 2014.
- [66] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [67] A. K. Sethi and S. P. Sethi, “Flexibility in manufacturing: A survey,” *International Journal of Flexible Manufacturing Systems*, vol. 2, no. 4, pp. 289–328, 1990.

- [68] S. K. Saren and V. Tiberiu, "Review of Flexible Manufacturing System Based on Modeling and Simulation," *ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering.*, vol. Volume XXV, no. 1, 2016.
- [69] G. Kim, Y. Kwon, E. S. Suh, and J. Ahn, "Analysis of Architectural Complexity for Product Family and Platform," *Journal of Mechanical Design, Transactions of the ASME*, vol. 138, no. 7, pp. 1–11, 2016.
- [70] O. Asikoglu and T. W. Simpson, "A new method for evaluating design dependencies in product architectures," *12th AIAA Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. September, pp. 1–11, 2012.
- [71] M. Lafou, L. Mathieu, S. Pois, and M. Alochot, "Manufacturing System Flexibility: Product Flexibility Assessment," *Procedia CIRP*, vol. 41, pp. 99–104, 2016.
- [72] F. M. Kasie, G. Bright, and A. Walker, "Decision support systems in manufacturing: a survey and future trends," *Journal of Modelling in Management*, vol. 12, no. 3, pp. 432–454, 2017.
- [73] J. Igba, K. Alemzadeh, P. M. Gibbons, and K. Henningsen, "A framework for optimising product performance through feedback and reuse of in-service experience," *Robotics and Computer-Integrated Manufacturing*, vol. 36, pp. 2–12, 2015.
- [74] M. von Stietencron, K. A. Hribernik, C. C. Røstad, B. Henriksen, and K. D. Thoben, "Applying closed-loop product lifecycle management to enable fact based design of boats," in *IFIP Advances in Information and Communication Technology*, vol. 517, pp. 522–531, Springer New York LLC, 2017.
- [75] C. a. Coello Coello and G. B. Lamont, *Applications Of Multi-Objective Evolutionary Algorithms*. 2004.
- [76] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, 2008.
- [77] G. Chiandussi, M. Codegone, S. Ferrero, and F. E. Varesio, "Comparison of multi-objective optimization methodologies for engineering applications," *Computers and Mathematics with Applications*, vol. 63, no. 5, pp. 912–942, 2012.
- [78] G. Ortega, E. Filatovas, E. M. Garzón, and L. G. Casado, "Non-dominated sorting procedure for Pareto dominance ranking on multicore CPU and/or GPU," *Journal of Global Optimization*, vol. 69, pp. 607–627, 11 2017.

- [79] L. B. De Oliveira, C. G. Marcelino, A. Milanes, P. E. Almeida, and L. M. Carvalho, “A successful parallel implementation of NSGA-II on GPU for the energy dispatch problem on hydroelectric power plants,” in *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 4305–4312, Institute of Electrical and Electronics Engineers Inc., 11 2016.
- [80] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [81] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pp. 95–100, 2001.
- [82] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, “Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients,” *Information Sciences*, vol. 177, pp. 5033–5049, 11 2007.
- [83] X. Li, “A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization,” *Genetic and Evolutionary Computation — GECCO 2003*, vol. 2723, pp. 37–48, 6 2003.
- [84] R. Sedgewick and K. Wayne, “Algorithms,” tech. rep.
- [85] J. J. Craig, P. Prentice, and P. P. Hall, “Introduction to Robotics Mechanics and Control Third Edition,” tech. rep., 2005.
- [86] I. Viana, J.-J. Orteu, N. Cornille, and F. Bugarin, “Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model,” *Journal of Electronic Imaging*, vol. 24, no. 6, p. 061118, 2015.
- [87] H. Huang, J. Liu, S. Liu, T. Wu, and P. Jin, “A method for classifying tube structures based on shape descriptors and a random forest classifier,” *Measurement: Journal of the International Measurement Confederation*, vol. 158, p. 107705, 2020.
- [88] F. Hui, P. Payeur, and A. M. Cretu, “Visual tracking of deformation and classification of non-rigid objects with robot hand probing,” *Robotics*, vol. 6, no. 1, 2017.
- [89] J. K. Oh, S. Lee, and C. H. Lee, “Stereo vision based automation for a bin-picking solution,” *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 362–373, 2012.
- [90] E. Gunpinar and S. Khan, *A multi-criteria based selection method using non-dominated sorting for genetic algorithm based design*. No. 65, Springer US, 2019.