



**EBOOK INTERACTIVO PARA LA ENSEÑANZA-APRENDIZAJE DE LA
COMPRESIÓN DE DATOS**

YULIAN CEDEÑO OCAMPO

UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE INGENIERÍA, DEPARTAMENTO DE INGENIERÍA DE SISTEMAS E
INDUSTRIAL
BOGOTÁ D.C., COLOMBIA

2023

**EBOOK INTERACTIVO PARA LA ENSEÑANZA-APRENDIZAJE DE LA
COMPRESIÓN DE DATOS**

YULIAN CEDEÑO OCAMPO

**Tesis de grado para optar el título de
Magister en Ingeniería – Telecomunicaciones**

Director:

Mg. Edgar Miguel Vargas Chaparro

Profesor Asociado

Departamento de Ingeniería de Sistemas e Industrial

Línea de Investigación:

Señales e Información

UNIVERSIDAD NACIONAL DE COLOMBIA

**FACULTAD DE INGENIERÍA, DEPARTAMENTO DE INGENIERÍA DE SISTEMAS E
INDUSTRIAL**

BOGOTÁ D.C., COLOMBIA

2023

Agradecimientos

Quiero expresar mi profunda gratitud a las personas quienes me apoyaron para alcanzar el éxito en este trabajo final de maestría. Deseo agradecer a mi director, profesor Edgar Miguel Vargas Chaparro, a quien, gracias a su apoyo, orientación y dedicación, a lo largo de este proyecto, permitieron dar forma a mi investigación y mejorar su calidad, y cuyos resultados al día de hoy son muestra de su compromiso con mi crecimiento académico y fomento de la investigación. Gracias totales por sus directrices, comentarios y sugerencias.

También agradezco a mi familia por su amor incondicional, a mi padre y madre a quienes amo profundamente, a mi hermana, a mi hijo y a mi novia, cuyos ánimos en esos momentos difíciles y constante apoyo siempre me impulsaron a seguir adelante. Gracias por su aliento, paciencia y todos los sacrificios realizados para lograr construirme hacia el conocimiento, la ciencia y la excelencia académica.

Finalmente, extiendo mi especial agradecimiento a todas las personas y entidades que de alguna manera contribuyeron a mi crecimiento académico y al desarrollo satisfactorio de este trabajo final de maestría.

Resumen

EBOOK INTERACTIVO PARA LA ENSEÑANZA-APRENDIZAJE DE LA COMPRESIÓN DE DATOS

En el estudio de cursos con temáticas complejas, como es el caso de la comprensión de datos, los estudiantes experimentan dificultad en su aprendizaje debido a lo abstracto de dichas teorías y la ausencia de recursos digitales interactivos. Por lo tanto, el objetivo de este proyecto es realizar el diseño y desarrollo de un eBook interactivo que apoye la enseñanza – aprendizaje de la comprensión de datos. La metodología utilizada para el desarrollo del eBook está inspirada en el modelo de diseño instruccional ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación).

Como resultado del análisis, diseño y desarrollo se obtuvo un eBook interactivo compuesto por módulos de aprendizaje orientados a las siguientes temáticas: conceptos básicos de la comprensión de datos, introducción a la teoría de la información, técnicas de compresión sin pérdidas (métodos estadísticos, compresión basada en contexto y métodos de diccionario) y técnicas de compresión con pérdida.

Para la implementación se realiza una encuesta de contexto al público objetivo (estudiantes del curso de “Teoría de la información y sistemas de comunicación” de la Universidad Nacional de Colombia), se identifican y mitigan las amenazas a la validez que podrían afectar la investigación, se realiza el diseño de una prueba de *pre/post test* y de una prueba de usabilidad, y se diseña y ejecuta una estrategia instruccional en la que se aplica el *pre-test* al público objetivo, se realiza la intervención educativa utilizando el eBook interactivo, y se aplica el *post-test* y la prueba de usabilidad al público objetivo.

Con respecto a la evaluación de la efectividad de la acción formativa en el público objetivo, las pruebas estadísticas indican que existe evidencia significativa para afirmar que fue efectiva la intervención pedagógica con el eBook interactivo.

Palabras clave: compresión de datos, eBook interactivo, enseñanza, aprendizaje.

Abstract

INTERACTIVE EBOOK FOR TEACHING-LEARNING OF DATA COMPRESSION

In the study of courses with complex topics, as is the case of data compression, students experience difficulty in their learning due to the abstract nature of these theories and the absence of interactive digital resources. Therefore, the objective of this project is to design and develop an interactive eBook to support the teaching-learning about data compression. The methodology used for the development of the eBook is inspired by the instructional design model ADDIE (Analysis, Design, Development, Implementation, and Evaluation).

As a result of the analysis, design, and development, an interactive eBook was obtained composed of learning modules oriented to the following topics: basics of data compression, introduction to information theory, lossless compression techniques (statistical methods, context-based compression, and dictionary methods) and lossy compression techniques.

For the implementation, a context survey is carried out on the target audience, the threats to validity that could affect the investigation are identified and mitigated, a *pre/post test* and a usability test are designed, and an instructional strategy is executed in which the *pre-test* is applied to the target audience, the educational intervention is carried out using the interactive eBook, and the *post-test* and the usability test are applied to the target audience.

Regarding the evaluation of the effectiveness of the training action in the target audience, the statistical tests indicate that there is significant evidence to affirm that the pedagogical intervention with the interactive eBook was effective.

Keywords: data compression, interactive eBook, teaching, learning

Este Trabajo Final de maestría fue calificado en septiembre de 2023 por el siguiente evaluador:

Jorge Eduardo Ortiz Triviño PhD.
Profesor Facultad de Ingeniería
Universidad Nacional de Colombia

Contenido

Resumen.....	iii
Abstract.....	iv
Contenido.....	v
Lista de figuras.....	viii
Lista de tablas	x
1. Aspectos disciplinares: compresión de datos	5
1.1 Conceptos básicos de la compresión de datos.....	5
1.1.1 Técnicas de compresión de datos	6
1.1.2 La compresión de datos y los sistemas de comunicación.....	7
1.1.3 Medidas de rendimiento de la compresión.....	8
1.2 Breve introducción a la teoría de la información	12
1.2.1 Fuentes de información	12
1.2.2 Contenido de información de un símbolo	12
1.2.3 Información promedio o entropía.....	14
1.3 Codificación de fuente	16
1.3.1 Longitud promedio de código	17
1.3.2 Teorema de codificación de fuente	17
1.3.3 Clasificación de los códigos.....	18
1.4 Métodos Estadísticos	20
1.4.1 Codificación Huffman.....	20
1.4.2 Códigos aritméticos.....	24
1.5 Compresión basada en contexto.....	30
1.5.1 Codificación run-length (RLE)	30
1.5.2 Codificación mover-al-frente (MTF)	31
1.5.3 Codificación PPM	32
1.6 Métodos de diccionario.....	37
1.6.1 Lempel-Ziv 77 (LZ77)	38
1.6.2 Lempel-Ziv-Welch (LZW).....	42
1.7 Técnicas de compresión con pérdida	46
1.7.1 Cuantización escalar.....	47
1.7.2 Codificación diferencial	54
1.7.3 Codificación de transformación	64

1.7.4	Esquemas de análisis/síntesis	73
2.	Aspectos didácticos: eBooks interactivos	85
2.1	Reseña histórica de la evolución de los eBooks.....	85
2.2	Taxonomía de los eBooks	87
2.3	Concepto de eBook interactivo	88
2.4	Arquitectura de los eBooks interactivos	90
2.5	Características del eBook interactivo.....	93
2.6	Beneficios de los eBooks interactivos en ambientes educativos	94
3.	Revisión de trabajos previos	96
3.1	Recolección de la información.....	96
3.2	Análisis de la información	98
3.2.1	The Development of Interactive Physics eBook in Rigid Body Equilibrium and Rotational Dynamics98	
3.2.2	New Interactive Books for Control Education	99
3.2.3	Design and research potential of interactive textbooks: the case of fractions.....	99
3.2.4	Using interactive multimedia eBooks for learning blood cell morphology in pediatric hematology.....	100
3.2.5	The development of a GameBook for education.....	101
3.3	Discusión y conclusiones de la revisión	101
4.	Metodología: análisis, diseño y desarrollo del eBook interactivo.....	103
4.1	Fase de análisis	103
4.1.1	Caracterización del público objetivo.....	103
4.1.2	Definición de los objetivos de aprendizaje.....	104
4.1.3	Definición del contenido de enseñanza	106
4.1.4	Descripción del enfoque pedagógico	107
4.1.5	Selección de la herramienta de desarrollo.....	107
4.2	Fase de diseño	112
4.2.1	Vista estática: Diseño Conceptual.....	112
4.2.2	Vista Funcional: Modos de ejecución.....	114
4.2.3	Estructuras de navegación.....	115
4.3	Desarrollo.....	120
4.3.1	Prototipo del eBook.....	120
4.3.2	Pruebas al prototipo con grupo focal	124
4.3.3	Resultados de las pruebas al prototipo	125
5.	Metodología: implementación y evaluación del eBook interactivo	127
5.1	Implementación del eBook interactivo	127

5.1.2	Identificación y mitigación de las amenazas a la validez.....	128
5.1.3	Diseño de prueba pre-test/post-test y de prueba de usabilidad	129
5.1.4	Diseño y ejecución de la estrategia instruccional.....	130
5.2	Evaluación del eBook interactivo	132
5.2.1	Evaluación de la efectividad de la acción formativa en el público objetivo	132
5.2.2	Evaluación de la usabilidad del eBook interactivo por parte del público objetivo.	135
5.2.3	Análisis de resultados.....	138
6.	Conclusiones y recomendaciones.....	141
6.1	Conclusiones.....	141
6.2	Recomendaciones	143
ANEXOS	144
A.	Anexo: Muestras de scripts de código desarrollados	144
A-1.	Scripts de Cámara y Navegabilidad.....	144
B.	Anexo: Prueba de pre/post-test	149
B-1.	Cuestionario del pre/post-test.....	149
B-2.	Solución cuestionario pre/post-test	154
C.	Anexo: Test e informe de usabilidad	160
C-1.	Test de usabilidad	160
C-2.	Informe de usabilidad	163
D.	Anexo: Solución a los ejercicios del eBook Interactivo	169
E.	Anexo: Productos académicos relacionados con el trabajo final	178
F.	Anexo: Software eBook Interactivo Compresión de Datos.	184
	Referencias bibliográficas.....	187

Lista de figuras

Figura 1. Compresión y reconstrucción (descompresión) (Sayood, 2017).	6
Figura 2. Diagrama de bloques simplificado de un sistema de comunicación (Sayood, 2017).	8
Figura 3. Codificación de fuente. Tomada de (Hwei, 2003).	16
Figura 4. Decodificación Huffman	23
Figura 5. Imágenes monocromáticas de ejemplo.	29
Figura 6. Ejemplo de la codificación LZ77 (Suman, Anjali, Sonal, 2015)	39
Figura 7. Decodificación de la tripla $\langle 7, 4, r \rangle$	40
Figura 8. (a) Mapeo para un codificador de 3 bits. A cualquier entrada que se reciba se le asignará una palabra de código dependiendo del intervalo en el que caiga dicha entrada. Como se están usando 3 bits para representar cada valor, este codificador se conoce como codificador (cuantizador) de 3 bits. (b) Mapeo del decodificador correspondiente al codificador de (a). (c) Mapa de entrada-salida del cuantizador. Dependiendo del intervalo en el que caiga cada valor de muestra que acepta el cuantizador, este proporciona una palabra de código de salida y un valor de representación (Sayood, 2017).	48
Figura 9. Cuantizaciones imagen de prueba. a) imagen original; b) 1bit/pixel; c) 2bits/pixel; d) 3bits/pixel.	53
Figura 10. (a) Histograma imagen de ejemplo. (b) Histograma diferencias pixel-a-pixel imagen de ejemplo (Sayood, 2017).	55
Figura 11. Sistema de codificación diferencial. Se dibujó un rectángulo punteado alrededor de la parte del codificador que imita al decodificador, para obtener una copia de la muestra reconstruida utilizada para generar la siguiente diferencia (Sayood, 2017).	57
Figura 12. Onda sinusoidal y diferencias muestra a muestra.	58
Figura 13. Reconstrucciones onda sinusoidal.	59
Figura 14. Sistema DPCM (Sayood, 2017).	60
Figura 15. Segmento de voz. Un hombre pronunciando la palabra “test”.	62
Figura 16. Secuencia reconstruida usando un predictor de 3er. orden y un cuantizador uniforme de 8 niveles. La secuencia reconstruida luce como la original, pero hay distorsión significativa en áreas donde los valores de salida de la fuente son pequeños, debido a que en estas regiones la entrada al cuantizador es cercana a cero. Como el cuantizador no tiene un nivel de salida “0”, su salida cambia entre los dos niveles más internos. Si se escucha esta señal, se oiría un silbido en la señal reconstruida.	63
Figura 17. Proceso de compresión JPEG.	66
Figura 18. Tablas de cuantización estándar JPEG.	68
Figura 19. Escaneo en zig-zag del bloque JPEG (Blelloch, 2013).	70
Figura 20. (a) Una imagen de tonos discretos. (b) Sus coeficientes de la DCT. (c) Cuantización ligera truncando a enteros. (d) La IDCT, resultados deficientes.	71
Figura 21. (a) Un patrón de tonos continuos. (b) Sus coeficientes de la DCT. (c) Cuantización profunda, solo 4 coeficientes distintos de 0.	72
Figura 22. Transformaciones afines en \mathbb{R}^2 . Tomada de (Hernández & Niño, 1992).	74
Figura 23. Transformación afín contractiva.	75
Figura 24. Transformaciones afines generadas por el IFS de la	76
Figura 25. Aplicación del IFS de ejemplo a tres imágenes iniciales distintas (Salomon & Motta, 2010).	77
Figura 26. Hoja de olmo y collage.	78
Figura 27. Imagen original Elif.	82
Figura 28. Primeras seis iteraciones de la decodificación fractal.	83
Figura 29. Reconstrucción final imagen Elif.	83
Figura 30. Diagrama UML casos de uso para la arquitectura del eBook interactivo.	92

Figura 31. Proceso de recolección de información: Aplicación de la ecuación de búsqueda y criterios de inclusión y exclusión.	96
Figura 32. Vista estática. Mockup de la Perspectiva Inicio	112
Figura 33. Mockup PC interactivo - actividad de evaluación	113
Figura 34. Mockup PC interactivo – pantalla de progreso.	113
Figura 35. Mockup modo de estudio - creación de perfil con token de usuario.	114
Figura 36. Mockup modo de lectura – focalización en el libro de estudio.	115
Figura 37. Mockup modo de estudio - botones interactivos en el libro de estudio.	116
Figura 38. Mockup pantalla inicial y menú principal.	117
Figura 39. Esquema de navegación Menú Principal	118
Figura 40. Esquema de navegación Menú de Pausa.	119
Figura 41. Entorno y elementos de interacción - PC interactivo.	120
Figura 42. Entorno y elementos de interacción - Libro de estudio.	121
Figura 43. Entorno y elementos de interacción - Interfaz de usuario	122
Figura 44. Cámara y Navegabilidad.	122
Figura 45. Pantalla créditos y comunicación.	124
Figura 46. Diagrama de cajas para los puntajes obtenidos en el pre-test y el post-test.	133
Figura 47. Prueba de rangos de Wilcoxon.	135
Figura 48. Resultados de evaluación de usabilidad.	137
Figura 49. Promedio de las calificaciones por categoría de las preguntas del test de usabilidad.	137

Lista de tablas

Tabla 1. Códigos binarios (Hwei, 2003).	18
Tabla 2. Código Huffman generado.	22
Tabla 3. Compresión de audio con codificación Huffman (Sayood, 2017).	23
Tabla 4. Resultados de la compresión de imágenes de ejemplo usando codificación aritmética adaptativa.	29
Tabla 5. Códigos Huffman RLE ITU-T T4 (Blelloch, 2013).	31
Tabla 6. Ejemplo codificación MTF (Wikipedia).	32
Tabla 7. Estructura de datos PPM tras codificar el texto “assanissimassa”; incluye las frecuencias (f) y las probabilidades (p) (Salomon & Motta, 2010).	35
Tabla 8. Actualización estructura de datos PPM.	36
Tabla 9. Códigos para la representación de la longitud de coincidencia. Se muestra el índice, el número de bits selectores para seguir el índice, y las longitudes representadas por el índice y los bits selectores. Por ejemplo, el índice 277 representa el rango de longitudes desde 67 hasta 82, para especificar cuál de los 16 valores realmente ocurrió, el código es seguido por 4 bits selectores.	41
Tabla 10. Código Huffman para el alfabeto de la longitud de coincidencia.	41
Tabla 11. Parámetros para cuantizar la senoide con un	58
Tabla 12. Resultados del cálculo para examinar el desempeño de los predictores de orden 1, 2 y 3.	62
Tabla 13. Código IFS de transformaciones.	76
Tabla 14. Ecuación de búsqueda en bases de datos.	96
Tabla 15. Criterios de inclusión y exclusión.	97
Tabla 16. Lista de artículos seleccionados en el proceso de recolección de información.	97
Tabla 17. Resumen de análisis de las publicaciones.	102
Tabla 18. Caracterización del público objetivo para el eBook Interactivo.	104
Tabla 19. Contenido de enseñanza y objetivos de aprendizaje	106
Tabla 20. Selección herramienta de desarrollo	111
Tabla 21. Encuesta de contexto.	128
Tabla 22. Amenazas a la validez en una investigación cuantitativa descriptiva y estrategias de mitigación.	128
Tabla 23. Estrategia instruccional sesión 1.	130
Tabla 24. Estrategia instruccional sesión 2.	131
Tabla 25. Estrategia instruccional sesión 3.	131
Tabla 26. Estrategia instruccional sesión 4.	131
Tabla 27. Estrategia instruccional sesión 5.	132
Tabla 28. Estrategia instruccional sesión 6.	132
Tabla 29. Estadísticos descriptivos del pre-test y el pos-test.	133
Tabla 30. Resultados prueba de normalidad Kolmogórov-Smirnov.	134
Tabla 31. Resultados prueba de Wilcoxon.	134
Tabla 32. Promedio (redondeado) de respuestas correctas por módulo, en las pruebas pre-test y post-test.	139

Introducción

La compresión de datos es el proceso de transformar una cadena de datos de entrada (los datos originales) en otra cadena de datos (la salida o cadena comprimida), cuyo tamaño es menor al de la cadena de entrada, se interpreta como un proceso que elimina la complejidad innecesaria (redundancia) de la información y así maximiza la sencillez, conservando su poder descriptivo y no redundante (Salomon y Motta, 2010).

Con el auge del Internet y los flujos de información, los procesos o algoritmos de compresión de datos toman cada vez mayor importancia, pues las personas valoran el almacenar grandes cantidades de información y requieren que el acceso a este contenido sea de manera expedita, específicamente si es contenido alojado en la web. Estos dos factores son altamente optimizados al emplear procesos de compresión de datos (Salomon y Motta, 2010).

En la actualidad, un gran porcentaje de la información es comprimida con alguno de los formatos de compresión conocidos. Por ejemplo, la mayoría de las imágenes se comprimen con los formatos JPEG, GIF, etc. La televisión de alta definición utiliza el formato MPEG-2, y en general, todo tipo de información es previamente comprimida para su uso y almacenamiento.

“En el estudio de temáticas abstractas, como es el caso de la compresión de datos, los estudiantes encuentran dificultad en su aprendizaje debido a la ausencia de recursos educativos digitales interactivos (García, 2016). Este hecho se evidencia en los estudios realizados por Guerrero y Pérez (2008) y Hamada y Hassan (2017), en los que determinan que los estudiantes tienen dificultades a la hora de comprender los conceptos y aplicaciones de la teoría de la información y la compresión de datos. Existe una gran cantidad de contenidos digitales para la enseñanza y aprendizaje de los conceptos de la compresión de datos a diferentes niveles de profundidad, muchos de los cuales están disponibles principalmente en formatos de texto tales como PDF, y algunos pocos, están plasmados en videos ilustrativos que ofrecen de manera no formal contenidos, lo cual demuestra la ausencia de material pedagógico interactivo para la

enseñanza de la compresión de datos, siendo ésta una temática de gran importancia considerando los grandes volúmenes de datos que se manejan en la sociedad actual” (Cedeño & Vargas, 2021).

“La educación es uno de los pilares para la producción del conocimiento y la mejora en la calidad de vida de los individuos, pero en ocasiones esta puede verse truncada o limitada, debido a diferentes factores que afectan la correcta evolución del aprendizaje, por lo cual se hace necesario el uso de nuevas herramientas que permitan consolidar el adecuado curso de todo proceso educativo” (Cedeño & Vargas, 2019).

Actualmente, con el creciente avance tecnológico en materia de dispositivos digitales móviles tales como Smartphones, Tablets, Laptops, entre otros, es importante considerar el potencial de estos dispositivos en los procesos de enseñanza y el aprendizaje mediante libros electrónicos (Cedeño & Vargas, 2019). El término *eBook* (electronic book) o libro electrónico se utiliza para denominar los libros digitales o la versión electrónica de un libro físico, estos eBooks pueden ser visualizados en un sin número de dispositivos tales como PC, smartphone, PDAs, e-readers como el Kindle Amazon, entre otros dispositivos electrónicos de bolsillo (Mutalib, Noor, Hashim, & Mahari, 2012; Sepúlveda & Quiceno, 2015).

“Los eBooks ofrecen a los estudiantes y profesores una herramienta adicional para apoyar o mejorar el proceso de aprendizaje. El uso de libros electrónicos en lugar de libros de texto en las aulas de clases es una nueva tendencia gracias a sus ventajas, que van desde la disminución del peso con respecto a las versiones en físico de los libros hasta la generación de un proceso de aprendizaje divertido para los estudiantes, debido a las características atractivas del eBook como lo son las funciones amigables para el usuario, gráficos novedosos, control del tamaño de texto, etc. (Mutalib et al., 2012). Recientemente, se han adelantado investigaciones que evalúan la posibilidad de utilizar formatos comunes de eBooks para crear *eBooks interactivos* o libros gamificados, similares al formato iBook de Apple, para la enseñanza y el aprendizaje, creando un modelo de libro dinámico que puede funcionar como un juego educativo que presenta una muy buena usabilidad y un prometedor potencial pedagógico (Figueiredo, Bidarra, & Natalio, 2014)” (Cedeño & Vargas, 2019). “Los eBooks interactivos, mediante la integración de elementos multimedia, permiten nuevas prácticas de aprendizaje de mayor interacción (García, 2016), haciendo uso de las tecnologías de la información y la comunicación presentes en distintos

dispositivos tales como un teléfono móvil, un computador o una tableta (Wu et al., 2013)” (Cedeño & Vargas, 2021).

En la revisión sistemática de literatura efectuada, no se encontraron eBooks interactivos que apoyen el proceso de enseñanza-aprendizaje en el área de la comprensión de datos (Cedeño & Vargas, 2021). Por lo tanto, en el presente trabajo se realiza el desarrollo de un eBook interactivo multiplataforma para la enseñanza-aprendizaje de la comprensión de datos, que permita un aprendizaje mediante actividades dinámicas con posibilidades de validación y autochequeo de lo aprendido y con un alto grado de interactividad (Cedeño & Vargas, 2021).

Por lo tanto, como objetivo general del presente trabajo se plantea desarrollar un eBook interactivo multiplataforma para la enseñanza-aprendizaje de la comprensión de datos, cumpliendo con los siguientes objetivos específicos:

- Caracterizar los eBooks mediante una revisión sistemática de literatura seleccionada.
- Realizar el análisis y diseño del eBook interactivo multiplataforma.
- Desarrollar el eBook interactivo multiplataforma.
- Implementar el eBook interactivo multiplataforma y evaluar su usabilidad y su efectividad en la consecución de los objetivos de aprendizaje.

Para cumplir este objetivo se hace uso de una metodología inspirada en el modelo de diseño instruccional ADDIE (Allen, 2006), cuyo nombre es un acrónimo de los pasos clave que lo conforman (Análisis, Diseño, Desarrollo, Implementación y Evaluación), y consta de las siguientes fases:

- Análisis: “Inicialmente es necesario caracterizar la población a la cual estará dirigido el eBook, definir los objetivos de aprendizaje, definir el contenido de la enseñanza y determinar la plataforma más apropiada para el desarrollo del eBook” (Cedeño & Vargas, 2019).
- Diseño: Teniendo en cuenta el análisis realizado, se diseña la interfaz gráfica, la navegabilidad y los contenidos multimedia interactivos del eBook interactivo.
- Desarrollo y Pruebas: Desarrollo del eBook interactivo multiplataforma con base en los resultados de las etapas anteriores. Realización de pruebas periódicas al prototipo obtenido con un grupo focal.

- Implementación y Evaluación. Implementación del eBook y evaluación de su efectividad con respecto a los resultados de la acción formativa y de percepción en cuanto a la experiencia de aprendizaje.

Considerando lo anterior, este documento tiene la siguiente estructura: en el Capítulo 1 se exponen los contenidos fundamentales de la comprensión de datos, que conforman los aspectos disciplinares del proyecto. En el Capítulo 2 se definen los aspectos didácticos del proyecto, mediante la caracterización de los eBooks interactivos y la descripción de los beneficios de su implementación en ambientes educativos. En el Capítulo 3 se desarrolla una revisión de trabajos previos, para determinar los diferentes aportes realizados hasta el momento en la enseñanza y aprendizaje de la comprensión de datos usando de eBooks interactivos. El Capítulo 4 describe la aplicación de las tres primeras fases de la metodología seleccionada para el presente proyecto, a saber: análisis, diseño y desarrollo del eBook interactivo. El Capítulo 5 describe la aplicación de las dos últimas fases de la metodología, para lograr la implementación y evaluación del eBook interactivo. Por último, en el Capítulo 6 se presentan las conclusiones y las recomendaciones generadas por el desarrollo de este proyecto.

1. Aspectos disciplinares: compresión de datos

En el presente capítulo se exponen los contenidos relevantes sobre compresión de datos que constituyen los aspectos disciplinares del proyecto, el cuerpo del contenido educativo que será apoyado por el eBook interactivo. Por lo tanto, en la Sección 1.1 se definen los conceptos básicos de la compresión de datos; en la Sección 1.2 se proporciona una breve introducción a las temáticas de la teoría de la información fundamentales para la compresión de datos; en la Sección 1.3 se expone la codificación de la fuente; en la Sección 1.4 se describen los métodos de compresión estadísticos; en la Sección 1.5 se estudia la compresión basada en contexto; en la Sección 1.6 se presentan los métodos de diccionario; finalmente en la Sección 1.7 se exponen las técnicas de compresión con pérdida.

1.1 Conceptos básicos de la compresión de datos

¿Qué es la compresión?

La compresión de datos es el proceso de transformar una cadena de datos de entrada (los datos originales) en otra cadena de datos (la salida o cadena comprimida), cuyo tamaño es menor al de la cadena de entrada .

La compresión de datos se interpreta como un proceso que elimina la complejidad innecesaria (redundancia) de la información y así maximiza la sencillez, conservando su poder descriptivo y no redundante (Salomon y Motta, 2010).



La compresión de datos reduce la cantidad de espacio necesaria para almacenar datos o reduce la cantidad de tiempo necesaria para transmitir datos (Suman, Anjali, Sonal, 2015).

El área de la compresión de datos también se conoce como *codificación de la fuente*, dado que se supone que la entrada de símbolos es emitida por una cierta fuente de información, y debe codificarse antes de ser enviada a su destino.

Aunque existen gran cantidad de métodos de compresión, apropiados para distintos tipos de datos, todos se basan en el principio de eliminar la redundancia de los datos originales. “Todos los datos no aleatorios tienen alguna estructura que se puede aprovechar para lograr una representación más pequeña de los datos. Las redundancias en los datos dependen del tipo de datos (texto, imagen, sonido, etc.), por lo que se tienen que desarrollar métodos de compresión específicos para cada tipo de datos; no existe un algoritmo eficiente y universal para la compresión de datos” (Salomon y Motta, 2010).

1.1.1 Técnicas de compresión de datos

Cuando se habla de técnicas de compresión se hace referencia a dos algoritmos:

- Algoritmo de compresión: toma una entrada x y genera una representación x_c que requiere menos bits.
- Algoritmo de reconstrucción (descompresión): opera sobre la representación comprimida x_c para generar la reconstrucción y (ver Figura 1).

Según la convención establecida, cuando se habla de algoritmo de compresión, se hace referencia a los dos algoritmos: compresión y reconstrucción (descompresión).

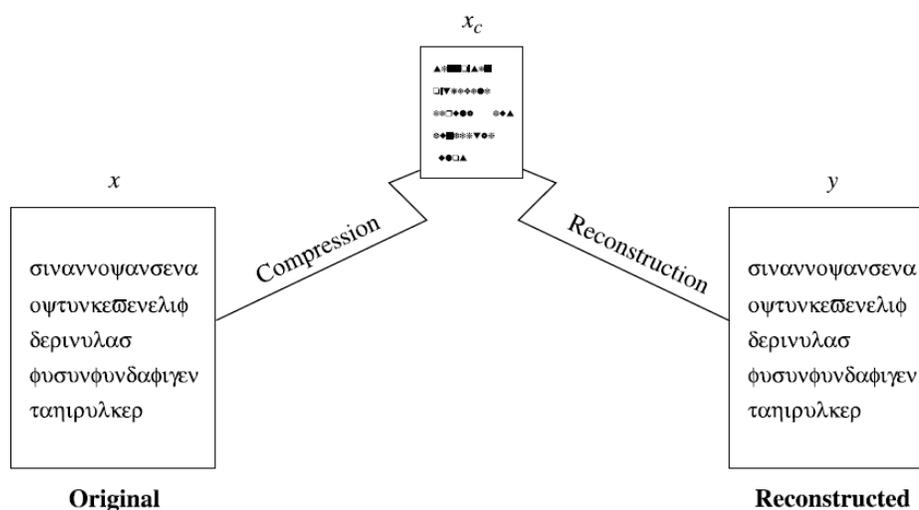


Figura 1. Compresión y reconstrucción (descompresión) (Sayood, 2017).

Según los requerimientos de la descompresión, los métodos de compresión de datos se pueden dividir en dos clases:

- **Métodos de compresión sin pérdida:** no involucran pérdidas de información (y es idéntico a x), los datos originales se pueden recuperar exactamente a partir de los datos comprimidos.

Estos métodos son generalmente usados para aplicaciones que no toleran diferencias entre los datos originales y los reconstruidos. La compresión de textos es un área de aplicación, ya que es importante que la reconstrucción sea idéntica al texto original, debido a que pequeñas diferencias pueden conducir a sentencias con significados diferentes.

- **Métodos de compresión con pérdida:** involucran alguna pérdida de información (y es diferente de x), y los datos que han sido comprimidos generalmente no pueden ser recuperados o reconstruidos exactamente.

Estos métodos tienen sentido especialmente en la compresión de imágenes, video o sonido, ya que si la pérdida de datos es pequeña, en general no será perceptible (Sayood, 2017).

1.1.2 La compresión de datos y los sistemas de comunicación

Una fuente de información genera una señal que con frecuencia es analógica, en la figura se ha representado como una función del tiempo $x(t)$. Si la información no está en forma discreta, se discretiza para generar una secuencia de valores $\{x_n\}$. Esta secuencia discreta es la entrada al algoritmo de compresión (codificador de fuente), que intenta reducir el número promedio de bits por muestra usado para codificar la secuencia. Para proteger contra errores, la salida del codificador de fuente se codifica usando un codificador de canal que introduce redundancia de forma controlada. La salida del decodificador de fuente $\{y_n\}$ es suministrada al usuario. La meta del proceso de compresión es usar tan pocos bits como sea posible para codificar la secuencia portadora de información $\{x_n\}$ mientras que se mantiene la secuencia reconstruida $\{y_n\}$ como una aproximación cercana a $\{x_n\}$.

La Figura 2 muestra un diagrama de bloques simplificado de un sistema de comunicación.

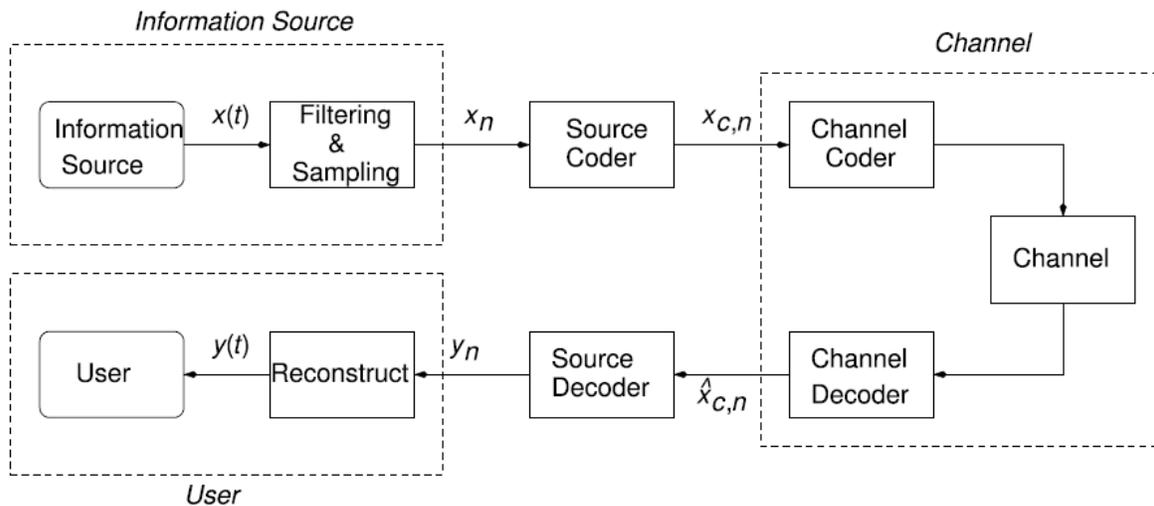


Figura 2. Diagrama de bloques simplificado de un sistema de comunicación (Sayood, 2017).

1.1.3 Medidas de rendimiento de la compresión

El rendimiento de un algoritmo de compresión puede medirse según diversos aspectos, tales como la complejidad del algoritmo, la memoria requerida para su implementación, el tiempo de ejecución del algoritmo, la cantidad de compresión lograda y qué tanto se asemeja la reconstrucción a la información original.

Se estudiarán los últimos dos aspectos mencionados, debido a su gran popularidad y a que son independientes del hardware/software en el que el algoritmo se ejecute.

Factor de compresión

Consiste en determinar el cociente entre el número de bits requeridos para representar los datos antes y después de la compresión (Salomon y Motta, 2010):

$$FC = \frac{\text{Tamaño original}}{\text{Tamaño después de comprimir}} : 1 \quad (1-1)$$

Ejemplo 1

Se tiene una imagen de 256×256 píxeles que ocupa 65,536 bytes para su almacenamiento, y una vez comprimida requiere solo 16,384 bytes. Se dirá entonces que el factor de compresión es de 4:1 (Sayood, 2017).

Razón de compresión

Corresponde a la inversa del factor de compresión:

$$RC = \frac{\text{Tamaño después de comprimir}}{\text{Tamaño original}} \quad (1-2)$$



Un valor menor que 1 significa compresión siendo esta mayor mientras más cercano esté el valor a 0. Valores mayores a 1 indican compresión negativa, es decir una expansión de la información original (Salomon y Motta, 2010).

La razón de compresión también puede ser expresada en términos de la reducción por compresión, como un porcentaje del tamaño de los datos originales:

$$\text{Reducción Compresión} = (1 - RC) * 100\% \quad (1-3)$$

Tasa de compresión

Otra manera de describir el desempeño de la compresión es proporcionar el número promedio de bits requeridos para representar una muestra, lo que se conoce generalmente como la *tasa* de la compresión (Sayood, 2017).

Ejemplo 2

Retomando el ejemplo anterior:

$$RC = \frac{16384}{65536} = 0.25 ,$$

indicando que el algoritmo de compresión reduce la imagen a un 25% del tamaño original, lo que es equivalente a que se redujo en un 75% el tamaño de la misma según el cálculo de la reducción por compresión:

$$\text{Reducción Compresión} = (1 - 0.25) * 100\% = 75\%$$

Para el caso de la imagen comprimida descrita, el número promedio de bits por pixel en la representación comprimida es 2. Por lo tanto, se dice que la tasa es de **2 bits por pixel** (Sayood, 2017).

Distorsión

En la compresión con pérdida, la información reconstruida difiere en cierta medida de los datos originales, por lo que para determinar la eficiencia de la compresión se debe cuantificar esta diferencia, la cual es denominada *distorsión*.



Las técnicas de compresión con pérdida son utilizadas principalmente para comprimir datos provenientes de señales analógicas, tales como voz y video, para las cuales los términos de calidad de la compresión son establecidos con base en las limitaciones propias del cuerpo humano, por ejemplo, gama de colores, espectro audible, etc. El modelado matemático de las respuestas de los sistemas humanos es de gran complejidad, por lo que se utilizan medidas de distorsión aproximadas para determinar la calidad de la información reconstruida.

Considérese $\{x_n\}$ como la secuencia de datos de entrada (la salida de la fuente) y $\{y_n\}$ como la secuencia reconstruida.

►) Para resumir la información de la secuencia diferencia $\{x_n - y_n\}$ y obtener un único valor representativo, se calcula el *error cuadrático medio* σ^2 (Salomon y Motta, 2010 & Sayood, 2017):

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - y_n)^2 \quad (1-4)$$

►) Para algunas aplicaciones específicas, como la compresión de imágenes, la distorsión no es perceptible mientras se encuentre bajo un umbral preestablecido, por lo que se debe calcular el *valor máximo de error*:

$$d_{\infty} = \max_n |x_n - y_n| \quad (1-5)$$

Ejercicio 1

Se tiene un documento que ocupa 131,072 bytes para su almacenamiento, y una vez comprimido requiere solo 16,384 bytes.

- a) Calcule el factor de compresión FC.
- b) Calcule la razón de compresión RC.
- c) Calcule la reducción por compresión.

1.2 Breve introducción a la teoría de la información

1.2.1 Fuentes de información

Una fuente práctica en un sistema de comunicación es un dispositivo que produce mensajes y puede ser análoga o discreta. Específicamente se estudiarán las fuentes discretas, considerando que las fuentes análogas se pueden transformar en discretas mediante el uso de técnicas de muestreo y cuantización.

Una fuente de información discreta tiene solamente un conjunto finito de símbolos como posibles salidas. El conjunto finito de símbolos de la fuente es denominado el *alfabeto fuente*, y los elementos del conjunto se llaman *símbolos* o *letras*.

Adicionalmente las fuentes de información se pueden clasificar como fuentes con memoria o sin memoria. Una fuente *con memoria* es aquella en la que el símbolo actual depende de los símbolos previos. Una fuente *sin memoria* es aquella en la que cada símbolo producido es independiente de los símbolos previos (Hwei, 2003).

Para el estudio de los procesos de compresión de datos se usarán principalmente las fuentes discretas sin memoria, identificadas por sus siglas en inglés DMS (Discrete Memoryless Source).

1.2.2 Contenido de información de un símbolo

Mensajes que contienen conocimiento de alta probabilidad de ocurrencia transmiten relativamente poca información, y mensajes que contienen conocimiento de baja probabilidad de ocurrencia transmiten relativamente mucha información.

Así, una medida matemática de la información debería ser función de la probabilidad de la salida de la fuente y debería satisfacer los siguientes axiomas:

1. La información debe ser proporcional a la incertidumbre de la salida.
2. La información contenida en salidas independientes se debe sumar.

Considere una DMS denotada por X , con alfabeto $A = \{x_1, x_2, \dots, x_m\}$ y siendo $P(x_i)$ la probabilidad de ocurrencia del símbolo x_i . El *contenido de información* o *información propia* del símbolo x_i , denotado por $I(x_i)$, se define como (Hwei, 2003):

$$I(x_i) = -\log_b(P(x_i)) \quad (1-6)$$

Si la base del logaritmo es $b = 2$, la unidad para la $I(x_i)$ es el *bit* (*binary unit*). Aquí la unidad bit, de forma abreviada “b”, es una medida del contenido de información y no se debe confundir con el término *bit* que significa “dígito binario”.

El contenido de información $I(x_i)$ satisface las siguientes propiedades:

$$\begin{aligned} I(x_i) &= 0 \quad \text{para } P(x_i) = 1 \\ I(x_i) &\geq 0 \\ I(x_i) &> I(x_j) \quad \text{si } P(x_i) < P(x_j) \\ I(x_i x_j) &= I(x_i) + I(x_j) \quad \text{si } x_i \text{ y } x_j \text{ son independientes} \end{aligned}$$

Ejemplo 3

Para la frase “Hola_Mundo”, encuentre la información propia de cada caracter.

Dada la DMS X con alfabeto:

$$A = \{H, o, l, a, _, M, u, n, d\},$$

se obtiene la distribución de probabilidad con base en la frecuencia de cada caracter:

$$P(x_i) = \{0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$$

La probabilidad $P(x_i)$ para cada caracter corresponde a su frecuencia sobre la cantidad total de caracteres.

$$\begin{aligned} I(H) &= -\log_2(0.1) = 3.32 & I(a) &= -\log_2(0.1) = 3.32 & I(u) &= -\log_2(0.1) = 3.32 \\ I(o) &= -\log_2(0.2) = 2.32 & I(_) &= -\log_2(0.1) = 3.32 & I(n) &= -\log_2(0.1) = 3.32 \\ I(l) &= -\log_2(0.1) = 3.32 & I(M) &= -\log_2(0.1) = 3.32 & I(d) &= -\log_2(0.1) = 3.32 \end{aligned}$$

Como se aprecia, la información propia del mensaje "o" es menor a la del resto de mensajes debido a su mayor probabilidad de ocurrencia.

1.2.3 Información promedio o entropía

En un sistema de comunicación práctico, se transmiten usualmente largas secuencias de símbolos desde una fuente de información. Por lo tanto, es de más interés conocer la información promedio que una fuente produce, que el contenido de información de cada símbolo.

¿Qué es la entropía?

El valor promedio de $I(x_i)$ sobre el alfabeto de la fuente X con m símbolos diferentes está dado por (Hwei, 2003):

$$\begin{aligned} H(X) &= E[I(x_i)] = \sum_{i=1}^m P(x_i) I(x_i) \\ &= - \sum_{i=1}^m P(x_i) \log_2(P(x_i)) \text{ b/símbolo} \end{aligned} \quad (1-7)$$

La cantidad $H(X)$ se conoce como la *entropía* de la fuente X . Es una medida del contenido de información promedio por símbolo de la fuente, o de la cantidad de incertidumbre promedio de la fuente X que se resuelve usando el alfabeto.

La entropía de la fuente $H(X)$ satisface la siguiente relación:

$$0 \leq H(X) \leq \log_2(m)$$

- La cota inferior corresponde a la ausencia de incertidumbre, que ocurre cuando un símbolo tiene probabilidad $P(x_i) = 1$ mientras que $P(x_j) = 0$ para $j \neq i$, de modo que X emite el mismo símbolo x_i todo el tiempo.
- La cota superior corresponde a la máxima incertidumbre, que ocurre cuando $P(x_i) = \frac{1}{m}$ para todo i , es decir cuando todos los símbolos tienen la misma probabilidad de ser emitidos por X (Hwei, 2003).

Ejemplo 4

Para una fuente binaria X que genera los símbolos 1 y 0 con igual probabilidad, la entropía de la fuente $H(X)$ es (Hwei, 2003):

$$H(X) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1 \text{ b/símbolo}$$

Ejemplo 5

Calcular la información promedio para el siguiente texto:

“por supuesto que las maquinas no pueden pensar como la gente lo hace”

Símbolos x_i :

x_i	_	a	c	d	e	g	h	i	l	m	n	o	p	q	r	s	t	u
$P(x_i)$	$\frac{3}{17}$	$\frac{3}{34}$	$\frac{1}{34}$	$\frac{1}{68}$	$\frac{2}{17}$	$\frac{1}{68}$	$\frac{1}{68}$	$\frac{1}{68}$	$\frac{3}{68}$	$\frac{1}{34}$	$\frac{5}{68}$	$\frac{3}{34}$	$\frac{1}{17}$	$\frac{1}{34}$	$\frac{1}{34}$	$\frac{5}{68}$	$\frac{1}{34}$	$\frac{5}{68}$

Aplicando la ecuación para la entropía:

$$H(X) = -\sum_{i=1}^m P(x_i) \log_2(P(x_i))$$

x_i	_	a	c	d	e	g	h	i	l	m	n	o	p	q	r	s	t	u
$P(x_i)I(x_i)$	0.442	0.309	0.15	0.09	0.363	0.09	0.09	0.09	0.199	0.15	0.277	0.309	0.24	0.15	0.15	0.277	0.15	0.277

$$H(X) = 3.803 \text{ b/símbolo}$$

Ejercicio 2

- Calcule la entropía $H(X)$ del idioma español, considerando que está conformado por 27 caracteres equiprobables.
- Para la DMS X con alfabeto $A = \{x_1, x_2, x_3, x_4\}$ con probabilidades $P(x_i) = \{0.4, 0.3, 0.2, 0.1\}$, encuentre la cantidad de información contenida en los mensajes $M_1 = x_1x_2x_1x_3$ y $M_2 = x_4x_3x_3x_2$ (Hwei, 2003).
- Si la velocidad a la que la fuente X emite símbolos es r (símbolos/s), la *velocidad de información* R de la fuente es $R = rH(X)$ (b/s). Calcule la R de una fuente telegráfica que tiene 2 símbolos: “.” y “-”. La duración del “.” es 0.2 s y la de la “-” es 3 veces la duración del “.”. La probabilidad de ocurrencia del “.” es 2 veces la de la “-”, y el tiempo entre símbolos es 0.2 s (Hwei, 2003).
- Calcule el valor de la entropía para el mensaje “Hola_Mundo”.

1.3 Codificación de fuente

Se llama *codificación de fuente* a la conversión de la salida de una DMS en una secuencia de símbolos binarios o palabra de código binaria. Esta conversión es realizada por un dispositivo llamado *codificador de fuente* (Figura 3).

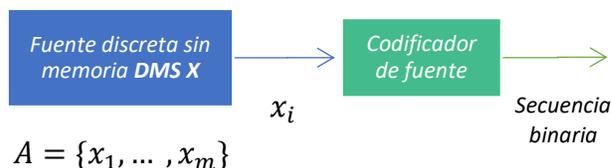


Figura 3. Codificación de fuente. Tomada de (Hwei, 2003).



Un objetivo de la codificación de fuente es minimizar la tasa de bits promedio requerida para la representación de la fuente, reduciendo la redundancia de la información de la fuente (compresión).

1.3.1 Longitud promedio de código

Considere una DMS X con entropía $H(X)$ y alfabeto $A = \{x_1, x_2, \dots, x_m\}$ con probabilidades de ocurrencia correspondientes $P(x_i)$ ($i = 1, \dots, m$). La palabra de código binario asignada al símbolo x_i por el codificador tiene una longitud de n_i bits. La longitud promedio L de código, por símbolo de la fuente está dada por (Hwei, 2003):

$$L = \sum_{i=1}^m P(x_i) n_i \quad \text{b/símbolo} \quad (1-8)$$

La longitud L representa el número promedio de b/símbolo de la fuente usado en el proceso de codificación de fuente.

La *eficiencia* del código, η , se define como

$$\eta = \frac{L_{min}}{L}, \quad (1-9)$$

donde L_{min} es el mínimo valor posible de L . Cuando η se aproxima a 1 el código se denomina *eficiente* (Hwei, 2003).

1.3.2 Teorema de codificación de fuente

Para una DMS X con entropía $H(X)$, la longitud promedio de código, L , está acotada como:

$$L \geq H(X), \quad (1-10)$$

y además, L se puede hacer tan cercana a $H(X)$ como se desee, escogiendo apropiadamente el código. Entonces, $L_{min} = H(X)$, y la eficiencia del código se puede reescribir como:

$$\eta = \frac{H(X)}{L}. \quad (1-11)$$

Según la ecuación (1-10), L no podrá ser inferior a la entropía de la fuente $H(X)$, indicando que el valor de la entropía establece la compresión máxima alcanzable. Comparando las ecuaciones (1-7) y (1-8), se deduce que esta compresión se alcanzaría si $n_i = -\log_2(P(x_i))$.

1.3.3 Clasificación de los códigos

Considere la Tabla 1 donde se ha codificado una fuente X de tamaño 4, con códigos binarios con los símbolos 0 y 1.

Tabla 1. Códigos binarios (Hwei, 2003).

x_i	Código 1	Código 2	Código 3	Código 4	Código 5	Código 6
x_1	00	00	0	0	0	1
x_2	01	01	1	10	01	01
x_3	00	10	00	110	011	001
x_4	11	11	11	111	0111	0001

- **Códigos de longitud fija:** un código de *longitud fija* es aquel en el que la longitud de las palabras del código es fija. Los códigos 1 y 2 de la Tabla 1 son de longitud fija igual a 2.
- **Códigos de longitud variable:** un código de *longitud variable* es aquel en el que la longitud de las palabras de código no es fija. Todos los códigos de la Tabla 1 excepto los códigos 1 y 2 son códigos de longitud variable.
- **Códigos distintos:** un código es *distinto* si cada palabra del código es distinguible de las demás palabras de este código. Todos los códigos de la Tabla 1 excepto el código 1, son códigos distintos (obsérvese las palabras de código para x_1 y x_3).
- **Códigos libres de prefijo:** un código *libre de prefijo* es aquel en el que ninguna palabra del código se puede formar adicionando símbolos del código a otra palabra del código, es decir, ninguna palabra del código es prefijo de otra. Los códigos 2, 4 y 6 de la Tabla 1 corresponden a códigos libres de prefijo.
- **Códigos unívocamente decodificables:** un código es *unívocamente decodificable*, si la secuencia original de la fuente se puede reconstruir sin ambigüedades a partir de la secuencia binaria codificada. Una condición suficiente para que un código sea unívocamente decodificable es que sea un código libre de prefijo.

El código 3 de la Tabla 1 no es unívocamente decodificable, por ejemplo, la secuencia binaria “1001” puede corresponder a la secuencia de la fuente $x_2x_3x_2$ o $x_2x_1x_1x_2$. El código 5 de la Tabla 1 no es libre de prefijo, y aun así es unívocamente decodificable ya que el bit “0” indica el comienzo de todas las palabras del código.

- **Códigos instantáneos:** un código unívocamente decodificable se denomina código *instantáneo* si se reconoce el final de cada palabra del código sin examinar símbolos subsiguientes del mismo. Los códigos instantáneos tienen la propiedad de ser libres de prefijo.
- **Códigos óptimos:** un código se denomina *óptimo* si es instantáneo y tiene longitud promedio mínima L para una fuente determinada con una distribución de probabilidad dada para los símbolos o mensajes de la fuente.

La longitud promedio de un código óptimo satisface (Blelloch, 2013):

$$L \leq H(X) + 1 \quad (1-12)$$

Ejemplo 6

Considere la DMS X con alfabeto $A = \{a, b, c, d\}$ con distribución de probabilidad $P(x_i) = \{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$. Para cada uno de los códigos que se muestran, calcule la longitud promedio de código.

a) Código de longitud constante:

x_i	Código	n_i
a	00	2
b	01	2
c	10	2
d	11	2

$$L = \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{2} \cdot 2$$

$$L = 2 \text{ b/símbolo}$$

b) Código de longitud variable:

x_i	Código	n_i
a	000	3
b	001	3
c	01	2
d	1	1

$$L = \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 + \frac{1}{4} \cdot 2 + \frac{1}{2} \cdot 1$$

$$L = 1.75 \text{ b/símbolo}$$

Se observa que los dos códigos de este ejemplo satisfacen la ecuación (1-12) (la entropía de la fuente es $H(X)=1.75$) y por tanto son códigos óptimos, sin embargo, se debe preferir el de longitud variable ya que su L es menor y por tanto su eficiencia η será mayor.

Ejercicio 3

Considere los siguientes códigos:

x_i	Código	$P(x_i)$
a	10	1/8
b	01	1/8
c	00	5/8
d	010	1/8

x_i	Código	$P(x_i)$
a	10	1/16
b	01	1/8
c	001	3/8
d	101	7/16

x_i	Código	$P(x_i)$
a	1	1/5
b	01	2/5
c	101	3/10
d	011	1/10

x_i	Código	$P(x_i)$
a	000	3/10
b	001	1/10
c	01	2/5
d	1	1/5

x_i	Código	$P(x_i)$
a	00	3/8
b	01	2/5
c	10	1/8
d	11	1/10

x_i	Código	$P(x_i)$
a	01	1/2
b	001	1/5
c	0001	1/5
d	0000	1/10

- Identifique si cada uno de los códigos corresponde o no a un código libre de prefijo.
- Calcule la longitud promedio de código, L , de los códigos.

1.4 Métodos Estadísticos

Los métodos estadísticos usan códigos de tamaño variable, asignando a los mensajes o símbolos de mayor probabilidad palabras de código de menor longitud, y a los mensajes o símbolos menos frecuentes palabras de mayor longitud, con lo que se logra reducir L (la longitud promedio del código) (Salomon y Motta, 2010). Por lo tanto, estos métodos requieren el conocimiento de la distribución de probabilidad de los mensajes o símbolos de la fuente.

1.4.1 Codificación Huffman

Esta técnica fue desarrollada por David Huffman como parte de una tarea del primer curso sobre teoría de información, impartido por Robert Fano en el MIT. Los códigos generados empleando esta técnica se conocen como *códigos de Huffman*. Estos códigos son óptimos para una distribución de probabilidades dada (Sayood, 2017).

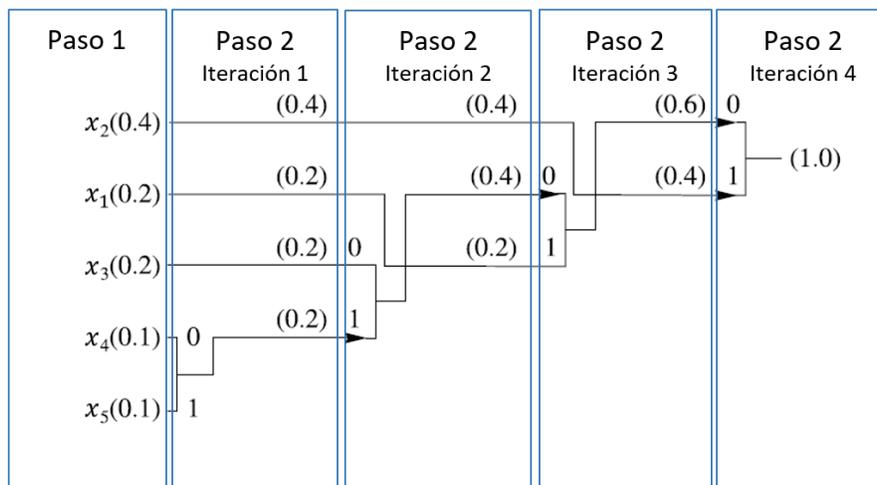
Algoritmo de codificación Huffman

Una manera de construir un código Huffman es usar el hecho de que este se puede representar como un árbol binario, en el cual los nodos exteriores u hojas corresponden a los símbolos. La generación del árbol se realiza con el siguiente procedimiento (Sayood, 2017; Hwei, 2003; Brelloch, 2013):

1. Listar los símbolos de la fuente en orden descendente de probabilidad. Asignar cada símbolo a un nodo hoja.
2. Repetir hasta que solamente quede un único nodo (nodo raíz):
 - Conectar las hojas correspondientes a los dos símbolos con las más bajas probabilidades en un único nodo, y tratar este nodo como un símbolo de un alfabeto reducido. La probabilidad de este símbolo es la suma de las probabilidades de sus descendientes.
 - Reordenar los nodos correspondientes al alfabeto reducido.
3. Para obtener el código de cualquier símbolo, recorrer el árbol desde la raíz hasta el nodo hoja correspondiente al símbolo, asignando un “0” a la palabra de código cada vez que el recorrido conduzca por una rama superior y un “1” cada vez que el recorrido conduzca por una rama inferior.

Ejemplo 7

Diseñe un código de Huffman para la fuente X con alfabeto $A = \{x_1, x_2, x_3, x_4, x_5\}$ con $P(x_1) = P(x_3) = 0.2$, $P(x_2) = 0.4$ y $P(x_4) = P(x_5) = 0.1$ (Sayood, 2017):



Se obtiene el código a partir del árbol de codificación creado:

Tabla 2. Código Huffman generado.

x_i	Código	$P(x_i)$	n_i	$P(x_i)n_i$
x_1	01	0.2	2	0.4
x_2	1	0.4	1	0.4
x_3	000	0.2	3	0.6
x_4	0010	0.1	4	0.4
x_5	0011	0.1	4	0.4

Se obtiene $L = 2.2$ **b/símbolo**, confirmando que el código Huffman generado es óptimo ya que satisface la expresión (1-12):

$$2.2 \leq 2.12 + 1 .$$

Algoritmo de decodificación Huffman

El decodificador debe contar con el árbol de Huffman para el alfabeto, solo así puede leer y decodificar las secuencias de símbolos recibidas. El algoritmo para decodificar es el siguiente (Salomon y Motta, 2010):

1. Comenzar en el nodo raíz y leer el primer bit de los datos comprimidos.
2. Si el valor leído es “0”, dirigirse hacia la rama asociada al “0” del árbol, y si es “1”, continuar por la otra rama.
3. Leer el bit siguiente y desplazarse por otra rama hacia las hojas del árbol.
4. Cuando el decodificador llega a una hoja, se encuentra el símbolo sin comprimir.
5. El proceso comienza de nuevo en el nodo raíz con el bit siguiente.

Ejemplo 8

Considere el árbol de codificación Huffman del ejemplo 7. Decodifique la secuencia de símbolos 0100000101101:

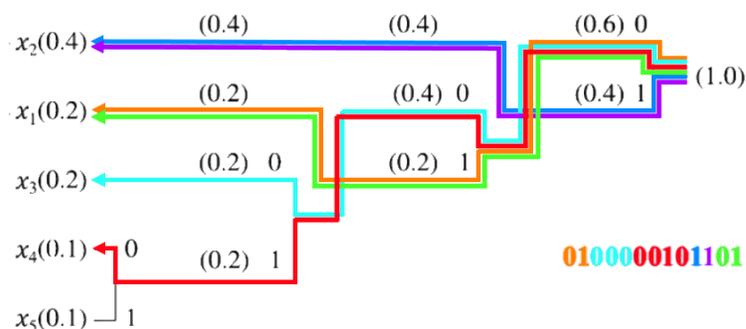


Figura 4. Decodificación Huffman

Siguiendo el algoritmo de decodificación se obtiene la secuencia reconstruida: $x_1x_3x_4x_2x_2x_1$.

Aplicaciones de la codificación Huffman

🔊 Otra clase de datos muy apropiados para compresión son los datos de audio calidad de CD, la señal de audio para cada canal estéreo se muestrea a 44.1KHz, y cada muestra se representa mediante 16 bits. Por lo tanto, la cantidad de datos almacenados en un CD es enorme y la compresión es muy útil en este caso. En la Tabla 3 se muestra para diferentes materiales de audio, el tamaño del archivo, la entropía, el tamaño estimado de archivo comprimido si se usa un codificador Huffman, y el factor de compresión resultante (Sayood, 2017).

Tabla 3. Compresión de audio con codificación Huffman (Sayood, 2017).

File Name	Original File Size (bytes)	Entropy (bits)	Estimated Compressed File Size (bytes)	Compression Factor
Mozart	939,862	12.8	725,420	1.30
Cohn	402,442	13.8	349,300	1.15
Mir	884,020	13.7	759,540	1.16

1.4.2 Códigos aritméticos

El método de Huffman es simple y eficiente para símbolos individuales, sin embargo, se puede demostrar que el único caso en el que produce códigos ideales de longitud variable (cuya longitud promedio de código es igual a la entropía) es cuando los símbolos tienen probabilidades que son potencias negativas de 2.

🔊) Lo anterior debido a que el método de Huffman asigna una palabra de código con un número entero de bits a cada símbolo del alfabeto. La teoría de la información muestra que, por ejemplo, a un símbolo con una probabilidad de 0.4 podría idealmente asignársele un código de 1.32 bits ya que $-\log_2 0.4 \cong 1.32$ (ver sección 1.3.2). Sin embargo, el método de Huffman normalmente le asigna una palabra de código de 1 o 2 bits.

La codificación aritmética evita este problema mediante la asignación de un código, conocido como etiqueta o *tag*, a una secuencia de símbolos contenida en un archivo de entrada. El método comienza con el intervalo $[0, 1)$, lee el archivo origen símbolo a símbolo y utiliza la probabilidad de cada símbolo para reducir el intervalo. Especificar un intervalo más estrecho requiere más bits, por lo que el código construido por el algoritmo crece continuamente.

Para lograr compresión, el algoritmo está diseñado de modo que un símbolo de alta probabilidad reduce el intervalo menos que uno de baja probabilidad, de tal manera que los símbolos con mayor probabilidad contribuyen con menos bits al código. La salida de la codificación aritmética es un número perteneciente al intervalo final, que constituye la etiqueta o *tag* generado (Salomon y Motta, 2010).

Algoritmo de codificación aritmética

A continuación, se encuentra el procedimiento para la codificación aritmética (Sayood, 2017; Salomon y Motta, 2010):

1. Obtener la función de distribución acumulada (FDA) para los símbolos:

$$F_X(x_i) \text{ para cada símbolo } x_i.$$

2. Definir el *intervalo actual* como $[0, 1)$:

$$low = 0, \quad high = 1$$

3. Repetir los siguientes pasos para cada símbolo x_k de la secuencia de entrada:

- Seleccionar el subintervalo para x_k :

$$newlow = low + (high - low) * F_X(x_k - 1)$$

$$newhigh = low + (high - low) * F_X(x_k)$$

- Definir el subintervalo como el nuevo *intervalo actual*:

$$low = newlow, \quad high = newhigh$$

4. Generar como salida cualquier número dentro del *intervalo actual* final, que corresponde a la etiqueta o *tag*.

Ejemplo 9

A continuación, un ejemplo para la determinación del *tag* que codifique la secuencia CASA. Se establece la longitud de la secuencia en 4 símbolos.

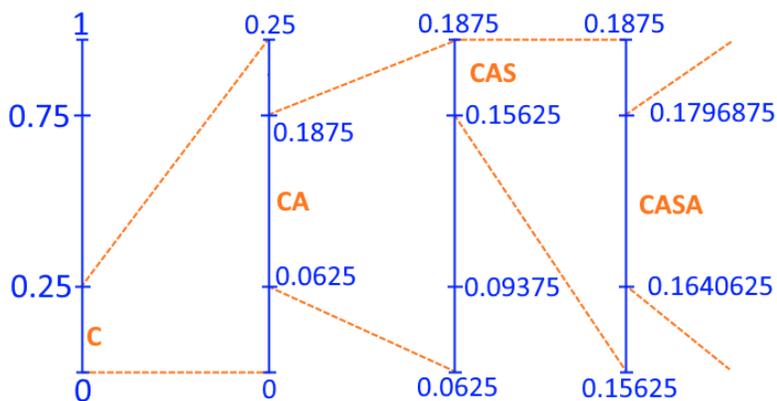
Se obtienen las probabilidades de cada símbolo x_i y el valor de su respectiva FDA:

x_i	$P(x_i)$	$F_X(x_i)$
C	0.25	0.25
A	0.5	0.75
S	0.25	1



Partiendo de $low = 0$ y $high = 1$, se procede a encontrar los sub-intervalos para cada uno de los símbolos de la secuencia:

Símbolo	Cálculo <i>low</i> y <i>high</i>	
C	<i>low</i>	$0+(1-0)\cdot 0= 0$
	<i>high</i>	$0+(1-0)\cdot 0.25= 0.25$
A	<i>low</i>	$0+(0.25-0)\cdot 0.25= 0.0625$
	<i>high</i>	$0+(0.25-0)\cdot 0.75= 0.1875$
S	<i>low</i>	$0.0625+(0.1875-0.0625)\cdot 0.75= 0.15625$
	<i>high</i>	$0.0625+(0.1875-0.0625)\cdot 1= 0.1875$
A	<i>low</i>	$0.15625+(0.1875-0.15625)\cdot 0.25= 0.1640625$
	<i>high</i>	$0.15625+(0.1875-0.15625)\cdot 0.75= 0.1796875$



Como se observa, el intervalo obtenido para una secuencia particular es disjunto de todos los demás intervalos para otras posibles secuencias, por tanto, cualquier valor de este intervalo se puede usar como un *tag* para codificar el texto ejemplo. Este valor es de libre elección, por lo cual se seleccionará el siguiente valor dada la menor cantidad de dígitos decimales:

$$\mathbf{tag = 0.17.}$$

Algoritmo de decodificación aritmética

El siguiente es el procedimiento para la decodificación aritmética (Sayood, 2017; Salomon y Motta, 2010):

1. Definir el *intervalo actual* como $[0, 1)$:

$$low = 0, \quad high = 1$$

2. Repetir los siguientes pasos mientras que la longitud de la cadena decodificada sea menor que la longitud de la secuencia codificada:

- Calcular *newtag*:

$$newtag = \frac{tag - low}{high - low}$$

- Hallar el símbolo x_k correspondiente al rango de la FDA en el que se encuentra *newtag*, este es el símbolo decodificado.
- Actualizar los límites del nuevo intervalo con los valores inferior y superior del rango seleccionado y hacer el valor de $tag = newtag$.

Ejemplo 10

Continuando con el ejemplo 9, para la secuencia CASA se obtuvo el **tag = 0.17**, para el cual a continuación se realizará el proceso de decodificación.

Partiendo de $low = 0$ y $high = 1$, se procede a realizar 4 iteraciones, ya que la longitud de la secuencia codificada es de 4 símbolos:



<i>newtag</i>	Rango	Símbolo
$(0.17-0)/(1-0) = \mathbf{0.17}$	$[0 - 0.25)$	C
$(0.17-0)/(0.25-0) = \mathbf{0.68}$	$[0.25 - 0.75)$	A
$(0.68-0.25)/(0.75-0.25) = \mathbf{0.86}$	$[0.75 - 1)$	S
$(0.86-0.75)/(1-0.75) = \mathbf{0.44}$	$[0.25 - 0.75)$	A

En la última columna se encuentra la secuencia decodificada obtenida.

Eficiencia de la codificación aritmética

“Sea s una secuencia de símbolos a codificar y b el número de bits necesarios para codificarlos. A medida que s se alarga, su probabilidad $P(s)$ se hace más pequeña y b se hace más grande. Como el logaritmo es la función que proporciona la cantidad de información, b debe crecer al mismo ritmo que $\log_2 P(s)$ decrece, por lo tanto, su producto debe ser casi constante.

La teoría de la información dice que b y $P(s)$ satisfacen:

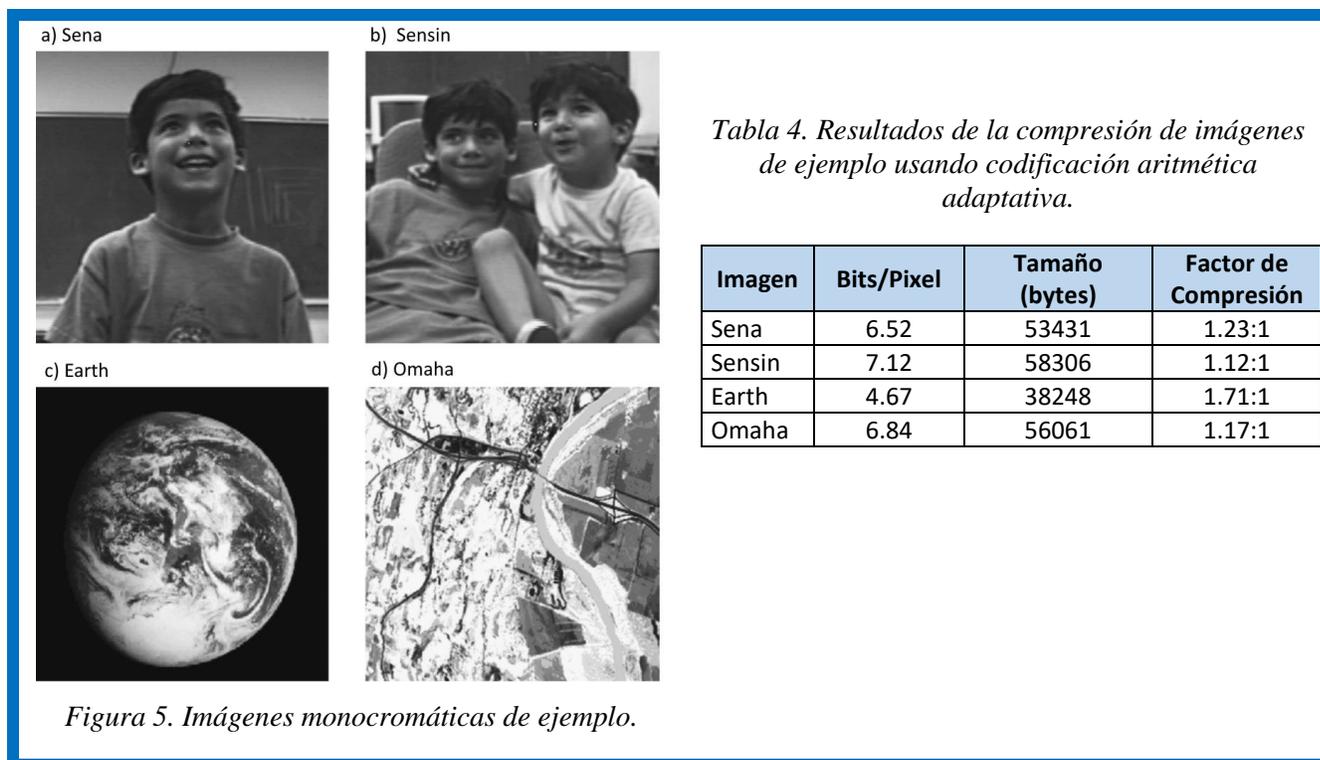
$$2 \leq 2^b P(s) < 4 \quad \rightarrow \quad 1 - \log_2 P(s) \leq b < 2 - \log_2 P(s) \quad (1-13)$$

A medida que s se alarga $P(s)$ disminuye, $-\log_2 P(s)$ se convierte en un número positivo más grande y de la ecuación (1-13) se ve que, en el límite, b se aproxima a $-\log_2 P(s)$. Es por esto por lo que la codificación aritmética puede, en principio, comprimir una cadena de símbolos a su límite teórico” (Salomon y Motta, 2010).

Aplicaciones de la codificación aritmética

Una aplicación de la codificación aritmética en la compresión de imágenes consiste en generar un código aritmético para el conjunto de valores que puede tomar cualquier píxel. Para imágenes monocromáticas, este conjunto usualmente consiste en enteros entre 0 y 255. Ejemplos de estas imágenes se muestran en la Figura 5.

Los resultados para las cuatro imágenes de la Figura 5 se muestran en la Tabla 4. La representación de las imágenes originales (sin comprimir) usa 8 bits/píxel. La imagen consta de 256 filas de 256 píxeles, así que la representación sin comprimir usa 65536 bytes (Sayood, 2017). Nótese que el factor de compresión es diferente para cada una de las imágenes.



Ejercicio 4

- Para el fragmento de texto "LAS_MQUINAS_NO_PUEDEN_PENSAR", obtenga el árbol del código Huffman.
- Con base en el árbol del código Huffman generado en el punto anterior, decodifique las secuencias binarias:
 - 100000101110
 - 11101011010011101101110
 - 10101011111110011111110
 - 10101111110111001010001
- Dada la frase "YO_NO_COMPRO_COCO_POR_QUE_POCO_COCO_COMO_Y_COMO_POCO_COCO_COMO_POCO_COCO_COMPRO", obtenga el tag de codificación para la palabra "COMPRO".

1.5 Compresión basada en contexto

 Hasta ahora se han visto una serie de técnicas que usan un modelo de probabilidad de la fuente para generar códigos eficientes. Si el modelo de probabilidad asumido no coincide con la distribución de probabilidad real de los símbolos, el código resultante puede no ser muy útil.

Se puede utilizar el contexto de un símbolo para comprimirlo. Además, conocer el contexto de un símbolo que está siendo codificado, podría permitir obtener un modelo de probabilidad más preciso para el símbolo. En esta sección veremos algoritmos que ponen en práctica estas ideas (Sayood, 2017).

1.5.1 Codificación run-length (RLE)

Tal vez el sistema de codificación más simple que toma ventaja del contexto es la codificación run-length (RLE, por su sigla en inglés, Run-Length Encoding) o codificación de rachas. Aunque existen variantes, la idea básica es identificar cadenas de símbolos adyacentes de igual valor y remplazarlas con una única ocurrencia junto con un contador (Blelloch, 2013).

Ejemplo 11

Si se tiene la cadena “aaabbccccdddd”, esta puede ser transformada en la cadena “3a2b4c5d”, logrando una compresión de la secuencia original.

Aplicaciones de RLE

Un ejemplo de uso real de RLE es el estándar ITU-T T4 (Group 3) para máquinas Fax, que transmiten imágenes en blanco y negro. El estándar T4 usa RLE para codificar cada secuencia de píxeles blancos y negros. El estándar T4 especifica el color de inicio ubicando un píxel blanco en primer lugar, con lo cual no es necesario enviar el color. Por ejemplo, la secuencia de rachas de blanco y negro “bbbbbbnbbbbbnnnnbnbn” se transforma en la secuencia “6b2n6b4n1b2n1b”. Debido al primer píxel blanco (no imprimible), el resultado es la secuencia “7264121” de solo 7

símbolos en comparación con los 22 caracteres iniciales para una razón de compresión de $RC = 7/22 = 0.32$. Adicionalmente, el estándar usa una codificación Huffman para codificar las rachas, y usa códigos separados para las rachas de pixeles blancos y negros (ver Tabla 5). Estos códigos de Huffman se basan en la probabilidad de cada racha medida sobre un gran número de documentos (Blelloch, 2013).

Tabla 5. Códigos Huffman RLE ITU-T T4 (Blelloch, 2013).

Longitud racha	Blanco – Palabra de código	Negro – Palabra de código
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
...
20	0001000	00001101000
...
64+	11011	0000001111
128+	10010	000011001000

1.5.2 Codificación mover-al-frente (MTF)

Otro esquema de codificación que toma ventaja del contexto es la codificación move-to-front (MTF, por su sigla en inglés, Move-To-Front), en particular toma ventaja de largas rachas de símbolos idénticos.

Algoritmo de codificación MTF (Sayood, 2017)

1. Comenzar con una lista del alfabeto fuente; al símbolo inicial de la lista se le asigna el número 0, al siguiente el número 1 y así sucesivamente.
2. Para cada símbolo de la cadena de entrada repetir:
 - Transmitir el número correspondiente a su lugar en la lista.
 - Mover el símbolo al inicio de la lista.

Ejemplo 92

A continuación, en la Tabla 6, se muestra un ejemplo de codificación MTF para la secuencia *bananaaa*:

Tabla 6. Ejemplo codificación MTF (Wikipedia).

Iteración	Lista símbolos indexados																									Salida	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		25
bananaaa	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1
bananaaa	b	a	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1
bananaaa	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	13
bananaaa	n	a	b	c	d	e	f	g	h	i	j	k	l	m	o	p	q	r	s	t	u	v	w	x	y	z	1
bananaaa	a	n	b	c	d	e	f	g	h	i	j	k	l	m	o	p	q	r	s	t	u	v	w	x	y	z	1
bananaaa	n	a	b	c	d	e	f	g	h	i	j	k	l	m	o	p	q	r	s	t	u	v	w	x	y	z	1
bananaaa	a	n	b	c	d	e	f	g	h	i	j	k	l	m	o	p	q	r	s	t	u	v	w	x	y	z	0
bananaaa	a	n	b	c	d	e	f	g	h	i	j	k	l	m	o	p	q	r	s	t	u	v	w	x	y	z	0
bananaaa	Codificación MTF final:																									1,1,13,1, 1,1,0,0	

Si se tiene una racha de símbolos, se transmite una secuencia de ceros; de este modo, largas rachas de cualquier símbolo se transforman en largas secuencias de 0's, que pueden entonces codificarse eficientemente usando esquemas diseñados específicamente para codificar largas repeticiones del mismo símbolo, tal como RLE, estudiado en la sección anterior.

1.5.3 Codificación PPM

La idea del algoritmo PPM (por su sigla en inglés, Prediction by Partial Matching) es usar contextos de alto orden para determinar la probabilidad del símbolo que está siendo codificado (el contexto de orden N de un símbolo son los N símbolos que lo preceden).

Sin embargo, usar contextos de alto orden requiere estimar y almacenar grandes cantidades de probabilidades, lo cual podría no ser factible. Se puede reducir la carga estimando las probabilidades a medida que la codificación avanza. De esta manera solo se necesita almacenar los contextos que han ocurrido en la secuencia que está siendo codificada, que son un número mucho menor que el número de todos los posibles contextos.

Si bien esto mitiga el problema del almacenamiento, también significa que se necesitará codificar símbolos que no han ocurrido previamente en un contexto dado. Para manejar esta situación, el alfabeto del codificador de la fuente contiene un símbolo *escape* que se usa para señalar que el símbolo a ser codificado no se ha visto en dicho contexto (Sayood, 2017).

Algoritmo de codificación PPM

La implementación de este algoritmo se lleva a cabo en dos etapas: (i) codificación del símbolo y (ii) actualización de las probabilidades en la estructura de datos.

i. Codificación del símbolo

1. Establecer el máximo orden del contexto, N .
2. Para cada símbolo s de la secuencia de entrada a codificar, repetir:
 - Mientras no se encuentre una aparición previa del contexto C de orden N seguido por el símbolo s :
 - Enviar caracter *escape*.
 - Reducir el orden del contexto: $N=N-1$.
 - Determinar la probabilidad P de que s aparezca en los siguientes datos de entrada con el contexto particular C :
 - Si orden del contexto $N=-1$, $P=1/(\text{tamaño del alfabeto})$.
 - Si no, P será la probabilidad respectiva actual en la estructura de datos para el contexto C de orden N seguido por el símbolo s .
 - Invocar un algoritmo de codificación (Huffman o aritmética) para codificar el símbolo s con la probabilidad P obtenida.

El decodificador, con base en la cantidad de caracteres *escape* recibidos, podrá identificar el orden del contexto para decodificar cada símbolo, y de esta manera estará sincronizado con el codificador (Salomon y Motta, 2010).

ii. Actualización de las probabilidades en la estructura de datos

Se da inicio con una estructura de datos vacía para los contextos de órdenes de N a 0 y la codificación estática de los símbolos de la DMS más el caracter *escape* para el orden -1 . La estructura de datos será idéntica y se realizaran las mismas actualizaciones de manera síncrona tanto en el codificador como en el decodificador.

La actualización de la estructura de datos puede presentar dos casos:

- **Símbolo codificado en el orden -1 :** es decir que no existió coincidencia de contexto en ningún orden, por lo que se procede con la actualización del orden 0 adicionando el símbolo codificado con frecuencia 1 , así como con la actualización de los contextos de órdenes superiores hasta N en los que se incluye el símbolo codificado y su respectivo contexto con frecuencia 1 .
- **Símbolo codificado en un orden específico N' :** Si existe coincidencia del contexto en un orden N' , se procede a incrementar la frecuencia del actual símbolo para todas sus entradas de predicción para los órdenes de contexto de 0 a N .

Adicionalmente, tras la actualización de cada contexto, es necesario actualizar el caracter *escape* haciendo su frecuencia igual a la cantidad de predicciones existentes en el contexto, por lo que se deben actualizar las probabilidades de las predicciones.

Ejemplo 13

La Tabla 7 muestra los contextos hasta el orden 2 generados durante la lectura y codificación de la cadena de 14 símbolos “*assanissimassa*” (Salomon y Motta, 2010). Se supone que el alfabeto se compone de las 26 letras, el espacio en blanco y el símbolo *escape*: un total de 28 símbolos. La probabilidad de encontrar un símbolo con orden -1 es por lo tanto de $1/28$. Sin compresión se necesitan $\log_2 28 \cong 5$ bits para codificar uno de los 28 símbolos.

Tabla 7. Estructura de datos PPM tras codificar el texto “assanissimassa”; incluye las frecuencias (f) y las probabilidades (p) (Salomon & Motta, 2010).

Orden 2			Orden 1			Orden 0		
Contexto	f	p	Contexto	f	p	Símbolo	f	p
as → s	2	2/3	a → s	2	2/5	a	4	4/19
esc	1	1/3	a → n	1	1/5	s	6	6/19
			esc →	2	2/5	n	1	1/19
ss → a	2	2/5	s → s	3	3/9	i	2	2/19
ss → i	1	1/5	s → a	2	2/9	m	1	1/19
esc	2	2/5	s → i	1	1/9	esc	5	5/19
			esc	3	3/9			
sa → n	1	1/2	n → i	1	1/2			
esc	1	1/2	esc	1	1/2			
			i → s	1	1/4			
an → i	1	1/2	i → m	1	1/4			
esc	1	1/2	esc	2	2/4			
			m → a	1	1/2			
ni → s	1	1/2	esc	1	1/2			
esc	1	1/2						
is → s	1	1/2						
esc	1	1/2						
si → m	1	1/2						
esc	1	1/2						
im → a	1	1/2						
esc	1	1/2						
ma → s	1	1/2						
esc	1	1/2						

El contexto actual de orden 2 es “sa”. Suponer que el siguiente símbolo de la cadena de entrada es “s” (Salomon & Motta, 2010).

Codificación del símbolo:

- El codificador PPM observa que “sa” nunca ha sido visto antes seguido de una “s”. Por ello el codificador envía el símbolo *escape* al codificador aritmético o Huffman, junto con la probabilidad $1/2$ predicha por el contexto de orden 2 de “sa”. Por lo tanto, gasta $-\log_2\left(\frac{1}{2}\right) = 1$ bit para codificar este *escape*.

- Cambiando al orden 1, el contexto actual se convierte en “a”, y el codificador PPM encuentra que una “a” seguida por una “s” ya ha aparecido antes y actualmente tiene una probabilidad asignada de $2/5$. La “s” es enviada al codificador aritmético o Huffman para que la codifique con esta probabilidad, lo que produce otros $-\log_2\left(\frac{2}{5}\right) = 1.32$ bits .

- En total se generan $1+1.32=2.32$ bits para codificar la “s”.

Actualización de la estructura de datos:

- En el orden 0, se incrementa la frecuencia del caracter “s” y se actualizan las probabilidades de todos los símbolos.
- En el orden 1, se actualiza la frecuencia de la predicción “ $a \rightarrow s$ ” y se actualizan las probabilidades de las predicciones de este contexto.
- Y finalmente, en el orden 2, se adiciona la predicción “ $sa \rightarrow s$ ”, se incrementa la frecuencia del caracter *escape* a 2 debido a que ahora se encuentra en un contexto con 2 posibles predicciones, y se actualizan las probabilidades de las predicciones de este contexto.

El resultado se muestra en la Tabla 8:

Tabla 8. Actualización estructura de datos PPM.

Orden 2			Orden 1			Orden 0		
Contexto	<i>f</i>	<i>p</i>	Contexto	<i>f</i>	<i>p</i>	Símbolo	<i>f</i>	<i>p</i>
as → s	2	2/3	a → s	3	3/6	a	4	4/20
esc	1	1/3	a → n	1	1/6	s	7	7/20
ss → a	2	2/5	esc →	2	2/6	n	1	1/20
ss → i	1	1/5	s → s	3	3/9	i	2	2/20
esc	2	2/5	s → a	2	2/9	m	1	1/20
sa → s	1	1/4	s → i	1	1/9	esc	5	5/20
sa → n	1	1/4	esc	3	3/9			
esc	2	2/4	n → i	1	1/2			
an → i	1	1/2	esc	1	1/2			
esc	1	1/2	i → s	1	1/4			
ni → s	1	1/2	i → m	1	1/4			
esc	1	1/2	esc	2	2/4			
is → s	1	1/2	m → a	1	1/2			
esc	1	1/2	esc	1	1/2			
si → m	1	1/2						
esc	1	1/2						
im → a	1	1/2						
esc	1	1/2						
ma → s	1	1/2						
esc	1	1/2						

Ejercicio 5

1. Seleccione la respuesta correcta a las siguientes preguntas:

- a) ¿En qué consiste la codificación RLE?
- b) ¿Cuándo se envía el caracter virtual *escape*?
- c) ¿Cuál codificación reorganiza el orden de los símbolos de mayor a menor frecuencia?
- d) ¿Cuál es la codificación RLE para la secuencia “aaabbccccddd”?
- e) ¿Con qué codificación se pueden comprimir las secuencias cortas como las generadas por RLE?

2. Realice la compresión MTF de la cadena “*sshthh_ii_e*” con base en el siguiente diccionario inicial (Sayood, 2017):

0	1	2	3	4	5
–	e	h	i	s	t

3. Tras la codificación PPM del catorceavo caracter (“a”) de la cadena “*assanissimassa*”, se obtiene la estructura de datos de la Tabla 7. Determine el nuevo estado de los contextos de orden 0, si el siguiente símbolo de entrada es “d” (Salomon & Motta, 2010).
4. Con referencia al numeral anterior, determine cuántos bits se requieren para codificar la “d” (Salomon & Motta, 2010).

1.6 Métodos de diccionario

 En secciones anteriores, se han estudiado algoritmos que hacen uso de la estructura probabilística de los datos para crear esquemas de codificación eficientes. La estructura probabilística no es la única clase de estructura existente en datos. En esta sección, se verán métodos de codificación que hacen uso de la existencia de patrones repetitivos en los datos.

La idea básica es simple: colocar los patrones recurrentes en un diccionario, y siempre que el patrón ocurra en los datos a comprimir, codificarlo con una referencia al diccionario. Se dice que el diccionario es *estático* si se crea al inicio de la compresión y se usa por un largo tiempo, o *adaptativo* si se construye de manera dinámica a medida que la compresión avanza. Estos

esquemas adaptativos están basados en una aproximación de análisis sintáctico desarrollada por Abraham Lempel y Jacob Ziv (Sayood, 2017).

1.6.1 Lempel-Ziv 77 (LZ77)

En la aproximación LZ77, el diccionario es una porción de la secuencia previamente codificada. El codificador examina la secuencia de entrada a través de una ventana deslizante que consiste de 2 partes: un *buffer de búsqueda* (diccionario) que contiene una porción de la secuencia recientemente codificada, y un *buffer de lectura anticipada* que contiene la siguiente porción de la secuencia a ser codificada (Sayood, 2017).

Algoritmo de codificación LZ77 (Suman, Anjali, Sonal, 2015; Blelloch, 2013; Sayood, 2017)

Mientras el buffer de lectura anticipada no esté vacío, hacer:

1. Hallar la coincidencia más larga entre un string que comienza al inicio del *buffer de lectura anticipada* y está completamente contenido en este, y un string que comienza en el *buffer de búsqueda* (diccionario).
2. Generar como salida la tripla $\langle o, l, c \rangle$, donde o (*offset*) representa la ubicación de la coincidencia, l la longitud de la coincidencia, y c el siguiente símbolo después de la coincidencia (siguiente símbolo a codificar).

La posición o se puede expresar con referencia al inicio del *buffer de lectura anticipada*, con “0” indicando que no se encontró coincidencia, “1” indicando una coincidencia que inicia en el símbolo previo, etc.

c se envía para considerar la situación en la que no se encuentre coincidencia para este símbolo en el *buffer de búsqueda*, caso en el que se envía la tripla $\langle 0, 0, c \rangle$.

3. Desplazar la ventana $l+1$ símbolos hacia adelante.

Si el tamaño del *buffer de búsqueda* (diccionario) es S , el tamaño de la ventana (buffers de búsqueda y de lectura anticipada) es W , y el tamaño del alfabeto fuente es A , entonces el número de bits necesarios para codificar la tripla es $\lceil \log_2(S) \rceil + \lceil \log_2(W) \rceil + \lceil \log_2(A) \rceil$. El segundo término es $\lceil \log_2(W) \rceil$ porque la longitud de la coincidencia puede exceder la longitud del *buffer de búsqueda* (Sayood, 2017).

Ejemplo 14

La Figura 6 muestra la codificación LZ77 para la cadena “abracadabrad”, con un *buffer de búsqueda* (diccionario) de tamaño 7 y un *buffer de lectura anticipada* de tamaño 5.

Cadena a codificar											Salida	
a	b	r	a	c	a	d	a	b	r	a	d	(0,0,a)
a	b	r	a	c	a	d	a	b	r	a	d	(0,0,b)
a	b	r	a	c	a	d	a	b	r	a	d	(0,0,r)
a	b	r	a	c	a	d	a	b	r	a	d	(3,1,c)
a	b	r	a	c	a	d	a	b	r	a	d	(2,1,d)
a	b	r	a	c	a	d	a	b	r	a	d	(7,4,d)
a	b	r	a	c	a	d	a	b	r	a	d	

Coincidencia
 Siguiete caracter

Figura 6. Ejemplo de la codificación LZ77 (Suman, Anjali, Sonal, 2015)

Los 12 caracteres del texto de ejemplo sin comprimir requieren 96 bits, mientras que las 6 triplas generadas requieren (con $S=7$, $W=12$ y $A=26$)

$$6 * ([\log_2(7)] + [\log_2(12)] + [\log_2(26)]) = 6 * (3 + 4 + 5) = 72 \text{ bits ,}$$

lo que indica que la codificación alcanzó una reducción por compresión de 25%:

$$\text{Reducción Compresión} = (1 - 0.75) * 100\% = 25\% .$$

Algoritmo de decodificación LZ77

Para ilustrar la decodificación se considera un solo paso. Inductivamente se asume que el decodificador ha construido correctamente la cadena hasta la posición actual de inicio del *buffer de lectura anticipada*, y se muestra que dada la tripla $\langle o, l, c \rangle$, el decodificador puede reconstruir la cadena hasta la siguiente posición de inicio de dicho buffer. Para hacer esto, el decodificador realiza los siguientes pasos (Salomon & Motta, 2010; Blelloch, 2013):

1. Retroceder o posiciones.
2. Copiar y escribir a la salida los siguientes l caracteres.
3. Escribir a la salida el tercer elemento de la tripla.

Ejemplo 15

La Figura 7 muestra la decodificación de la tripla $\langle 7, 4, r \rangle$ (Sayood, 2017).

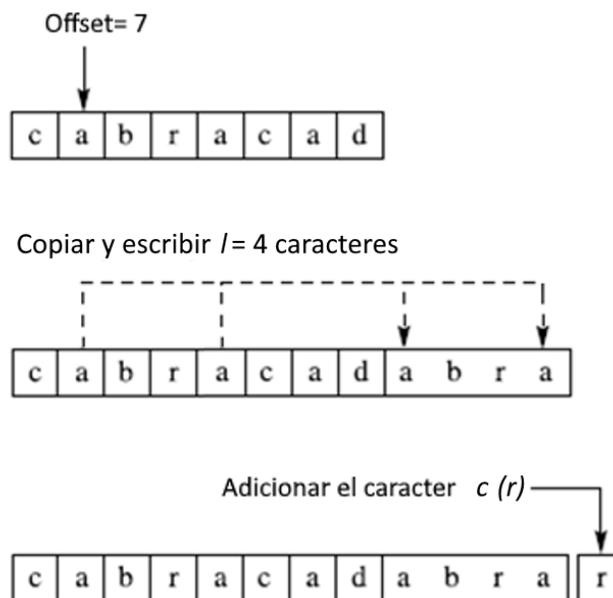


Figura 7. Decodificación de la tripla $\langle 7, 4, r \rangle$

Como se puede ver, LZ77 es un esquema adaptativo muy simple que no requiere conocimiento previo de la fuente y parece no requerir suposiciones acerca de sus características. Los autores de este algoritmo mostraron que su desempeño aproxima asintóticamente lo mejor que se podría obtener usando un esquema que tuviera pleno conocimiento de las estadísticas de la fuente. Si bien esto puede ser cierto de forma asintótica, en la práctica hay varias maneras de mejorar el desempeño del algoritmo LZ77. Además, al usar las porciones recientes de la secuencia, hay una especie de suposición que se utiliza: que los patrones se repiten muy cercanamente. Los autores removieron esta suposición e idearon un esquema diferente basado en diccionario adaptativo en LZ78 (Sayood, 2017).

Aplicaciones de la codificación LZ77

El estándar para compresión de imágenes PNG (por su sigla en inglés Portable Network Graphics), es uno de los primeros desarrollados colaborativamente en internet. El algoritmo de compresión usado en PNG está basado en la implementación *deflate* de LZ77. Esta implementación permite

longitudes de coincidencia de entre 3 y 258. En cada paso, el codificador examina 3 bytes, si no puede encontrar una coincidencia de al menos 3 bytes, saca el primer byte y examina los siguientes 3 bytes. Así, en cada paso saca el valor de un único byte o *literal*, o la pareja $\langle o, l \rangle$. Los alfabetos del *literal* y de *l* se combinan para formar un alfabeto de tamaño 286 (indexado mediante 0 - - 285). Los índices 0 – 255 representan bytes del literal y el índice 256 es un símbolo de final de bloque; los restantes 29 índices representan códigos para rangos de longitudes de entre 3 y 258, como se muestra en la Tabla 9 (Sayood, 2017).

Índice	Cantidad Bits Selectores	Longitud Coincidencia	Índice	Cantidad Bits Selectores	Longitud Coincidencia	Índice	Cantidad Bits Selectores	Longitud Coincidencia
257	0	3	267	1	15-16	277	4	67-82
258	0	4	268	1	17-18	278	4	83-98
259	0	5	269	2	19-22	279	4	99-114
260	0	6	270	2	23-26	280	4	115-130
261	0	7	271	2	27-30	281	5	131-162
262	0	8	272	2	31-34	282	5	163-194
263	0	9	273	3	35-42	283	5	195-226
264	0	10	274	3	43-50	284	5	227-257
265	1	11-12	275	3	51-58	285	0	258
266	1	13-14	276	3	59-66			

Tabla 9. Códigos para la representación de la longitud de coincidencia. Se muestra el índice, el número de bits selectores para seguir el índice, y las longitudes representadas por el índice y los bits selectores. Por ejemplo, el índice 277 representa el rango de longitudes desde 67 hasta 82, para especificar cuál de los 16 valores realmente ocurrió, el código es seguido por 4 bits selectores.

Los valores del índice se representan utilizando un código Huffman, el cual se muestra en la Tabla 10.

Tabla 10. Código Huffman para el alfabeto de la longitud de coincidencia.

Rangos Índice	# de Bits	Códigos Binarios
0-143	8	00110000 - 10111111
144-255	9	110010000 - 111111111
256-279	7	0000000 - 0010111
280-287	8	11000000 - 11000111

El *offset* puede tomar valores entre 1 y 32768, los cuales se dividen en 30 rangos. Los valores de los 30 rangos se codifican también usando un código Huffman, diferente a los códigos Huffman del *literal* y los valores de la *longitud de coincidencia*, y el código es seguido por un número de bits selectores para especificar la distancia particular dentro del rango.

1.6.2 Lempel-Ziv-Welch (LZW)

En esta sección se describirá la variante LZW de LZ78, debida a Terry Welch, ya que es la más usada en la práctica. El uso de un diccionario es similar a LZ77, excepto por que el diccionario se almacena explícitamente en lugar de índices de una ventana (Blelloch, 2013).

La palabra de código consiste ahora de únicamente un índice de la posición en el diccionario de la cadena encontrada. Para esto, el diccionario debe ser inicializado con todas las letras o símbolos del alfabeto fuente (Morales, 2003; Sayood, 2017).

Algoritmo de codificación LZW

Cada vez que el codificador lee un símbolo de entrada x , lo concatena en una cadena S y se busca en el diccionario la nueva cadena $S \cdot x$ (\cdot denota concatenación); este proceso se repite mientras $S \cdot x$ se encuentre en el diccionario. En un momento dado, la concatenación del siguiente símbolo x provoca un fallo en la búsqueda: la cadena S está en el diccionario, pero la cadena $S \cdot x$ no. En este punto, el codificador: (a) escribe a la salida el índice del diccionario correspondiente a la cadena S , (b) agrega la cadena $S \cdot x$ (llamada frase) en la siguiente entrada disponible del diccionario, y (c) inicializa la cadena S con el símbolo x (Salomon & Motta, 2010).

A continuación, se encuentran los pasos que sigue el algoritmo de codificación LZW (Salomon & Motta, 2010; Morales, 2003):

1. Inicializar el diccionario con todas las letras o símbolos del alfabeto fuente.
2. S = cadena vacía.
3. Mientras existan símbolos en la entrada:
 - x = siguiente símbolo de entrada
 - Si $S \cdot x$ se encuentra en el diccionario:
 - $S = S \cdot x$
 - Si no:
 - Escribir en la salida el índice en el diccionario de la cadena S
 - Adicionar $S \cdot x$ al diccionario, y hacer $S = x$

Ejemplo 16

Obtener la codificación LZW para la cadena “poco_coco”.

A continuación, se muestra: a) un resumen de la ejecución del algoritmo y b) el diccionario final obtenido (las celdas en gris corresponden a la inicialización del diccionario).

RESUMEN DE LA EJECUCIÓN DEL ALGORITMO						DICCIONARIO	
S	x	S · x	S · x en diccionario	Salida	Adicionar al diccionario	Índice	Símbolo/Cadena
""	p	p	SI			1	c
p	o	po	NO	3	5 → po	2	o
o	c	oc	NO	2	6 → oc	3	p
c	o	co	NO	1	7 → co	4	–
o	_	o_	NO	2	8 → o_	5	po
_	c	_c	NO	4	9 → _c	6	oc
c	o	co	SI			7	co
co	c	coc	NO	7	10 → coc	8	o_
c	o	co	SI			9	_c
						10	coc

Se ha llegado al final de la cadena, por tanto, se escribe a la salida el índice del buffer S almacenado:

$$S = co,$$

obteniendo así el código LZW de la cadena ejemplo “poco_coco”:

$$\text{Salida} = 3212477.$$

 LZW se adapta lentamente a la entrada de los datos ya que la construcción de las cadenas o frases se realiza caracter a caracter. A medida que la codificación avanza, la longitud de las entradas del diccionario continúa aumentando; entradas más largas indican que el diccionario está capturando más estructura de la secuencia (Morales, 2003; Sayood, 2017).

Algoritmo de decodificación LZW

La decodificación LZW se realiza de la siguiente forma: se inicializa el diccionario con todas las letras o símbolos del alfabeto fuente. El decodificador lee el archivo comprimido, que consiste en índices al diccionario que indican la posición de donde se recuperarán los símbolos no comprimidos para escribirlos en el archivo de salida. El decodificador también construye el diccionario de la misma forma que el codificador, conforme nuevos símbolos son recuperados del diccionario.

Cuando se lee el primer índice del archivo comprimido, se recupera el símbolo x almacenado en esa posición en el diccionario y se asigna a S ($S=x$). S se escribe al archivo de salida y se lee el siguiente índice. El contenido del índice se almacena en una variable T y también se escribe en el archivo de salida, el primer símbolo S_1 de T se concatena con S para formar la cadena $S \cdot S_1$, si $S \cdot S_1$ no existe en el diccionario entonces se agrega. S es ahora T , se lee el siguiente índice y se repiten los mismos pasos (Morales, 2003).

A continuación se encuentran los pasos de la decodificación LZW:

1. Inicializar el diccionario con todas las letras o símbolos del alfabeto fuente.
2. S = cadena vacía.
3. Mientras existan índices en el archivo comprimido:
 - Ind = Siguiete índice en archivo comprimido.
 - T = Diccionario[Ind]. Escribir al archivo de salida T .
 - S_1 = Primer símbolo de T .
 - Adicionar $S \cdot S_1$ al diccionario, si dicha cadena no se encuentra.
 - $S = T$.

Ejemplo 17

Tomar la salida del codificador del ejemplo 16: “3 2 1 2 4 7 7”, y decodificarla usando el algoritmo LZW.

A continuación se muestra: a) un resumen de la ejecución del algoritmo y b) el diccionario final obtenido (las celdas en gris corresponden a la inicialización del diccionario).

RESUMEN DE LA EJECUCIÓN DEL ALGORITMO					
<i>S</i>	<i>Ind.</i>	<i>T</i>	Salida	<i>S</i> ₁	Adicionar al diccionario (<i>S</i> · <i>S</i> ₁)
""	3	<i>p</i>	<i>p</i>	<i>p</i>	
<i>p</i>	2	<i>o</i>	<i>o</i>	<i>o</i>	5 → <i>po</i>
<i>o</i>	1	<i>c</i>	<i>c</i>	<i>c</i>	6 → <i>oc</i>
<i>c</i>	2	<i>o</i>	<i>o</i>	<i>o</i>	7 → <i>co</i>
<i>o</i>	4	_	_	_	8 → <i>o_</i>
_	7	<i>co</i>	<i>co</i>	<i>c</i>	9 → <i>_c</i>
<i>co</i>	7	<i>co</i>	<i>co</i>	<i>c</i>	10 → <i>coc</i>

DICCIONARIO	
Índice	Símbolo/Cadena
1	<i>c</i>
2	<i>o</i>
3	<i>p</i>
4	_
5	<i>po</i>
6	<i>oc</i>
7	<i>co</i>
8	<i>o_</i>
9	<i>_c</i>
10	<i>coc</i>

Se ha finalizado la decodificación, recuperando la cadena original mostrada:

Cadena decodificada= "poco_coco"

Como se observa, el diccionario a partir de los símbolos iniciales se va reconstruyendo con base en los índices recibidos.

Aplicaciones de la codificación LZW

El comando *compress* de UNIX es una de las primeras aplicaciones de LZW. El tamaño del diccionario es adaptativo. Se comienza con un tamaño de 512, esto significa que las palabras de código transmitidas tienen una longitud de 9 bits. Una vez que el diccionario se ha llenado, su tamaño se dobla a 1024 entradas, de modo que las palabras de código transmitidas en este punto tienen 10 bits. El tamaño del diccionario se duplica progresivamente a medida que se llena. De esta manera, durante la parte inicial del proceso de codificación, cuando las cadenas en el diccionario no son muy largas, las palabras de código usadas para codificarlas también tienen pocos bits.

El tamaño máximo de la palabra de código, b_{max} , puede ser establecido por el usuario entre 9 y 16, siendo 16 bits el valor por defecto. Una vez que el diccionario contiene $2^{b_{max}}$ entradas, *compress* se convierte en una técnica de codificación de diccionario estático. En este punto el algoritmo monitorea la razón de compresión, si cae debajo de un umbral, el diccionario se vacía y se reinicia su proceso de construcción. De este modo, el diccionario siempre refleja las características locales de la fuente (Sayood, 2017).

Ejercicio 6

1. Encuentre la codificación LZ77 para el texto “PEPE_PECAS_PICA_PAPAS”, considerando que tanto el *diccionario* como el *buffer de lectura anticipada* son de tamaño 10.
2. Con base en el diccionario inicial dado, realice la codificación LZW para el texto “PEPE_PECAS_PICA_PAPAS”.

Índice	Símbolo/ Cadena
1	A
2	C
3	E
4	I
5	P
6	S
7	_

1.7 Técnicas de compresión con pérdida

Se denomina compresión con pérdida a todo tipo de compresión en la que la información final decodificada presenta diferencias con la información original, en el sentido en que se pierde parte de esta información la cual se espera sea imperceptible para el usuario. Adicionalmente, es importante identificar correctamente el tipo de información a comprimir con esta clase de algoritmos, ya que estos aprovechan características específicas de cada tipo, tales como frecuencias audibles, tasas de muestreo entre otras, para reducir la información original.

A continuación, se explicarán algunas técnicas de compresión con pérdidas, tanto aplicables de manera general, como también para tipos de información específicos.

1.7.1 Cuantización escalar

En muchas aplicaciones de compresión con pérdidas, se requiere representar cada salida de la fuente usando uno de entre un número pequeño de palabras de código. El número de posibles salidas de la fuente es en general mucho más grande que el número de palabras de código disponibles para representarlas. El proceso de representar un conjunto muy grande de valores mediante un conjunto más pequeño de valores se conoce como *cuantización*.

El problema de cuantización

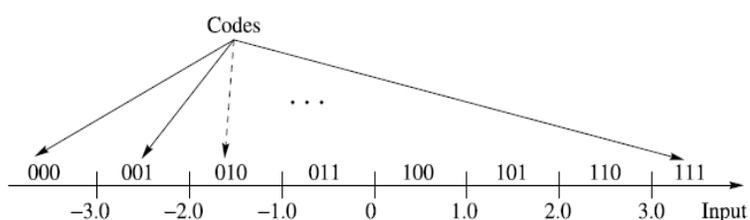
El cuantizador consta de dos mapeos (Sayood, 2017):

1. El *mapeo del codificador*, que divide el rango de valores que genera la fuente en un número de intervalos. El codificador representa todas las salidas de la fuente que caen en un intervalo particular mediante una palabra de código que representa ese intervalo. Como puede haber muchos valores de muestras distintos que pueden caer en un intervalo dado, el mapeo del codificador es irreversible y genera pérdida de información. Cuando el valor de muestra proviene de una fuente análoga, el codificador se conoce como conversor analógico-digital (A/D).
2. El *mapeo del decodificador*, que para cada palabra de código generada por el codificador, genera un valor reconstruido. Debido a que una palabra de código representa un intervalo completo, no hay manera de saber cuál valor en el intervalo fue generado por la fuente, y el decodificador genera un valor que representa todos los valores en el intervalo, usualmente el punto medio del intervalo. Si la reconstrucción genera una señal analógica, el decodificador se conoce como un conversor digital-analógico (D/A).

Ejemplo 108

En la Figura 8(a) se muestra el mapeo del codificador para 8 valores de reconstrucción, y en la Figura 8(b) se muestra el mapeo del decodificador correspondiente. La pareja codificador-decodificador se conoce como *cuantizador*, y el mapeo del cuantizador para la pareja codificador-decodificador de 3 bits mostrada en las Figura 8(a) y 8(b) se puede representar mediante el mapa de entrada-salida de la Figura 8(c).

(a)



(b)

Input Codes	Output
000	-3.5
001	-2.5
010	-1.5
011	-0.5
100	0.5
101	1.5
110	2.5
111	3.5

(c)

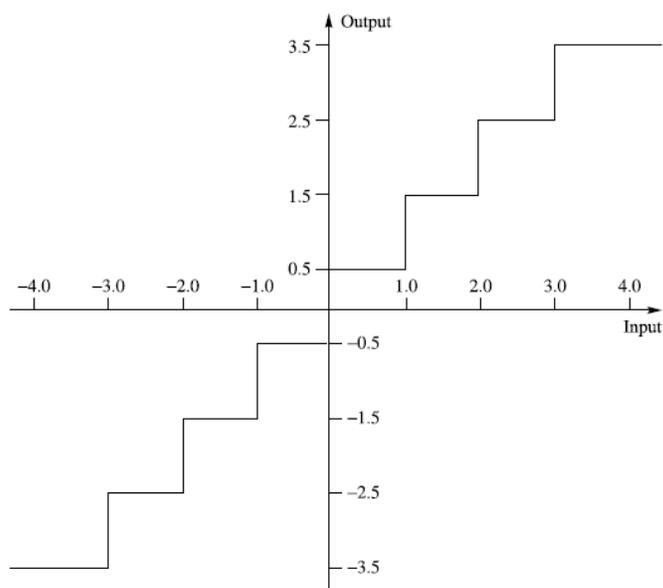


Figura 8. (a) Mapeo para un codificador de 3 bits. A cualquier entrada que se reciba se le asignará una palabra de código dependiendo del intervalo en el que caiga dicha entrada. Como se están usando 3 bits para representar cada valor, este codificador se conoce como codificador (cuantizador) de 3 bits. (b) Mapeo del decodificador correspondiente al codificador de (a). (c) Mapa de entrada-salida del cuantizador. Dependiendo del intervalo en el que caiga cada valor de muestra que acepta el cuantizador, este proporciona una palabra de código de salida y un valor de representación (Sayood, 2017).

Para plantear el problema de diseño de un cuantizador, suponer que se tiene una entrada modelada por una variable aleatoria X con pdf $f_X(x)$. Si se quiere cuantizar esta fuente usando un cuantizador con M intervalos, se deben especificar $M+1$ extremos para los intervalos, conocidos como *fronteras de decisión* $\{b_i\}_{i=0}^M$, y valores representativos para cada uno de los intervalos, conocidos como *niveles de reconstrucción* $\{y_i\}_{i=1}^M$ [1].

Si se usan palabras de código de longitud fija para representar la salida del cuantizador, entonces el tamaño del alfabeto de salida especifica inmediatamente la tasa del cuantizador, es decir, el número promedio de bits requeridos para representar una salida del cuantizador. Si el número de salidas del cuantizador es M , entonces la tasa está dada por:

$$R = \lceil \log_2 M \rceil . \quad (1-14)$$

En este caso, el problema de diseño del cuantizador se puede plantear así (Sayood, 2017):

*Dada una entrada con pdf $f_X(x)$ y el número de niveles M del cuantizador, hallar las fronteras de decisión $\{b_i\}_{i=0}^M$ y los niveles de reconstrucción $\{y_i\}_{i=1}^M$ que minimicen el **error cuadrático medio de cuantización (distorsión)** dado por*

$$\sigma_q^2 = \int_{-\infty}^{\infty} (x - Q(x))^2 f_X(x) dx = \sum_{i=1}^M \int_{b_{i-1}}^{b_i} (x - y_i)^2 f_X(x) dx, \quad (1-15)$$

donde $Q(\cdot)$ denota la operación de cuantización, de modo que

$$Q(x) = y_i \quad \text{sii} \quad b_{i-1} < x \leq b_i,$$

y la diferencia entre la entrada del cuantizador x y la salida $y = Q(x)$ se conoce como **error de cuantización** o **distorsión del cuantizador** o **ruido de cuantización**.

Si se permite el uso de códigos de longitud variable, tales como códigos Huffman o aritméticos, junto con el tamaño del alfabeto, la selección de las fronteras de decisión también afectará la tasa del cuantizador. Si n_i es la longitud de la palabra de código correspondiente a la salida y_i , y $P(y_i)$ es la probabilidad de ocurrencia de y_i , entonces la tasa está dada por

$$R = \sum_{i=1}^M n_i P(y_i). \quad (1-16)$$

[1] Frecuentemente se modelan fuentes discretas con distribuciones continuas debido a que estos modelos pueden simplificar el proceso de diseño considerablemente, y los diseños resultantes funcionan bien a pesar de la suposición incorrecta.

Sin embargo, las probabilidades $\{P(y_i)\}$ dependen de las fronteras de decisión $\{b_i\}$:

$$P(y_i) = \int_{b_{i-1}}^{b_i} f_X(x) dx .$$

Por lo tanto, la tasa R es una función de las fronteras de decisión y está dada por:

$$R = \sum_{i=1}^M n_i \int_{b_{i-1}}^{b_i} f_X(x) dx . \quad (1-17)$$

De esta discusión y de las ecuaciones (1-15) y (1-17), se ve que, para una entrada de fuente dada, las particiones que se seleccionen y su representación determinarán la distorsión generada durante el proceso de cuantización. Las particiones que se seleccionen y sus códigos binarios determinarán la tasa del cuantizador. Por tanto, el problema de encontrar las particiones, los códigos y los niveles de representación óptimos, están todos conectados, y se puede reformular el enunciado del problema de diseño de un cuantizador (Sayood, 2017):

Dada una restricción de distorsión

$$\sigma_q^2 \leq D^*$$

hallar las fronteras de decisión, los niveles de reconstrucción y los códigos binarios que minimicen la tasa dada por (1-17) mientras que se satisface esta restricción.

O dada una restricción de tasa

$$R \leq R^*$$

hallar las fronteras de decisión, los niveles de reconstrucción y los códigos binarios que minimicen la distorsión dada por (1-15) mientras que se satisface esta restricción.

Este enunciado del problema de diseño de un cuantizador es más general y también más complejo. Afortunadamente, en la práctica hay situaciones en las que se puede simplificar el problema, como se verá a continuación.

Cuantización uniforme de una fuente distribuida uniformemente

El tipo más simple de cuantizador es el uniforme, en el que todos los intervalos son del mismo tamaño, excepto posiblemente los dos intervalos exteriores, es decir, las fronteras de decisión están espaciadas uniformemente. Los valores de reconstrucción también están espaciados uniformemente, con el mismo espaciamiento de las fronteras de decisión; en los intervalos interiores estos valores son los puntos medios de los intervalos. Este espaciado constante se conoce como el tamaño del paso Δ . El cuantizador de la Figura 8 es un cuantizador uniforme con $\Delta=1$. Por lo tanto, el diseño de un cuantizador uniforme consiste en encontrar el tamaño del paso Δ que minimiza la distorsión para un proceso de entrada dado y un número de niveles de decisión M (Sayood, 2017).

El caso más simple de diseño de un cuantizador es: diseñar un cuantizador uniforme para una fuente distribuida uniformemente. Suponer que se quiere diseñar un cuantizador de M niveles para una entrada que está uniformemente distribuida en el intervalo $[-X_{\max}, X_{\max}]$. Se necesita dividir este intervalo en M intervalos con el mismo tamaño

$$\Delta = \frac{2 X_{\max}}{M}. \quad (1-18)$$

La distorsión en este caso es (a partir de la ecuación (1-15)):

$$\sigma_q^2 = 2 \sum_{i=1}^{\frac{M}{2}} \int_{(i-1)\Delta}^{i\Delta} \left(x - \frac{2i-1}{2} \Delta \right)^2 \frac{1}{2 X_{\max}} dx = \frac{\Delta^2}{12}. \quad (1-19)$$

Se puede llegar al mismo resultado más fácilmente notando que el error de cuantización $q = x - Q(x)$ yace en el intervalo $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$. Como la entrada es uniforme, el error de cuantización es también uniforme sobre este intervalo. Entonces, el error de cuantización cuadrático medio es el segundo momento de una variable aleatoria uniformemente distribuida en el intervalo $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$:

$$\sigma_q^2 = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} q^2 dq = \frac{\Delta^2}{12}. \quad (1-20)$$

Ejemplo 18-b

En la Figura 8(d) se muestra un cuantizador uniforme para una fuente distribuida uniformemente, que utiliza un código de longitud fija. Este sistema se conoce como PCM (Pulse Code Modulation) binario.

$$M = 8, \quad R = \lceil \log_2 M \rceil = \lceil \log_2 8 \rceil = 3.$$

$$X_{\max} = 4, \quad \Delta = \frac{2 X_{\max}}{M} = \frac{8}{8} = 1.$$

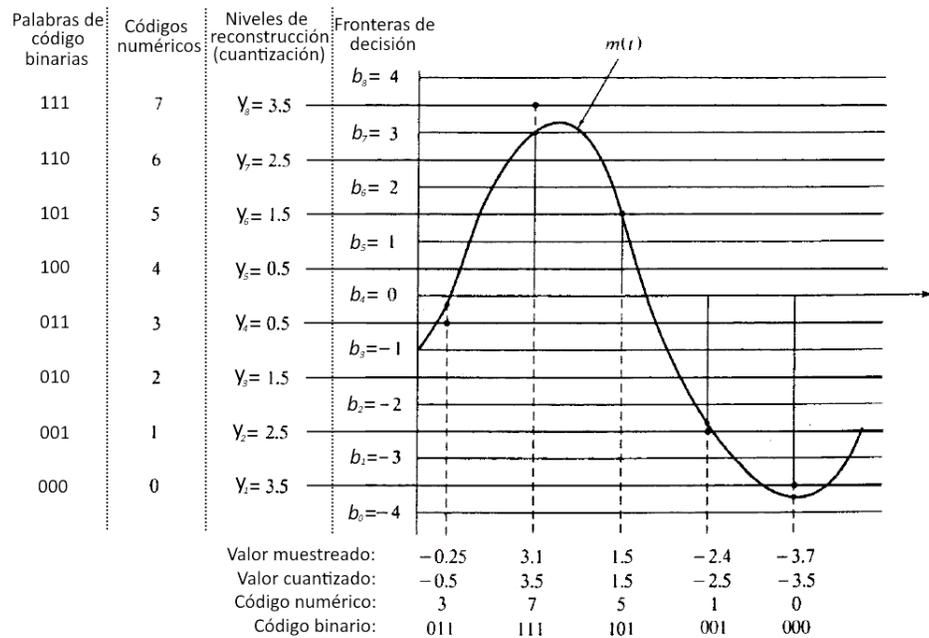


Figura 8(d). PCM binario (Adaptada de Hwei, 2003).

Aplicaciones de la cuantización escalar: Compresión de imágenes

Debido a la gran variedad de imágenes existentes, un modelo de probabilidad para los píxeles es difícil de obtener. Una aproximación es asumir que los valores de los píxeles están uniformemente distribuidos entre 0 y $2^b - 1$, donde b es el número de bits por píxel. Para la mayoría de las imágenes $b=8$, por tanto, se asume que los valores de los píxeles varían uniformemente entre 0 y 255. Se cuantizará la imagen de prueba de la Figura 9(a).

Si se usa solo un bit por píxel, se divide el rango $[0, 255]$ en 2 intervalos: $[0, 127]$ y $[128, 255]$. Los píxeles en el primer intervalo estarían representados por el valor 64 (punto medio) y los del segundo intervalo estarían representados por el valor 192 (punto medio). Es decir, los valores de frontera son $\{0, 128, 255\}$, y los valores de reconstrucción son $\{64, 192\}$. La imagen cuantizada se muestra en la Figura 9(b), en la que se han perdido casi todos los detalles.

Si se usa un cuantizador de 2 bits, con valores de frontera $\{0, 64, 128, 192, 255\}$ y niveles de reconstrucción $\{32, 96, 160, 224\}$, se obtiene la imagen cuantizada de la Figura 9(c) que contiene mayor detalle. Si se utilizan 3 bits por píxel, se incrementa el nivel de detalle, y la imagen reconstruida comienza a ser indistinguible de la original, al menos para un observador casual (ver Figura 9(d)) (Sayood, 2017).



Figura 9. Cuantizaciones imagen de prueba. a) imagen original; b) 1bit/píxel; c) 2bits/píxel; d) 3bits/píxel.

Ejercicio 7

Con base en los mapeos para un cuantizador de 3 bits mostrados en la Figura 8, complete la siguiente tabla de codificación (A/D) – reconstrucción (D/A) para las muestras generadas por la senoide $f(t) = 4 \cos(2\pi t)$ (Sayood, 2017).

t	$f(t)$	Salida codificador A/D	Salida reconstrucción D/A	Error
0.05				
0.10				
0.15				
0.20				

1.7.2 Codificación diferencial

“La codificación diferencial se basa en el hecho bien conocido de que los pixeles vecinos en una imagen (y las muestras adyacentes en sonido digitalizado) están correlacionados. Los valores correlacionados son generalmente similares, por lo que sus diferencias son pequeñas, y al codificar las diferencias:

$$d_n = x_n - x_{n-1} \quad (1-21)$$

en lugar de los valores reales se obtiene compresión, ya que se requieren menos bits para codificar las diferencias que para codificar los valores reales” (Salomon & Motta, 2010).

 En muchas fuentes de interés, la secuencia de salida muestreada $\{x_n\}$ no cambia significativamente de una muestra a la siguiente, por lo tanto, el rango dinámico y la varianza de la secuencia de diferencias $\{d_n = x_n - x_{n-1}\}$ son significativamente más pequeños que los de la secuencia de salida de la fuente. Dada la relación entre la varianza de la entrada del cuantizador y el error de cuantización generado ^[2], también es útil en términos de compresión con pérdida, buscar formas de codificar la diferencia de una muestra a la siguiente en lugar de codificar el valor real de la muestra (Sayood, 2017).

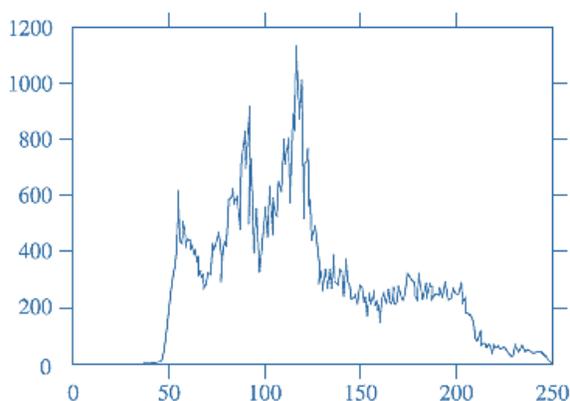
[2] Cuando se diseña un cuantizador para una fuente dada, el tamaño del intervalo de cuantización depende de la varianza de la entrada. Si se asume que la entrada está uniformemente distribuida, la varianza depende del rango dinámico de la entrada. A su vez, el tamaño del intervalo de cuantización determina la cantidad de ruido de cuantización.

Ejemplo 19

La Figura 10(a) es el histograma de la imagen de prueba de la Figura 9(a). Los valores de pixel varían sobre casi todo el rango de 0 a 255. Para representar estos valores exactamente se necesitan 8 bits por pixel. Para representarlos con pérdida con error en el bit menos significativo, se necesitan 7 bits por pixel.

La Figura 10(b) es el histograma de las diferencias. Más del 99% yace en el rango de -31 a 31. Si se está dispuesto a aceptar distorsión en el bit menos significativo, se necesitan 5 bits por pixel en lugar de 7 (Sayood, 2017).

(a)



(b)

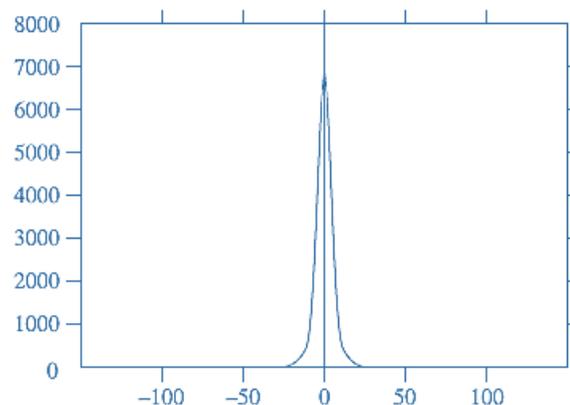


Figura 10. (a) Histograma imagen de ejemplo. (b) Histograma diferencias pixel-a-pixel imagen de ejemplo (Sayood, 2017).

El algoritmo básico

“Cualquier método adecuado, con o sin pérdidas, se puede utilizar para codificar las diferencias. En la práctica se suele utilizar la cuantización, dando como resultado una compresión con pérdida. La cantidad codificada no es la diferencia d_n sino un número similar cuantizado que se denota por \hat{d}_n :

$$\hat{d}_n = Q[d_n] = d_n + q_n, \quad (1-22)$$

donde q_n es el error de cuantización” (Salomon & Motta, 2010).

La compresión con pérdida de las diferencias genera acumulación de errores. En el receptor, la secuencia reconstruida $\{\hat{x}_n\}$ se obtiene mediante:

$$\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n. \quad (1-23)$$

Asumiendo que el transmisor y el receptor comienzan con el mismo valor x_0 ($\hat{x}_0 = x_0$), y siguiendo el proceso de cuantización y reconstrucción para las primeras muestras, se obtiene:

$$\begin{aligned} d_1 &= x_1 - x_0, & \hat{d}_1 &= Q[d_1] = d_1 + q_1 \\ \hat{x}_1 &= x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = \mathbf{x}_1 + \mathbf{q}_1 \\ d_2 &= x_2 - x_1, & \hat{d}_2 &= Q[d_2] = d_2 + q_2 \\ \hat{x}_2 &= \hat{x}_1 + \hat{d}_2 = x_1 + q_1 + d_2 + q_2 = \mathbf{x}_2 + \mathbf{q}_1 + \mathbf{q}_2. \end{aligned}$$

Continuando con este proceso, en la n -ésima iteración se obtiene:

$$\hat{x}_n = x_n + \sum_{k=1}^n q_k. \quad (1-24)$$

Se ve que el error de cuantización se acumula a medida que continúa el proceso. En teoría, si el proceso del error de cuantización tiene media 0, los errores se cancelarán entre sí a largo plazo. En la práctica, usualmente mucho antes de que eso pueda suceder, la precisión finita de las máquinas hace que el valor reconstruido se desborde (Sayood, 2017).

La solución a este problema surge al notar que el codificador y el decodificador están trabajando con diferentes piezas de información: el codificador genera la secuencia diferencia basado en los valores de muestra originales, mientras que el decodificador suma la diferencia cuantizada a una versión distorsionada de la señal original.

El problema se resuelve forzando al codificador y al decodificador a usar la misma información durante las operaciones de cálculo de la diferencia y de reconstrucción. La única información disponible para el receptor sobre la secuencia $\{x_n\}$ es la secuencia reconstruida $\{\hat{x}_n\}$. Como esta información también está disponible en el transmisor, se puede modificar la operación

de cálculo de la diferencia de modo que use el valor reconstruido de la muestra previa, en lugar de la propia muestra previa:

$$d_n = x_n - \hat{x}_{n-1}. \quad (1-25)$$

Usando esta nueva operación, y repitiendo el análisis de los procesos de cuantización y reconstrucción, se obtiene (de nuevo se asume que $\hat{x}_0 = x_0$):

$$\begin{aligned} d_1 &= x_1 - x_0, & \hat{d}_1 &= Q[d_1] = d_1 + q_1 \\ \hat{x}_1 &= x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1 \\ d_2 &= x_2 - \hat{x}_1, & \hat{d}_2 &= Q[d_2] = d_2 + q_2 \\ \hat{x}_2 &= \hat{x}_1 + \hat{d}_2 = \hat{x}_1 + d_2 + q_2 = x_2 + q_2. \end{aligned}$$

En la n -ésima iteración se obtiene:

$$\hat{x}_n = x_n + q_n \quad (1-26)$$

y no hay acumulación del ruido de cuantización, de hecho, el ruido de cuantización en la n -ésima secuencia reconstruida es el generado por la cuantización de la n -ésima diferencia. En la Figura 11 se muestra un diagrama de bloques para el sistema de codificación diferencial que se acaba de describir.

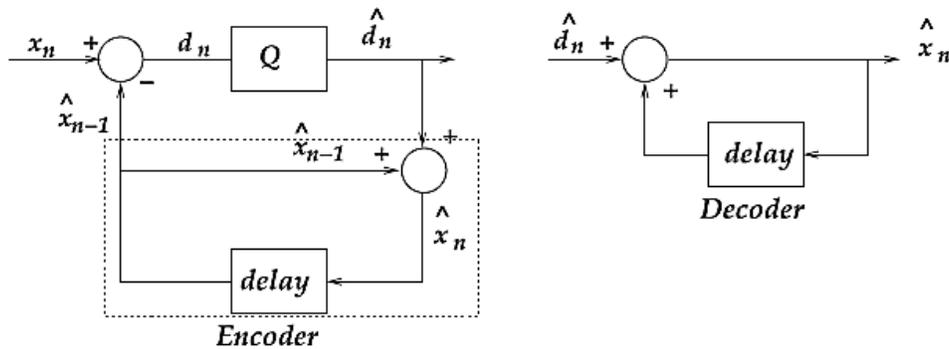


Figura 11. Sistema de codificación diferencial. Se dibujó un rectángulo punteado alrededor de la parte del codificador que imita al decodificador, para obtener una copia de la muestra reconstruida utilizada para generar la siguiente diferencia (Sayood, 2017).

Ejercicio 8

Considere la senoide mostrada en la Figura 12, que se ha muestreado a una tasa de 30 muestras por ciclo. Los parámetros para cuantizar la senoide con un cuantizador uniforme de 4 niveles, se muestran en la Tabla 11 (Sayood, 2017).

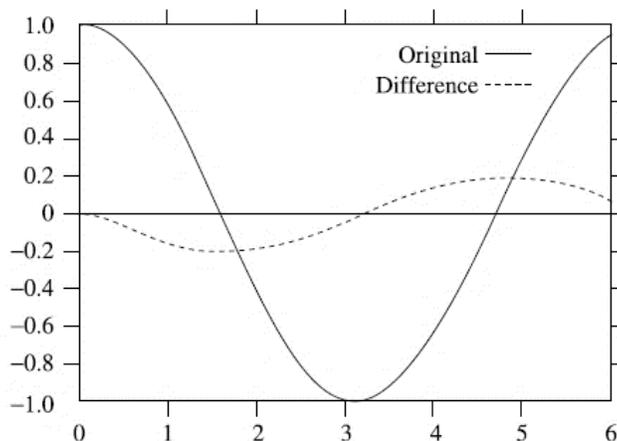


Figura 12. Onda sinusoidal y diferencias muestra a muestra.

Tabla 11. Parámetros para cuantizar la senoide con un cuantizador uniforme de 4 niveles.

	Rango dinámico de valores	Tamaño de paso para un cuantizador uniforme de 4 niveles	Rango del error de cuantización
Senoide	[-1 , 1]	0.5	[-0.25, 0.25]
Diferencias muestra a muestra (1-21)			
Diferencias muestra a muestra (1-25)			

- Llene las celdas de la tabla, para el caso en el que se toman las diferencias muestra a muestra (excluyendo la 1ra. muestra) de la senoide usando la ec. (1-21).
- Llene las celdas de la tabla, para el caso en el que se toman las diferencias muestra a muestra (excluyendo la 1ra. muestra) de la senoide usando la ec. (1-25).

- c. Las señales reconstruidas usando las aproximaciones de los literales a . y b . se muestran en la Figura 13. Identifique en la tabla cuál reconstrucción fue generada con cada una de las aproximaciones, que se muestran en la primera columna.

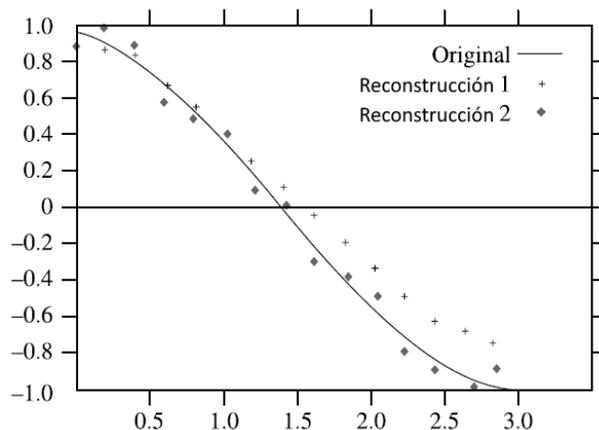


Figura 13. Reconstrucciones onda sinusoidal.

Aproximaciones cálculo de diferencias	Reconstrucción correspondiente (1 o 2)
$d_n = x_n - x_{n-1}$	
$d_n = x_n - \hat{x}_{n-1}$	

Modulación por codificación de pulsos diferencial

Se desea que los valores de las diferencias sean tan pequeños como sea posible. “Aprovechando que los ítems de datos que se están comprimiendo están correlacionados, es decir que un ítem x_i depende de varios de sus vecinos más próximos y no solo del ítem precedente x_{i-1} , puede obtenerse una mejor predicción (y como resultado diferencias más pequeñas) utilizando N de los vecinos previamente vistos para codificar el ítem actual x_i . Por lo tanto, se desea tener una función $p_i = f(\hat{x}_{i-1}, \hat{x}_{i-2}, \dots, \hat{x}_{i-N})$ para predecir x_i . Nótese que f tiene que ser una función de \hat{x}_{i-j} y no de x_{i-j} ya que el decodificador tiene que calcular la misma f . Cualquier método que utilice un predictor se denomina *modulación por codificación de pulsos diferencial o DPCM*” (Salomon & Motta, 2010).

Al reemplazar el bloque de retardo de la Figura 11 por un bloque predictor, se obtiene el sistema DPCM mostrado en la Figura 14.

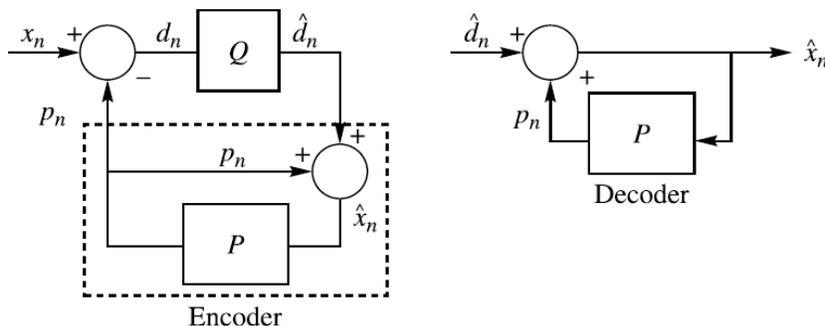


Figura 14. Sistema DPCM (Sayood, 2017).

Los sistemas de codificación diferencial como DPCM obtienen su ventaja por la reducción en la varianza y el rango dinámico de la secuencia de diferencias. Cuánto se reduce la varianza depende de qué tan bien el predictor puede predecir el siguiente símbolo basándose en los símbolos reconstruidos previamente.

En muchas aplicaciones se simplifica el problema restringiendo la función predictora a que sea lineal:

$$p_n = \sum_{i=1}^N w_i \hat{x}_{n-i} \quad (1-27)$$

donde N especifica el orden del predictor. Usando la suposición de cuantización fina ^[3], se puede escribir el problema de diseño del predictor así: hallar los $\{w_i\}$ que minimicen la varianza de la secuencia de diferencias, σ_d^2 :

$$\sigma_d^2 = E[(x_n - p_n)^2] = E \left[\left(x_n - \sum_{i=1}^N w_i x_{n-i} \right)^2 \right] \quad (1-28)$$

donde $E[\]$ es el operador valor esperado y se ha asumido que la secuencia de la fuente es una realización de un proceso estacionario en sentido amplio de valor real ^[4].

[3] Se supone que los tamaños de paso del cuantizador son tan pequeños que se puede reemplazar \hat{x}_n por x_n .

[4] La suposición de estacionariedad es una afirmación de que las características estadísticas del proceso bajo investigación no cambian con el tiempo. Por lo tanto, si se diseña un sistema para una entrada en función de las características estadísticas de la entrada hoy, el sistema seguirá siendo útil mañana porque la entrada no cambiará sus características.

Tomando la derivada de σ_d^2 con respecto a cada w_j e igualando a cero, se obtiene un sistema de N ecuaciones con N incógnitas:

$$\frac{\partial \sigma_d^2}{\partial w_j} = -2E \left[\left(x_n - \sum_{i=1}^N w_i x_{n-i} \right) x_{n-j} \right] = 0 \quad \text{para } j = 1, 2, \dots, N. \quad (1-29)$$

Tomando los valores esperados, se puede reescribir estas ecuaciones como

$$\sum_{i=1}^N w_i R_{xx}(i-j) = R_{xx}(j) \quad \text{para } j = 1, 2, \dots, N, \quad (1-30)$$

donde R_{xx} es la función de autocorrelación de x_n : $R_{xx}(k) = E[x_n x_{n+k}]$.

Estas ecuaciones se pueden escribir en forma matricial como:

$$\mathbf{Rw} = \mathbf{p}, \quad (1-31)$$

donde

$$\mathbf{R} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \cdots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(N-2) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(N-3) \\ \vdots & \vdots & & & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & R_{xx}(N-3) & \cdots & R_{xx}(0) \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_N \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ R_{xx}(3) \\ \vdots \\ R_{xx}(N) \end{bmatrix}$$

y donde se ha usado el hecho de que $R_{xx}(-k) = R_{xx}(k)$ para un proceso estacionario en sentido amplio de valor real. Estas ecuaciones se conocen como la forma discreta de las ecuaciones de Wiener-Hopf. Si se conocen los valores de autocorrelación $\{R_{xx}(k)\}$ para $k = 0, 1, \dots, N$, se pueden calcular los coeficientes del predictor como (Sayood, 2017):

$$\mathbf{w} = \mathbf{R}^{-1} \mathbf{p}. \quad (1-31a)$$

Aplicaciones de la codificación diferencial

Para la secuencia de voz de la Figura 15, hallar los predictores de orden 1, 2 y 3 y examinar su desempeño (Sayood, 2017).

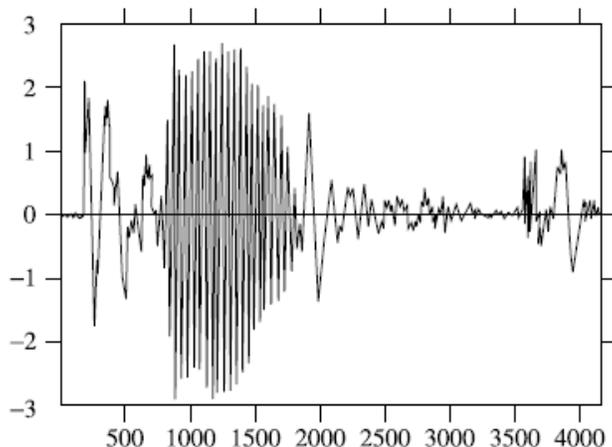


Figura 15. Segmento de voz. Un hombre pronunciando la palabra “test”.

- Se estiman los valores de correlación a partir de los datos. Dados M puntos de datos, se usa el siguiente promedio para hallar el valor de $R_{xx}(k)$:

$$R_{xx}(k) = \frac{1}{M-k} \sum_{i=1}^{M-k} x_i x_{i+k}.$$

Usando estos valores de autocorrelación, se obtienen los coeficientes para los 3 predictores diferentes. Estos coeficientes se usan para obtener la secuencia residual. Para ver la reducción en la varianza, se calcula el cociente entre la varianza de la salida de la fuente y la varianza de la secuencia residual. Los resultados se muestran en la Tabla 12.

Tabla 12. Resultados del cálculo para examinar el desempeño de los predictores de orden 1, 2 y 3.

Orden N del predictor	Coefficientes del predictor	Varianza salida fuente / Varianza secuencia residual	Tamaño de paso cuantizador Laplaciano de 8 niveles	SNR (dB)
N=1	$w_1 = 0.66$	2.04	0.3	3.87
N=2	$w_1 = 0.596$ $w_2 = 0.096$	3.37	0.4	9.81
N=3	$w_1 = 0.577$ $w_2 = -0.025$ $w_3 = 0.204$	6.28	0.5	10.16

- Ahora se introduce un cuantizador uniforme y se estudia el rendimiento del sistema DPCM. Al observar el histograma de la secuencia residual se encuentra que es altamente puntiagudo, por lo tanto, se asume que la entrada al cuantizador es Laplaciana [5]. Se ajusta el tamaño del paso del cuantizador con base en la varianza del residual, el tamaño óptimo se obtiene multiplicando el tamaño para una varianza de 1 por la desviación estándar de la entrada. Los tamaños de paso que se obtienen para un cuantizador Laplaciano de 8 niveles usando esta aproximación, se muestran en la Tabla 12.
- Se mide el desempeño del sistema DPCM usando la razón señal a ruido (SNR):

$$SNR(dB) = \frac{\sum_{i=1}^M x_i^2}{\sum_{i=1}^M (x_i - \hat{x}_i)^2}. \quad (1-32)$$

Los resultados se encuentran en la Tabla 12.

- La señal de voz reconstruida usando un predictor de orden 3 y un cuantizador de 8 niveles se muestra en la Figura 16:

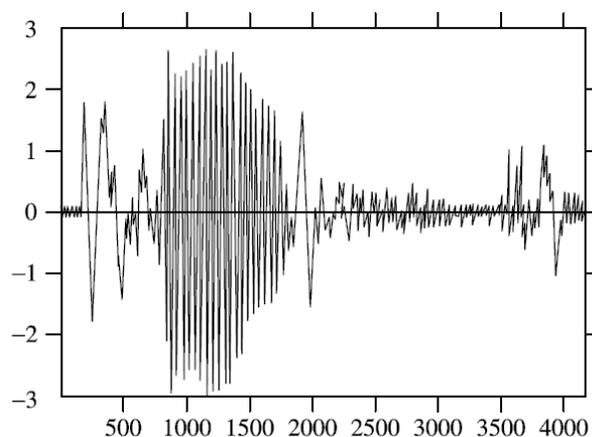


Figura 16. Secuencia reconstruida usando un predictor de 3er. orden y un cuantizador uniforme de 8 niveles. La secuencia reconstruida luce como la original, pero hay distorsión significativa en áreas donde los valores de salida de la fuente son pequeños, debido a que en estas regiones la entrada al cuantizador es cercana a cero. Como el cuantizador no tiene un nivel de salida "0", su salida cambia entre los dos niveles más internos. Si se escucha esta señal, se oiría un silbido en la señal reconstruida.

[5] La función de densidad para una variable aleatoria de media cero con distribución Laplaciana y varianza σ^2 es (Sayood, 2017):

$$f_X(x) = \frac{1}{\sqrt{2\sigma^2}} \exp \frac{-\sqrt{2}|x|}{\sigma}$$

1.7.3 Codificación de transformación

La idea de la codificación de transformación es transformar la entrada en una forma diferente que se pueda comprimir mejor, o para la que se puedan eliminar más fácilmente ciertos términos sin tanta pérdida cualitativa en la salida. Para el propósito de la compresión, las propiedades deseables de una transformada son (Blelloch, 2013):

1. Que haga que muchos de los coeficientes transformados sean pequeños.
2. Que haga que desde el punto de vista de la percepción, algunos de los términos sean más importantes que otros.

Transformada discreta del coseno

La transformada discreta del coseno DCT (Discrete Cosine Transform) es una de las transformadas más utilizadas en la práctica para la compresión de imágenes, más que la transformada discreta de Fourier DFT (Discrete Fourier Transform) ya que esta última asume periodicidad, lo que no es necesariamente cierto en imágenes (Blelloch, 2013).

La DCT bidimensional se define así (Salomon & Motta, 2010):

$$G_{ij} = \sqrt{\frac{2}{m}} \sqrt{\frac{2}{n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} \text{pixel}_{xy} \cos \left[\frac{(2y+1)j\pi}{2m} \right] \cos \left[\frac{(2x+1)i\pi}{2n} \right], \quad (1-33)$$

donde $C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$ para $f = 0, 1, \dots, n-1$ y $\begin{matrix} 0 \leq i \leq n-1 \\ 0 \leq j \leq m-1 \end{matrix}$.

El primer coeficiente G_{00} se denomina *coeficiente DC*, y el resto de los coeficientes reciben el nombre de *coeficientes AC* [6].

La DCT permite calcular los componentes de frecuencia de una matriz, de tal manera que se obtienen el coeficiente DC (valor promedio ponderado de los píxeles) y los coeficientes AC (diferencia entre el valor ponderado del píxel y el valor promedio).

[6] Los términos DC y AC se han heredado de la ingeniería eléctrica, donde se utilizan para *corriente continua* (Direct Current) y *corriente alterna* (Alternating Current).

La DCT inversa (IDCT) bidimensional está dada por:

$$pixel_{xy} = \sqrt{\frac{2}{m}} \sqrt{\frac{2}{n}} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} C_i C_j G_{ij} \cos \left[\frac{(2y+1)j\pi}{2m} \right] \cos \left[\frac{(2x+1)i\pi}{2n} \right], \quad (1-33a)$$

$$\text{donde } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases} \quad \text{para } f = 0, 1, \dots, n-1 \quad \text{y} \quad \begin{cases} 0 \leq x \leq n-1 \\ 0 \leq y \leq m-1 \end{cases}$$

“La característica más importante de la DCT, que hace que sea tan útil en la compresión de datos, es que toma datos correlacionados de la entrada y concentra su energía en solo unos pocos (los primeros) coeficientes de transformada [7]. Por lo tanto, la mayoría de los coeficientes de transformada producidos por la DCT son cero o números pequeños y solo unos pocos son grandes (por lo general los primeros). Los primeros coeficientes contienen la información importante de la imagen (frecuencias bajas) y los posteriores la menos importante (frecuencias altas, detalles de la imagen)” (Salomon & Motta, 2010).

Aplicación a la compresión de imágenes - JPEG

El formato JPEG (acrónimo de *Joint Photographic Experts Group*) es un buen ejemplo de compresión del mundo real porque:

- a. Se utiliza muy ampliamente en la práctica, y
- b. Usa muchas de las técnicas de compresión vistas, incluyendo códigos de Huffman, códigos aritméticos, codificación diferencial, codificación RLE, cuantización escalar y codificación de transformación.



JPEG se usa para imágenes fijas y es el estándar utilizado en la WEB para imágenes fotográficas. JPEG es un método de compresión con pérdidas para imágenes en color y en escala de grises. Fue diseñado para ser utilizado con material fotográfico y obras de arte naturalistas. No es el formato ideal para imágenes textuales u otras imágenes con grandes áreas de color sólido o un número muy limitado de colores distintos. Funciona mejor con imágenes de tonos continuos, donde los pixeles adyacentes tienen colores similares.

[7] La compresión de imágenes se basa en la correlación bidimensional de pixeles: un pixel tiende a parecerse a todos sus vecinos más próximos, no solo a los de su fila.

JPEG está diseñado para que el usuario pueda ajustar el factor de pérdida para compensar el tamaño (compresión) y la calidad de la imagen, y está diseñado para que la pérdida tenga el menor efecto en la percepción humana. Utiliza muchos parámetros, permitiendo a los usuarios experimentar y lograr la compensación deseada en la relación compresión/calidad (Blelloch, 2013; Salomon & Motta, 2010).

En la Figura 17 se proporciona una descripción general del proceso de compresión JPEG. A continuación, se cubrirán cada una de las etapas en este proceso.

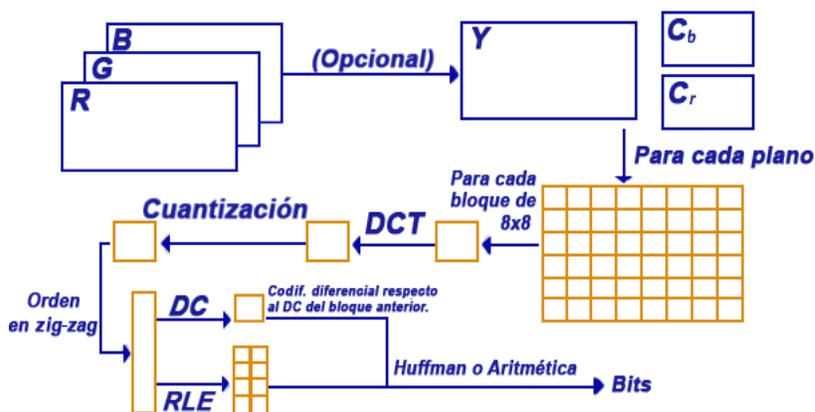


Figura 17. Proceso de compresión JPEG.

- Conversión RGB a YCbCr

La entrada a JPEG son 3 planos de color de 8 bits por píxel cada uno, representando el rojo, el verde y el azul (RGB). El primer paso (opcional) de la compresión JPEG es convertir estos planos en planos de color YCbCr, los cuales están diseñados para representar mejor la percepción humana (este paso se omite para imágenes en escala de grises). El plano Y representa el brillo (*luminancia*) de la imagen, y los componentes Cb y Cr representan la tonalidad del color (*crominancia*). La razón para convertir a YCbCr es que, en términos de percepción, es más importante acertar con la luminancia que con la tonalidad ^[8] (Blelloch, 2013).

La transformación de RGB a YCbCr se realiza con (Salomon & Motta, 2010):

$$\begin{aligned} Y &= (77/256) R + (150/256) G + (29/256) B, \\ C_b &= -(44/256) R + (87/256) G + (131/256) B + 128, \\ C_r &= (131/256) R + (110/256) G + (21/256) B + 128. \end{aligned} \quad (1-34)$$

[8] "El ojo es sensible a pequeños cambios en la luminancia pero no en la crominancia, por lo que la parte de crominancia puede posteriormente perder muchos datos y, por lo tanto, ser altamente comprimida, sin perjudicar globalmente la calidad de la imagen en demasía. Sin transformar a YCbCr, ninguno de los 3 componentes de color RGB tolerará muchas pérdidas, dando lugar a una peor compresión" (Salomon & Motta, 2010).

Estas expresiones son promedios ponderados de rojo, verde y azul (RGB). Los pesos no están balanceados ya que el ojo humano responde más al verde que al rojo, y más al rojo que al azul.

- **Submuestreo $YCbCr$**

JPEG mantiene todos los píxeles para la luminancia, pero normalmente submuestra los dos planos de color por un factor de 2 en cada dimensión, para un factor total de 4. Este es el primer componente con pérdidas de JPEG y genera un factor de compresión de 2 (Blelloch, 2013):

$$RC_{YCbCr} = \frac{1(Y) + 1/4(Cb) + 1/4(Cr)}{1(Y) + 1(Cb) + 1(Cr)} = \frac{(1 + 2(0.25))}{3} = 0.5 \rightarrow FC = 2.$$

“Puesto que el componente de luminancia no se modifica, no hay ninguna pérdida apreciable en la calidad de la imagen. Las imágenes en escala de grises no pasan por este proceso” (Salomon & Motta, 2010).

- **Aplicación de la DCT**

El siguiente paso del algoritmo JPEG es particionar cada uno de los planos de color en bloques de 8x8 píxeles. A continuación, cada uno de estos bloques se codifica por separado. El primer paso para codificar un bloque es aplicar la DCT bidimensional, esto retorna un bloque de 8x8 términos de frecuencia. Hasta aquí, no se introduce pérdida ni compresión (Blelloch, 2013).

Las razones para el tamaño del bloque (8x8) son (Salomon & Motta, 2010):

1. “La aplicación de la DCT a bloques grandes implica muchas operaciones aritméticas y es por lo tanto lenta. La aplicación de la DCT a bloques pequeños es rápida”.
2. “La experiencia muestra que, en una imagen de tonos continuos, las correlaciones entre los píxeles son de corto alcance; es decir, un píxel tiene un valor (componente de color o tono de gris) que es cercano al de sus vecinos más próximos, pero no al de sus vecinos lejanos”.

Por lo tanto, la DCT para JPEG se puede calcular a partir de la ecuación (1-33) con $m=n=8$:

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^7 \sum_{y=0}^7 pixel_{xy} \cos \left[\frac{(2x+1)i\pi}{16} \right] \cos \left[\frac{(2y+1)j\pi}{16} \right], \quad (1-35)$$

donde $C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$ y $0 \leq i, j \leq 7$.

- Cuantización escalar

Después de la DCT, el siguiente paso aplicado a los bloques es utilizar cuantización escalar uniforme sobre cada uno de los términos de frecuencia. Esta cuantización es controlable mediante parámetros del usuario y es la principal fuente de pérdida de información en la compresión JPEG. Dado que el ojo humano es más perceptivo a ciertos componentes de frecuencia que a otros, JPEG permite que el factor de escalado de la cuantización sea diferente para cada componente de frecuencia. Los factores de escalado se especifican mediante una tabla de 8x8 que se usa para dividir sabiamente la tabla 8x8 de componentes de frecuencia (Blelloch, 2013).

JPEG define las tablas de cuantización estándar para los componentes Y y CbCr. Estas tablas se muestran en la

Figura 18 (Salomon & Motta, 2010):

16 11 10 16 24 40 51 61	17 18 24 47 99 99 99 99
12 12 14 19 25 58 60 55	18 21 26 66 99 99 99 99
14 13 16 24 40 57 69 56	24 26 56 99 99 99 99 99
14 17 22 29 51 87 80 62	47 66 99 99 99 99 99 99
18 22 37 56 68 109 103 77	99 99 99 99 99 99 99 99
24 35 55 64 81 104 113 92	99 99 99 99 99 99 99 99
49 64 78 87 103 121 120 101	99 99 99 99 99 99 99 99
72 92 95 98 112 100 103 99	99 99 99 99 99 99 99 99
Luminancia	Crominancia

Figura 18. Tablas de cuantización estándar JPEG.

 En estas tablas los componentes más grandes están hacia la esquina inferior derecha, debido a que estos escalarán los componentes de frecuencias más altas a los que los humanos son menos sensibles. La selección de los números particulares en las tablas de cuantización se basa en estudios sobre la percepción humana (Blelloch, 2013).

“Cada número en la matriz de coeficientes de la DCT se divide entre el número correspondiente de la tabla de cuantización particular utilizada, y el resultado se redondea al entero más cercano. Si la cuantización se hace correctamente, quedarán muy pocos números distintos de cero en la matriz resultante, y típicamente estarán concentrados en la región superior izquierda” (Salomon & Motta, 2010).

- **Codificación**

JPEG comprime de forma separada los componentes DC (esquina superior izquierda) y los componentes AC (Blelloch, 2013):

- a. *Componentes DC*: utiliza una codificación diferencial, restando el valor del componente DC del bloque anterior al valor del componente DC del bloque actual. Luego realiza una codificación Huffman o aritmética de esta diferencia. La motivación para este método es que el componente DC es a menudo similar en bloques consecutivos, por lo que codificar la diferencia generará una mejor compresión.
- b. *Componentes AC*: los componentes AC de un bloque se comprimen convirtiéndolos primero a un orden lineal, recorriendo la tabla de frecuencias en *zig-zag* (ver Figura 19). La motivación para esto es que en el orden lineal las frecuencias de longitudes similares estarán cerca unas de otras. En particular, la mayoría de los ceros aparecerán como un gran bloque contiguo al final del orden lineal. Se utiliza una forma de codificación RLE para comprimir el orden lineal. Se codifica como una secuencia de parejas (*salto, valor*), donde *salto* es el número de ceros antes de un *valor*. La pareja especial (0,0) especifica el final del bloque. La secuencia de parejas es entonces comprimida usando codificación aritmética o Huffman.

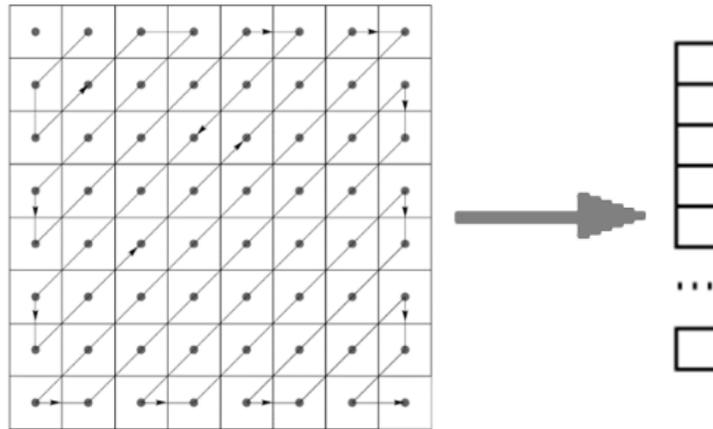


Figura 19. Escaneo en zig-zag del bloque JPEG (Blelloch, 2013).

- Descompresión JPEG

El decodificador revierte uno a uno los pasos de la compresión, por lo tanto, JPEG es un método de compresión *simétrico*. En particular, el decodificador JPEG funciona mediante el cálculo de la DCT Inversa (IDCT) que se obtiene a partir de la ecuación (1-33a) con $m=n=8$:

$$pixel_{xy} = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C_i C_j G_{ij} \cos \left[\frac{(2y+1)j\pi}{16} \right] \cos \left[\frac{(2x+1)i\pi}{16} \right], \quad (1-35a)$$

donde $C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$ y $0 \leq x, y \leq 7$.

Ejemplo 20

“Rendimiento de la DCT cuando se aplica a una imagen de tonos discretos. Se comienza con el patrón en forma de Y de la Figura 20(a), típico de una imagen de tonos discretos. Los 64 coeficientes de la DCT de este patrón se muestran en la Figura 20(b). La Figura 20(c) muestra los coeficientes después de haber sido cuantizados ligeramente (solo se han truncado al entero más cercano). Los resultados tras aplicar la IDCT a estos coeficientes cuantizados, se muestran en la Figura 20(d); es fácil ver que la reconstrucción del patrón original es deficiente. La conclusión es que la DCT es poco eficiente cuando se aplica a una imagen de tonos discretos” (Salomon & Motta, 2010).

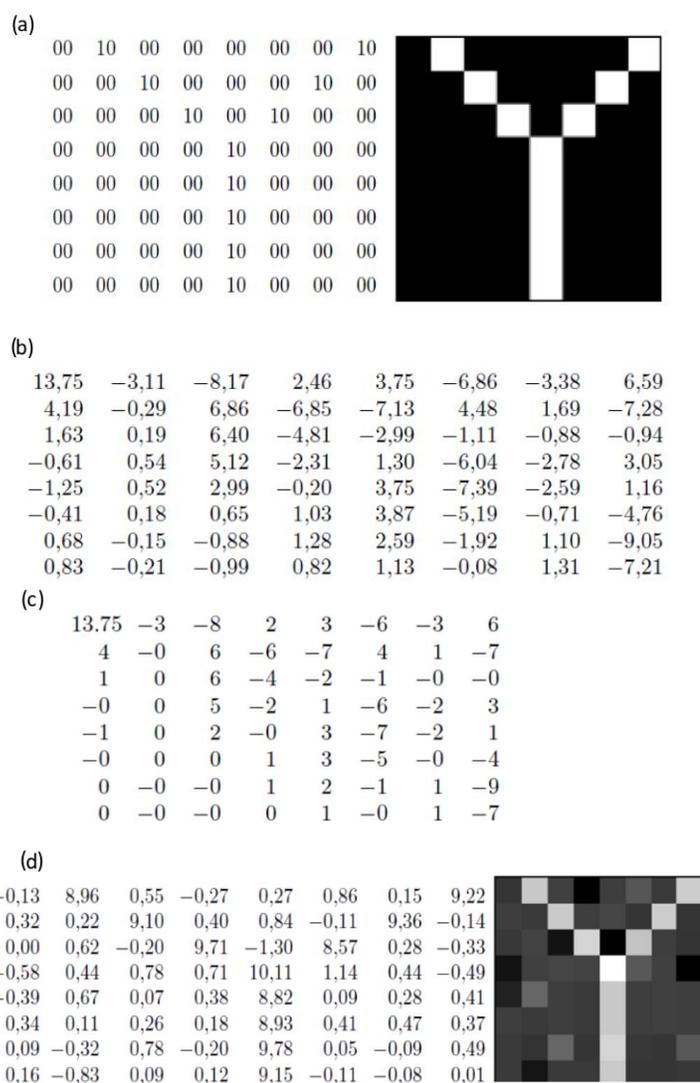
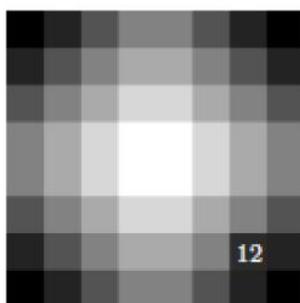


Figura 20. (a) Una imagen de tonos discretos. (b) Sus coeficientes de la DCT. (c) Cuantización ligera truncando a enteros. (d) La IDCT, resultados deficientes.

Ejercicio 9

“Rendimiento de la DCT cuando se aplica a una imagen de tonos continuos. Se comienza con el patrón altamente correlacionado de la Figura 21(a), es un ejemplo idealizado de una imagen de tonos continuos, ya que los pixeles adyacentes difieren en una cantidad constante, excepto el pixel de la fila 7, columna 7 (subrayado). Los 64 coeficientes de la DCT de este patrón se muestran en la Figura 21(b), es evidente que solo hay unos pocos coeficientes dominantes. La Figura 21(c) muestra los coeficientes después de haber sido toscamente cuantizados, por lo que solo permanecen 4 coeficientes no nulos. Halle los resultados de aplicar la IDCT a estos coeficientes cuantizados” (Salomon & Motta, 2010).

(a)



(b)

239	1,19	-89,76	-0,28	1,00	-1,39	-5,03	-0,79
1,18	-1,39	0,64	0,32	-1,18	1,63	-1,54	0,92
-89,76	0,64	-0,29	-0,15	0,54	-0,75	0,71	-0,43
-0,28	0,32	-0,15	-0,08	0,28	-0,38	0,36	-0,22
1,00	-1,18	0,54	0,28	-1,00	1,39	-1,31	0,79
-1,39	1,63	-0,75	-0,38	1,39	-1,92	1,81	-1,09
-5,03	-1,54	0,71	0,36	-1,31	1,81	-1,71	1,03
-0,79	0,92	-0,43	-0,22	0,79	-1,09	1,03	-0,62

(c)

239	1	-90	0	0	0	0	0
0	0	0	0	0	0	0	0
-90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figura 21. (a) Un patrón de tonos continuos. (b) Sus coeficientes de la DCT. (c) Cuantización profunda, solo 4 coeficientes distintos de 0.

1.7.4 Esquemas de análisis/síntesis

“La esencia de la compresión de datos consiste en entender y modelar la fuente de información, y luego codificar los datos teniendo como referencia el modelo de la fuente. Esto es exactamente lo que hacen los esquemas de análisis/síntesis. Estos esquemas se basan en la disponibilidad de un modelo paramétrico para la generación de salida de la fuente. Cuando tal modelo existe, el transmisor analiza la salida de la fuente y extrae los parámetros del modelo, los cuales son transmitidos al receptor. El receptor utiliza el modelo, junto con los parámetros transmitidos, para sintetizar una aproximación a la salida de la fuente” (Vargas & Cedeño, 2022).

La compresión fractal de imágenes encaja en estos esquemas, ya que también utiliza la idea de representar los datos con un proceso para generarlos, y el receptor regenera la salida de la fuente utilizando las “instrucciones” del transmisor; es decir, lo que se almacena o transmite no son las muestras de la salida de la fuente, sino un método para sintetizar la salida (Sayood, 2017).

Compresión fractal de imágenes

“Los fractales han sido populares desde la década de los 70’s y tienen muchas aplicaciones. Una de estas aplicaciones, relativamente poco utilizada, es la compresión de datos. La aplicación de los fractales en la compresión de imágenes se efectúa mediante *sistemas de funciones iteradas* (IFS).

El principio de la compresión fractal de imágenes es considerar que, dado que la forma que se quiere representar es una imagen, lo que se busca es deducir el conjunto de funciones, llamado código IFS, que por iteraciones reconstruya esta imagen. Existen sistemas en la actualidad que aplican esta técnica y que pueden comprimir una imagen hasta 10.000 a 1; estos niveles de compresión que no se logran utilizando otra técnica, son posibles porque básicamente se considera comprimida la imagen en la descripción de las funciones que la generan, esto es lo que se almacena, y para recuperar la imagen simplemente se iteran las funciones y así se reconstruye la imagen almacenada” (Vargas & Cedeño, 2022).

- Transformaciones afines contractivas

Las *transformaciones afines* se pueden describir como combinaciones de rotaciones, escalados, esfuerzos cortantes y traslaciones de los ejes coordenados en un espacio n -dimensional. Para los propósitos de este trabajo se considerarán solo transformaciones afines en \mathbb{R}^2 .

Una transformación afín en \mathbb{R}^2 está compuesta por una transformación lineal y una traslación. La forma general para una transformación afín es:

$$w \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \quad (1-36)$$

donde los coeficientes a, b, c, d, e y f son números reales. La transformación lineal está representada por la matriz que se muestra a continuación:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad (1-37)$$

con la cual se puede realizar escalado en cada eje, rotación, esfuerzo cortante en el eje x y en el eje y de la imagen. La traslación desplaza el origen a las coordenadas (e, f) (ver Figura 22).

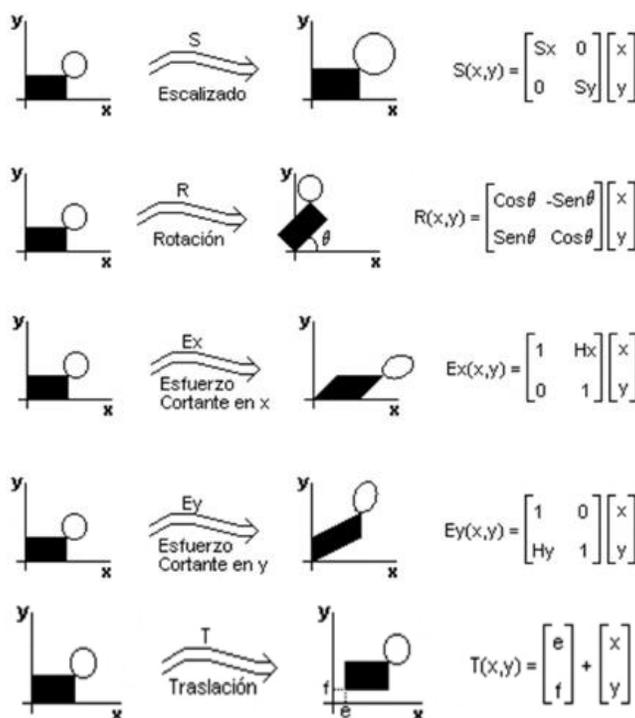


Figura 22. Transformaciones afines en \mathbb{R}^2 . Tomada de (Hernández & Niño, 1992).

Un ejemplo en dos dimensiones de una transformación afín es:

$$w(x, y) = \left(\frac{1}{2}x + \frac{1}{4}y + 1, \frac{1}{4}x + \frac{1}{2}y + 2 \right).$$

Esta transformación mueve el punto $(0,0)$ a $(1,2)$ y mueve $(-1,0.5)$ a $(0.625,2)$. Se nota esta transformación con w . $w(S)$ denota la imagen por w del conjunto de puntos S .

Es fácil ver que las transformaciones descritas anteriormente pueden cambiar longitudes y ángulos. El escalado cambia las longitudes de los objetos, la rotación y los esfuerzos cortantes cambian los ángulos. Sin embargo, una propiedad que se conserva es el paralelismo de las líneas: un par de líneas paralelas permanecerán así después de aplicar cualquiera de estas transformaciones. Una transformación que conserva el paralelismo se llama *afín*.

En la Figura 23 se muestra qué le hace w a una imagen de una cara sonriente F , que yace en el plano xy . El resultado es una nueva cara comprimida $w(F)$. La transformación afín deformó y trasladó la cara. Nótese que los ojos en la cara transformada $w(F)$ están más cerca que en la imagen F . Se dice entonces que la transformación es *contractiva* porque ella siempre mueve los puntos de tal forma que se encuentren más cerca.

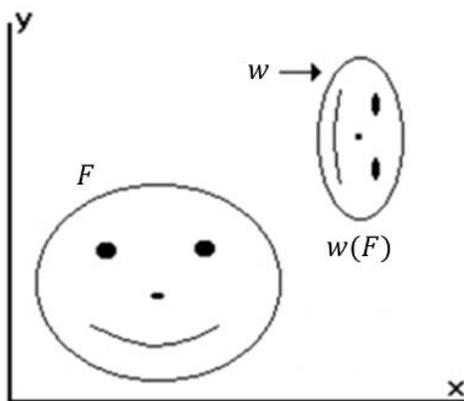


Figura 23. Transformación afín contractiva.

La conclusión clave de esta sección es que cualquier transformación afín bidimensional puede ser completamente especificada por solo seis números (Vargas & Cedeño, 2022).

- Sistemas de funciones iteradas (IFS)

Un sistema de funciones iteradas (IFS) es una colección de funciones que se aplican iterativamente sobre un conjunto inicial. Para propósitos del presente trabajo se consideran IFS de la forma

$$\{w_i: i = 1, 2, \dots, N\},$$

donde las w_i son transformaciones afines contractivas en \mathbb{R}^2 . La descripción de un IFS se llama un *código IFS*, el cual puede ser almacenado en una tabla que contiene los coeficientes de las transformaciones.

Ejemplo 21

El siguiente es un ejemplo de un IFS en \mathbb{R}^2 compuesto de tres transformaciones afines contractivas y su correspondiente código IFS (ver Tabla 13) (Salomon & Motta, 2010):

$$\begin{aligned} w_1 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ w_2 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}, \\ w_3 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 0 \end{bmatrix}. \end{aligned}$$

Tabla 13. Código IFS de transformaciones.

w	a	b	c	d	e	f
w_1 :	1/2	0	0	1/2	0	0
w_2 :	1/2	0	0	1/2	0	1/2
w_3 :	1/2	0	0	1/2	1/2	0

“Las tres transformaciones reducen una imagen a la mitad de su tamaño original; además, w_2 y w_3 colocan dos copias de la imagen en los desplazamientos relativos $(0, 1/2)$ y $(1/2, 0)$, como se muestra en la Figura 24.

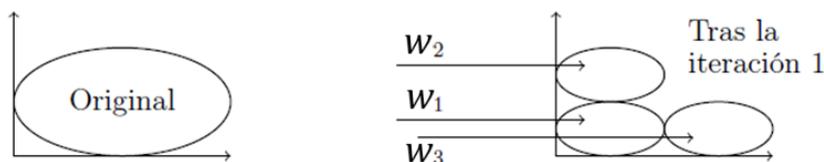


Figura 24. Transformaciones afines generadas por el IFS de la Tabla 13.

La Figura 25 muestra la aplicación del IFS de ejemplo a tres imágenes iniciales distintas: un cuadrado, un círculo y una Z. Cuando el IFS se aplica algunas veces a una imagen, todavía es posible discernir las copias individuales de la imagen original. Sin embargo, cuando se aplica varias veces, el resultado es el triángulo de Sierpinski, por lo que el resultado final no depende de la forma de la imagen inicial. La imagen final obtenida en el límite, después de aplicar un IFS un número infinito de veces, se llama *atractor del IFS*” (Salomon & Motta, 2010).

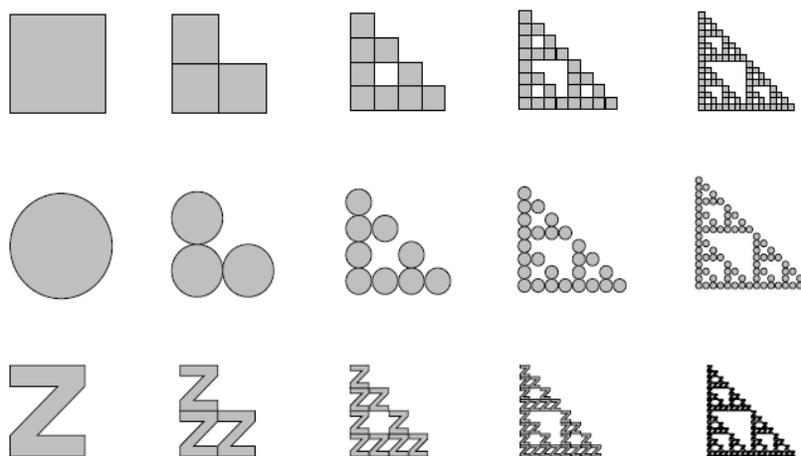


Figura 25. Aplicación del IFS de ejemplo a tres imágenes iniciales distintas (Salomon & Motta, 2010).

El resultado de cada iteración del IFS es una imagen que contiene todas las imágenes de todas las iteraciones anteriores. Si se aplica el mismo IFS muchas veces, es posible ampliar el resultado para magnificarlo, y todavía ver los detalles de las imágenes originales. En principio, si se aplica el IFS un número infinito de veces, el resultado final mostrará los detalles a cualquier aumento, será un *fractal*. Los objetos fractales con frecuencia son estadísticamente *autosimilares* o invariantes con el cambio de escala, es decir que al hacer ampliaciones de un objeto fractal se obtienen objetos tan complejos como el original (partes del objeto son idénticas a todo el objeto).

Matemáticamente, el efecto conjunto de todas las transformaciones que conforman el IFS se puede representar como la transformación $W(\cdot)$, que puede verse como la unión de las transformaciones w_i :

$$W = \bigcup_i w_i. \quad (1-38)$$

Entonces la transformación W tiene un único punto fijo A tal que:

$$A = W(A) = \bigcup_{i=1}^N w_i(A) \quad \text{y} \quad A = \lim_{n \rightarrow \infty} W^n(I_0) \quad (1-39)$$

para toda imagen inicial I_0 . El punto fijo A es el atractor del IFS o fractal determinístico generado por el IFS (Vargas & Cedeño, 2022).

- Codificación IFS

“Lo anterior sugiere una posible manera de comprimir imágenes reales: si se logra dividir una imagen en partes de tal manera que cada parte sea idéntica (o al menos muy cercana) a la imagen completa, entonces se puede comprimir la imagen mediante un IFS, averiguando las transformaciones que transforman la imagen completa en cada parte. La transformación para cada parte se expresa mediante unos pocos números, que se convierten en el *stream* comprimido” (Vargas & Cedeño, 2022).

Lo anterior está formalizado en el *Teorema del Collage de Barnsley* (Barnsley, 2012), que establece que para encontrar un IFS cuyo atractor es “cercano” a una imagen I según cierta métrica (esto es su atractor se “ve” como I), se debe encontrar un conjunto de transformaciones contractivas tal que la unión (*collage*) de las imágenes de I bajo las transformaciones sea cercana a I . En la Figura 26 se ve un collage de una hoja de olmo, se tienen cinco transformaciones afines de la hoja de olmo que unidas se ven como la hoja original, entonces con esas cinco transformaciones se puede almacenar la forma de la hoja de olmo.



Figura 26. Hoja de olmo y collage.

El codificador funcionará de la siguiente manera (Blelloch, 2013):

1. Establecer la imagen a comprimir I .
2. Calcular las transformaciones $\{w_i: i = 1, 2, \dots, N\}$ tal que $I \approx W(I) = \bigcup_{i=1}^N w_i(I)$.
3. Transmitir (o almacenar) los coeficientes que identifican el código IFS correspondiente.

- Decodificación IFS

a) Algoritmo de iteración determinística

El decodificador usará los coeficientes del código IFS para reensamblar W y reconstruir su punto fijo, la imagen (el atractor A o fractal generado por el IFS) a partir de una imagen arbitraria I_0 (Hernández & Niño, 1992):

1. $A = I_0$
 2. Hacer
 - $T = \emptyset$
 - Para i desde 1 hasta N :

$$T = T \cup w_i(A)$$
 - $A = T$
 - Despliegue A
- Mientras (no se interrumpa)

“Este algoritmo parte de una imagen arbitraria I_0 y aplica a esta imagen cada una de las transformaciones w_i , uniendo tales imágenes en una sola imagen llamada T ; luego repite el mismo proceso sobre T en cada iteración. Este algoritmo es lento debido a que se calcula la imagen de cada punto para cada transformación, en cada iteración. Además, se debe disponer de memoria para almacenar dos imágenes, la que se tiene en cada momento y la que se está componiendo en la respectiva iteración” (Vargas & Cedeño, 2022).

b) Algoritmo de iteración aleatoria (Chaos Game)

Un IFS con probabilidad es un IFS $\{w_i: i = 1, 2, \dots, N\}$ en el cual se asigna a cada transformación contractiva w_i una probabilidad de aplicación p_i , tal que:

$$\sum_{i=1}^N p_i = 1, \quad p_i > 0 \quad i = 1, 2, 3, \dots, N. \quad (1-40)$$

Un IFS con probabilidad se nota

$$\{w_i: i = 1, 2, \dots, N; p_i: i = 1, 2, \dots, N\}.$$

Partiendo de un punto arbitrario x_0 , el algoritmo de generación mediante iteración aleatoria es (Hernández & Niño, 1992):

1. $x = x_0$
2. Hacer
 - Seleccionar un número aleatorio k en $\{1, 2, 3, \dots, N\}$, con p_i la probabilidad de que k sea i .
 - $x = w_k(x)$
 - Dibujar x
 Mientras (no se interrumpa)



“Este algoritmo parte de un punto arbitrario x_0 , escoge de manera aleatoria una de las transformaciones de acuerdo con las probabilidades asignadas a ellas, calcula el punto imagen de x_0 por la transformación seleccionada, lo llama x y lo dibuja. Repite este proceso sobre x . La sucesión de puntos x forman una imagen que se aproxima al fractal. Este algoritmo es muchas veces más rápido que el de iteración determinística debido a que se calcula solo la imagen de un punto en cada iteración. Además, requiere memoria para almacenar solo una imagen que contiene los puntos que se han generado. Este algoritmo se llama en la literatura el juego del caos (chaos game) debido a que se están simulando sistemas dinámicos caóticos que tienen un atractor extraño” (Vargas & Cedeño, 2022)..

El resultado de la aplicación del algoritmo siempre genera el mismo objeto fractal (al considerar infinitas iteraciones), es decir la componente aleatoria del algoritmo no afecta el fractal descrito. La convergencia del algoritmo de iteración aleatoria es independiente de la asignación de probabilidades (Elton, 1988).

Compresión fractal de imágenes en el mundo real

“Hasta ahora se ha supuesto una imagen monocromática, donde el único problema es determinar qué píxeles deben ser negros. La codificación IFS puede extenderse fácilmente para comprimir imágenes en escala de grises, en este caso el problema es determinar qué píxeles dibujar y qué nivel de gris utilizar en cada uno. También se puede extender para comprimir imágenes en color, solo es necesario separar la imagen en tres componentes de color y comprimir cada uno individualmente como una imagen en escala de grises” (Salomon & Motta, 2010).

La compresión fractal mediante IFS fue descrita por primera vez por el Dr. Michael Barnsley y el Dr. Alan Sloan en 1988 (Barnsley & Sloan, 1988). Aunque afirmaron obtener tasas de compresión extraordinarias, el costo computacional de la codificación era prohibitivo. El principal proveedor de tecnología de compresión fractal es Iterated Systems, cofundada por Barnsley y Sloan.

Hoy en día, la compresión fractal parece lograr relaciones de compresión que son competitivas con JPEG a velocidades de codificación razonables.

La compresión fractal describe una imagen en términos de sí misma, en lugar de una cuadrícula de píxeles. Esto significa que las imágenes fractales pueden ser en cierto modo independientes de la resolución. De hecho, se puede fácilmente renderizar una imagen fractal en una cuadrícula más fina o más gruesa que la de la imagen de origen. Esta independencia de la resolución puede ser útil para presentar imágenes de calidad en una variedad de pantallas y medios impresos (Blelloch, 2013).

Aplicaciones de la compresión fractal de imágenes

La imagen de la Figura 27 fue codificada usando la aproximación fractal. La imagen original tenía un tamaño de 256 x 256 píxeles, y cada píxel fue codificado usando 8 bits; por lo tanto, el espacio de almacenamiento requerido era de 65536 bytes. Las transformaciones del IFS para la imagen comprimida requirieron un total de 4580 bytes, lo cual se traduce en una tasa promedio de 0.56 bits por píxel.

El proceso de decodificación comenzó aplicando el IFS a una imagen cuyos píxeles eran todos cero. Las seis primeras iteraciones de este proceso se muestran en la Figura 28. El proceso converge en nueve iteraciones, la imagen final se muestra en la Figura 29. Nótese que la reconstrucción no está libre de distorsiones o artefactos, estos son especialmente visibles en las regiones del mentón y el cuello (Sayood, 2017).



Figura 27. Imagen original Elif.



Figura 28. Primeras seis iteraciones de la decodificación fractal.



Figura 29. Reconstrucción final imagen Elif.

Ejercicio 10

1. Aplique la transformación de distorsión $\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$ a los cuatro puntos $(1, 0)$, $(3, 0)$, $(1, 1)$ y

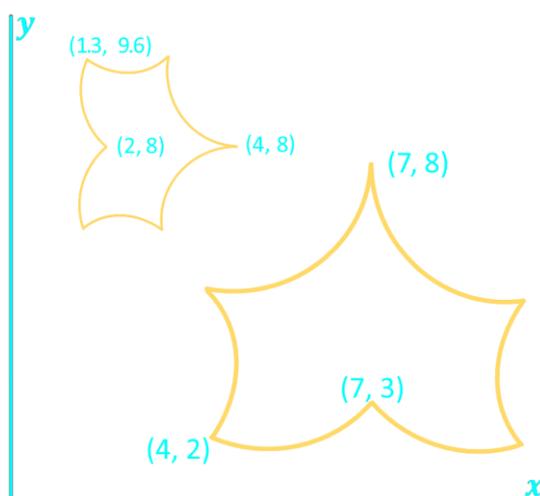
$(3, 1)$ (Salomon & Motta, 2010):

a. ¿Cuáles son los puntos transformados?

$$(x,y) \rightarrow (x-y,y) \quad (1,0) \quad (3,0) \quad (0,1) \quad (2,1)$$

b. ¿Qué figura geométrica representan los puntos resultantes tras la transformación?

2. Calcule los coeficientes de la transformación afín que lleva la hoja grande a la hoja pequeña en la figura que se muestra:



2. Aspectos didácticos: eBooks interactivos

En el presente Capítulo se realiza la caracterización de los eBooks (electronic Books) interactivos, con base en una revisión histórica de su evolución, el estudio de su arquitectura y la descripción de los beneficios de su uso en ambientes educativos. Por lo tanto, en la Sección 2.1 se presenta una breve reseña histórica de la evolución de los eBooks, en la Sección 2.2 se expone la taxonomía de los eBooks, en la Sección 2.3 se define el concepto de eBook interactivo, en la Sección 2.4 se expone la arquitectura de los eBooks interactivos, en la sección 2.5 se listan las principales características de un eBook interactivo, y en la sección 2.6 se presentan los beneficios de los eBooks interactivos en ambientes educativos.

2.1 Reseña histórica de la evolución de los eBooks

La historia de los eBooks se remonta a 1971 con el Proyecto Gutenberg (Lebert, 2009) iniciado por Michael Hart, que consistió en la digitalización de gran cantidad de libros clásicos para ser distribuidos libremente en la red; en la fase inicial el proyecto presentó un crecimiento limitado debido al poco acceso a los equipos de cómputo en la época, sin embargo el Proyecto Gutenberg aún continúa con sus actividades y en la actualidad contiene más de 57.000 eBooks accesibles desde su biblioteca mediante el enlace www.gutenberg.org (Jin, 2014).

Seguidamente, con el desarrollo de la Web en 1990 y la aparición del primer navegador en 1993, Internet comienza a ser parte de la cotidianidad (Lebert, 2009) y con ello todas sus ventajas; los libros en físico clásicos comienzan a hacer alusión a direcciones web que complementan la experiencia del lector con lecturas asociadas y contenido multimedia, dando inicio a las primeras formas de interactividad con la información.

Con el pasar de los días, proliferan los sitios web entre los que cabe destacar Online Books Page, una página web en la cual se indexan en forma de inventario todos los textos electrónicos de libre acceso para los usuarios finales. Por otro lado, los periódicos comienzan a ofrecer sus lecturas diarias a través de sus sitios web, posicionando al Times Online de Reino Unido y al New York Times en línea como los pioneros en esta modalidad de prensa.

En 1998, con la salida al mercado del Rocket eBook como el primer equipo “*e-reader*”^[1] especializado para la lectura electrónica, inicia el auge de los eBooks con un crecimiento exponencial impulsado principalmente por los avances tecnológicos que incrementan el atractivo de los eBooks y los e-readers entre los lectores. (Jin, 2014).

Wikipedia abre el paso a las enciclopedias en la web en 2001. Se trata de una enciclopedia gratuita y colectiva, cuyo contenido es redactado en un entorno de colaboración por miles de voluntarios, con posibilidades de corregir y completar los artículos, propios o de otros contribuidores. Con el tiempo nacen otras enciclopedias que existen en la actualidad, como Citizendium y Encyclopedia of Life.

En 2002, el Instituto de Tecnología de Massachusetts - MIT publicó en la web el contenido de estudio de 32 cursos representativos, correspondiente a textos de conferencias, trabajos prácticos, ejercicios con sus respectivas soluciones, audios y videos educativos, lo cual aún en la actualidad favorece la ampliación del acceso a la educación.

Google como importante exponente en el ámbito informático se inclina hacia el mercado de los eBooks y en mayo de 2005 lanza una versión beta de Google Print, un proyecto de biblioteca digital que consta de 15 millones de libros digitalizados de bibliotecas asociadas. Este proyecto fue finalizado por conflictos con asociaciones de derechos de autor, pero más tarde, Google reanuda su proyecto con el nacimiento del actual Google Books (Lebert, 2009).

En 2007, la librería online Amazon.com lanza su e-reader Kindle como complemento para el acceso y la lectura a los más de 80.000 títulos disponibles a la fecha en su catálogo. El Kindle utiliza una pantalla con tecnología E-Ink, la cual permite visualizar el contenido mediante la reorganización de partículas especializadas que representan la tinta en una pantalla, permitiendo un ultra bajo consumo de energía y elevados ángulos de visión en los e-readers, incluso en ambientes de alta iluminación.

Los avances tecnológicos han sido el constante combustible para el crecimiento de los eBooks; en el trascurso de la década de 2010, surgieron los teléfonos inteligentes y las pantallas de alta resolución, haciendo de la lectura una experiencia cada vez más digital. A mediados de 2010 con el lanzamiento del iPad, fue anunciada la aplicación de software Apple Books que permite el acceso a eBooks de alto contenido multimedia denominados propietariamente

[1]Dispositivo portátil en el que se pueden leer versiones electrónicas de libros, periódicos, revistas, etc.

“iBooks”, los cuales permiten la interactividad del lector y adicionan funcionalidades de perfiles de usuario, sincronización y autoevaluación (Figueiredo, Bidarra, & Natalio, 2014; Sepúlveda & Suarez, 2015).

2.2 Taxonomía de los eBooks

La gran diversidad de los eBooks en sus formas, contenido, medios de acceso, etc., hace que existan numerosas clasificaciones, sin embargo, dado el consumo de información de tipo multimedia por parte de la sociedad actual, se describe la clasificación de los eBooks con base en su contenido (Camargo, 2018).

- *eBooks textuales*: corresponden a las versiones digitales de los libros físicos (libros digitales), compuestos únicamente por texto y cuyas páginas están organizadas en forma lineal guiando el camino de la lectura de manera unidireccional.
- *Audiobooks*: o libros parlantes, se caracterizan por que incluyen sonidos que pueden ser narraciones de páginas totales o parciales, siendo muy útiles para lecturas infantiles en lo cual son comúnmente empleados. Adicionalmente son usados de manera constante en escenarios de aprendizaje para casos de discapacidades visuales.
- *eBooks de imágenes*: contienen principalmente imágenes que pueden ser estáticas o animadas, se caracterizan por la preponderancia de información de tipo gráfico en comparación con el texto que es mínimo o nulo. En este tipo de eBooks se encuentran los álbumes fotográficos digitales, los catálogos digitales, etc.
- *eBooks multimedia*: combinan texto, imagen, animación, video, sonido y pueden permitir la interactividad del usuario con el contenido del eBook, logrando que se interese vivamente por el mundo informático y se auto estimule por sus logros.
- *eBooks hipertexto o hipertextuales*: son libros multimedia que adicionalmente incluyen hipertextos, es decir, presentan una estructura no lineal en la narración, a pesar de realizarse mediante texto, permitiendo al lector decidir el flujo de la lectura a través de hipervínculos que enlazan información adicional que enriquece el contenido base del eBook.

- *eBooks inteligentes*: incorporan técnicas de inteligencia artificial que proporcionan al lector servicios avanzados de interactividad, dotándolos de la capacidad de adaptar dinámicamente los contenidos y las actividades interactivas al interés del usuario.
- *eBooks telemáticos*: hacen uso de la conectividad para proveer características especiales como teleconferencias, envío de correos electrónicos, foros y chats, que permiten entornos de cooperación y aprendizaje grupal, así como también brindan acceso a recursos de consulta y profundización tales como las bibliotecas digitales.
- *eBooks ciberespaciales*: contienen entornos de realidad virtual “VR” o aumentada “AR”, permitiendo experiencias inmersivas para el usuario que facilitan el aprendizaje.
- *eBooks Interactivos*: por último, y en fase de crecimiento, esta clase de eBooks corresponde a la unión y consolidación de las clases previamente explicadas, permitiendo en su contenido todo tipo de elementos multimedia, hipermedia, interactividades, elementos de VR y componentes gamificados, con lo que se maximiza la experiencia de aprendizaje en los lectores (Figueiredo, Bidarra & Natalio, 2014; Gallego, Molina & Llorens, 2014).

2.3 Concepto de eBook interactivo

El término eBook o libro electrónico, se utiliza para denominar los libros digitales o la versión electrónica de un libro físico; estos eBooks pueden ser visualizados en un sin número de dispositivos tales como PCs, smartphones, tabletas digitales, e-readers como el Kindle Amazon, entre otros dispositivos electrónicos (Mutalib, Noor, Hashim, & Mahari, 2012; Sepúlveda & Suarez, 2015). Se refiere a una publicación digital no periódica, es decir que se complementa en uno o varios volúmenes finitos. Inicialmente fue concebido como la digitalización o versión digital de los libros impresos, pero el auge del contenido multimedia y el uso masivo del internet ha permitido la adopción de elementos de texto, gráficos, imágenes, sonidos y videos (Gama, 2002), encajando con las nuevas formas de lectura y escritura de la sociedad. Los eBooks ofrecen a los estudiantes y profesores una herramienta adicional para apoyar y mejorar el proceso de aprendizaje, pues aun cuando la mayoría de éstos son todavía una versión electrónica de sus versiones impresas, crece continuamente su acogida por parte de los lectores.

El uso de eBooks en lugar de libros de texto en las aulas de clases es una nueva tendencia, gracias a sus ventajas que van desde la disminución del peso con respecto a las versiones en físico de los libros, hasta la generación de un proceso de aprendizaje dinámico para los estudiantes, debido a las características atractivas del eBook como lo son: las funciones amigables para el usuario, gráficos novedosos, control del tamaño de texto, etc. (Mutalib et al., 2012).

En la actualidad el eBook se ha convertido en un elemento de hipermedia (hipertexto y multimedia) de gran complejidad y cuya elaboración y desarrollo involucra profesionales músicos, escritores, artistas del dibujo, diseñadores, entre otros, que permiten al eBook ser ese elemento extasiado en multimedia (Camargo, 2008).

Recientemente, se han adelantado investigaciones que evalúan la posibilidad de utilizar formatos comunes de eBooks para crear *eBooks interactivos* o libros gamificados (similares al formato iBook de Apple) para la enseñanza y el aprendizaje, creando un modelo de libro dinámico que puede funcionar como un juego educativo y que presenta una adecuada usabilidad y un prometedor potencial pedagógico (Figueiredo, Bidarra, & Natalio, 2014).

El aprendizaje virtual presenta un sin número de estrategias y herramientas educativas cada una con gran cantidad de ventajas; pues bien, el eBook interactivo es parte de estas herramientas y está compuesto de ellas, su variedad de contenido permite la inclusión de textos, imágenes, videos, juegos, puzzles y demás elementos multimedia e interactivos, consolidados en un único elemento fácil de transportar, acceder o compartir.

Un eBook interactivo es una herramienta de gran potencial para la educación, puede contener material de tipo texto cuya lectura instruye de manera directa al estudiante, elementos multimedia alusivos y complementarios que facilitan la absorción del conocimiento, e interactividades o evaluaciones gamificadas que permiten al lector autoevaluar, retroalimentar y maximizar lo aprendido (Figueiredo, Bidarra, & Natalio, 2014).

Los maestros buscan constantemente nuevas formas de proporcionar información y conocimiento de manera efectiva y atractiva para los estudiantes, emplean herramientas de presentación, tableros interactivos, audios y videos de ilustración buscando un "escenario" idóneo para lograr este objetivo. Siguiendo este ideal, el eBook interactivo se posiciona como la respuesta a esta búsqueda en donde todas estas estrategias y otros complementos propios de un ambiente

digital son integrados en un entorno educativo virtual, logrando mayor atracción por parte de los estudiantes y aumentando la relevancia del autoaprendizaje, ya que el lector es guiado directamente a través del contenido del eBook interactivo con base en los objetivos de aprendizaje (Binas, Štancel, Novák & Michalko, 2012).

El uso de eBooks interactivos en la educación tiene el potencial para resultados extremadamente prometedores. Su empleo está basado principalmente en el mayor interés que despiertan debido a sus características atractivas, por tanto, los eBooks interactivos entregan su material de conocimiento a través de plataformas que ilustran mediante el contenido multimedia, la interacción, los estudios de casos, etc., logrando mayores tasas de aprendizaje que las generadas por aplicaciones estándar.

Los dispositivos portátiles, tales como tablets y smartphones, se han convertido en el principal componente para adoptar una educación más interactiva, sin embargo, es necesario que los docentes y demás editores educativos centren sus esfuerzos en la generación de contenidos interactivos de calidad y se fomente su empleo y evolución en los procesos educativos.

Adicionalmente, si bien es fácil incluir gran cantidad de elementos interactivos en un eBook, los autores deben siempre estar alineados con la necesidad de una experiencia de usuario inmersiva y fluida, pues tal como en los libros clásicos, el flujo del contenido o la historia es un factor esencial para el éxito de la publicación.

2.4 Arquitectura de los eBooks interactivos

Todo desarrollo de software presenta una arquitectura específica que señala los lineamientos que rigen la estructura, funcionalidades e interacción entre las partes generales de dicho desarrollo. A continuación, se identifican las tres perspectivas: estática, funcional y dinámica, con las que se describe la arquitectura que rige los eBooks interactivos.

Vista Estática

Los eBooks interactivos son herramientas de software que siguen mayormente una arquitectura standalone, en la cual todo el software está concentrado en una única máquina dada su utilización de tipo monousuario y donde no es inconveniente realizar una tarea por vez. Su principal ventaja radica en su instalación y ejecución de manera sencilla y totalmente offline.

Este tipo de software puede contener distintos elementos (Cosentino, 2010), dentro de los que se pueden diferenciar la aplicación, las bases de datos y las conexiones de los controladores, los cuales son cargados localmente en el equipo donde se ejecute el eBook interactivo.

Vista funcional

De acuerdo con la arquitectura standalone, el eBook interactivo presentará una capa de aplicación o interfaz de usuario, la cual permitirá la interacción del usuario con el entorno de aprendizaje. Esta interacción debe facilitar dos modos de funcionamiento, el modo de lectura y el modo de estudio:

- *Modo Lectura*: el eBook permitirá su uso tal como un libro electrónico normal, ofrecerá un ambiente de lectura fluido permitiendo acceder a todo su contenido sin restricciones y desarrollar las interactividades sin ningún tipo de retroalimentación.
- *Modo de estudio*: el progreso de la lectura será controlado por el eBook, el cual tiene una ruta de estudio específica que busca generar un aprendizaje auto contenido, ésta se irá desplegando a medida que se cumplen los objetivos de aprendizaje, los cuales pueden ser evaluados a través de pequeños cuestionarios o asignaciones finales y con base en su resultado se determinará el "desbloqueo" del contenido siguiente. De no superar los objetivos requeridos, el eBook puede recomendar volver a leer secciones específicas, para fortalecer lo aprendido y obtener mejores resultados en el próximo intento (Binas, Štancel, Novák, & Michalko, 2012).

Adicionalmente, el eBook interactivo debe tener un elemento de conexión a un almacenamiento local en el que se almacenaran las credenciales del usuario, lo que permitirá poder guardar localmente su progreso y continuar jugando desde donde se detuvo en cualquier momento en el futuro (Droutsas, Patsilnakos & Symvonis, 2018).

Vista dinámica

Las relaciones y comunicaciones entre los elementos planteados para el eBook interactivo, serán direccionadas por la capa de aplicación debido a que ésta corresponde al núcleo del desarrollo. Desde allí, se ejecutan las distintas funciones indicadas en la vista anterior, permitiendo al usuario acceder al entorno virtual y a la ejecución del eBook interactivo en los modos de lectura o de estudio, cargar los perfiles de usuario almacenados localmente y almacenar avances y configuraciones de los perfiles. Esto se puede apreciar en el diagrama UML de la Figura 30.

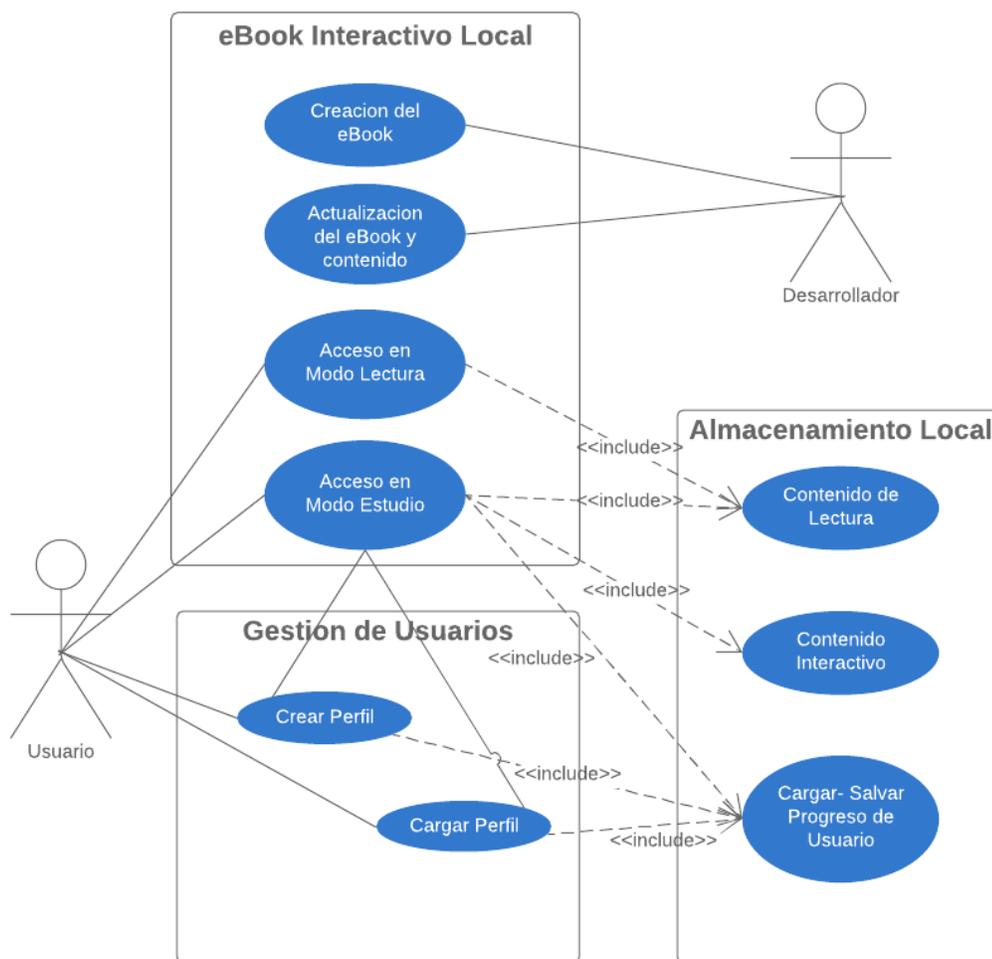


Figura 30. Diagrama UML casos de uso para la arquitectura del eBook interactivo.

2.5 Características del eBook interactivo

Un eBook interactivo debe tener las características y funcionalidades mínimas que se explican a continuación, para que permita alcanzar los objetivos de enseñanza y aprendizaje que se definan:

- Establecer un ambiente que genere al individuo el rol del lector o estudiante, que permita emplear el eBook como un elemento de consulta que fomenta el aprendizaje y la adquisición del conocimiento.
- “Los lectores del eBook son también sus evaluadores, por tanto, el autor debe crear un foro de discusión en Internet dedicado al eBook, cuya URL deberá estar presente en el mismo, en el que los usuarios puedan informar acerca de errores en el contenido, exponer sus reacciones y realizar preguntas directamente al autor o a otros lectores. Las opiniones expuestas en estos ambientes de discusión son muy útiles para el autor, quien puede realizar actualizaciones al eBook para dar solución a errores encontrados o para realizar adiciones de contenido o funcionalidades (Binas, Štancel, Novák & Michalko, 2012)” (Cedeño & Vargas, 2019).
- Uno de los pilares del eBook es ser un elemento de autoaprendizaje, por tanto, debe proporcionar actividades interactivas para el estudiante tales como pruebas, laboratorios o ejercicios que permitan evaluar los conocimientos adquiridos, cuyo control y realización se complete en el entorno del eBook interactivo, aunque requiera la realización de actividades externas. Un ejemplo simple sería un examen de matemáticas en donde el alumno accede a la actividad y procede a resolverla, pero dada la naturaleza de la misma, puede salir del entorno del eBook para realizar los cálculos en papel o con el uso de otros recursos físicos requeridos e ingresar nuevamente al entorno digital para ingresar las respuestas y validar los resultados.

En los casos de algunas tareas gráficas, por ejemplo actividades con conjuntos, se puede llegar a la solución directamente dentro del eBook con las herramientas proveídas en la interactividad. O en actividades de algoritmización se puede llegar a la solución de forma interactiva, mediante la creación de diagramas de flujo o modelamiento de los algoritmos por medio de bloques tipo Scratch¹ (Binas, Štancel, Novák & Michalko, 2012).

[1] Scratch es un entorno de desarrollo creado por el Instituto de Tecnología de Massachusetts – MIT. Se beneficia de los avances en el diseño de interfaces generar formas y modelos de desarrollo atractivos y accesibles a cualquier edad y nivel de formación.

2.6 Beneficios de los eBooks interactivos en ambientes educativos

Los eBooks interactivos se han convertido en una gran herramienta para la cultura y la educación a nivel global y son parte importante en el rol educativo de docentes y estudiantes, debido a los beneficios que se explican a continuación:

1. *Estudio y lectura en cualquier momento y lugar:* un eBook interactivo es un recurso digital que puede ser almacenado en un dispositivo móvil compatible, haciendo más fácil su transporte en lugar de llevar un morral lleno de libros físicos. Dicha portabilidad permite a los estudiantes realizar su consulta y estudio cuando lo deseen (Kitaboo, 2018).
2. *Diversidad de características interactivas:* en el mundo digital, las posibilidades provistas por el eBook superan las expectativas en los procesos educativos frente al libro tradicional (Armañanzas, 2010). Por un lado, el hipertexto sobrepasa el sentido lineal de la lectura clásica fomentando un aprendizaje abierto y autónomo, y presentando un contenido enriquecido que pluraliza la perspectiva del autor y el propio entendimiento del lector. Adicionalmente, el entorno digital enriquece la experiencia del usuario gracias a las funcionalidades interactivas, tales como anotaciones, zoom, índices, búsqueda, entre otras.
3. *Valor educativo y mejoras en el aprendizaje:* los eBook interactivos fundamentan su importancia para el aprendizaje en lo interesantes y llamativos que resultan para el lector, el secreto para un aprendizaje óptimo reside en la diversión (Armañanzas, 2010). Por tanto, la gamificación de los eBooks interactivos hace que estos sean divertidos para el lector y permite que su cerebro se sienta recompensado por adquirir nuevos conocimientos ampliando la necesidad de generar más y mejor aprendizaje.
4. *Aprendizaje autónomo:* en un eBook interactivo el lector se convierte en creador y fomentador de su propio aprendizaje, aprovechando y haciendo su propio camino a través del contenido y las interactividades que el docente plasmó en el eBook, dependiendo del interés y cultura de aprendizaje del lector.

5. *Cero costos de impresión:* los eBooks interactivos son netamente digitales y por tanto no requieren papel, eliminando totalmente la necesidad de imprimir y contribuyendo con la preservación del medio ambiente, siendo este un aspecto relevante considerando la cantidad de libros impresos que hoy en día se producen.
6. *Facilidad en la actualización del contenido:* al tratarse de un recurso educativo digital, permite su fácil actualización en cuanto a contenido y funcionalidades, a diferencia de los libros clásicos, cuyas actualizaciones corresponden a nuevas ediciones e impresiones físicas.
7. *Procesos de evaluación interactivos:* los eBooks interactivos incluyen autoevaluaciones interactivas que evalúan y refuerzan los conocimientos de los estudiantes, durante la lectura y al finalizar cada capítulo; estas evaluaciones muestran los resultados de manera inmediata permitiendo que el estudiante conozca sus debilidades y fortalezas en cuanto a lo aprendido.

3. Revisión de trabajos previos

El presente capítulo tiene como objetivo describir la revisión de la literatura realizada para determinar los trabajos previos sobre la enseñanza y aprendizaje de la compresión de datos usando eBooks interactivos. De esta manera, en la Sección 3.1 se describe la primera fase de la revisión sistemática de la literatura: recolección de la información. En la Sección 3.2 se describe la segunda fase de la revisión sistemática de la literatura: análisis de la información recolectada. Finalmente, en la Sección 3.3 se realiza la discusión de resultados y se exponen las conclusiones de la revisión.

3.1 Recolección de la información

Se realizó una búsqueda en las siguientes bases de datos: Springer Link, Science Direct, Scopus, IEEE Xplore y Engineering Database, usando la ecuación de búsqueda que se muestra en la Tabla 14.

Tabla 14. Ecuación de búsqueda en bases de datos.

Ecuación de búsqueda
<i>(interactive eBook) AND ((learning) OR (Teaching)) AND ((information theory) OR (data compression) OR (data coding) OR (source coding))</i>

Como resultado de la búsqueda se obtuvieron 254 publicaciones (ver Figura 31), que fueron filtradas considerando los criterios de inclusión y exclusión contenidos en la Tabla 15.

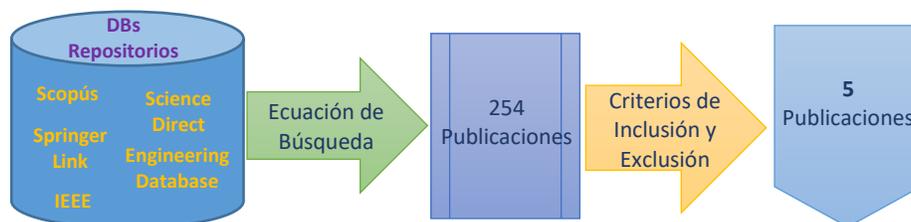


Figura 31. Proceso de recolección de información: Aplicación de la ecuación de búsqueda y criterios de inclusión y exclusión.

Tabla 15. Criterios de inclusión y exclusión

Criterios de inclusión	Criterios de exclusión
<ul style="list-style-type: none"> Publicaciones realizadas entre el año 2014 y el año 2022. 	<ul style="list-style-type: none"> Publicaciones realizadas antes del 2014
<ul style="list-style-type: none"> Desarrollos orientados a la educación de nivel medio y superior. 	<ul style="list-style-type: none"> Desarrollos orientados a la educación de nivel básico o inferior
<ul style="list-style-type: none"> Publicaciones escritas en inglés y español. 	<ul style="list-style-type: none"> Publicaciones escritas en idiomas distintos al inglés y español.

Se consideró un rango de fechas de las publicaciones entre el año 2014 y el año 2022, esto con el fin de rastrear las publicaciones más recientes y con las últimas tecnologías aplicadas. Se excluyeron publicaciones asociadas a desarrollos orientados a la educación de nivel básico o inferior, ya que la temática del eBook interactivo (compresión de datos) está dirigida a una educación de nivel medio y superior. Se consideraron artículos en inglés, ya que es el lenguaje en el cual se encuentra el mayor volumen de publicaciones, y adicionalmente en español por contar con un buen número de publicaciones asociadas al tema.

Después de aplicar los criterios de inclusión y exclusión a las 254 publicaciones obtenidas con la ecuación de búsqueda, quedaron finalmente 5 artículos seleccionados (ver Figura 31), los cuales se listan en la Tabla 16:

Tabla 16. Lista de artículos seleccionados en el proceso de recolección de información.

Artículo	Autor	Año
<i>The Development of Interactive Physics EBook in Rigid Body Equilibrium and Rotational Dynamics</i>	Adawiyah, Harjono & Gunawan	2018
<i>New Interactive Books for Control Education</i>	Guzman, Piguet, Dormido, Berenguel & Costa-Castelló	2018
<i>Design and research potential of interactive textbooks: the case of fractions</i>	Hoch, Reinhold & Werner	2018
<i>Using interactive multimedia eBooks for learning blood cell morphology in pediatric hematology</i>	Hsiao, Tiao, & Chen	2016
<i>The development of a GameBook for education</i>	Figueiredo & Bidarra	2015

3.2 Análisis de la información

Los artículos (y los eBooks asociados) seleccionados como resultado de la recolección de información se analizan a continuación, de acuerdo con la siguiente estructura:

- **Descripción general:** Se mencionan los autores, el contenido de enseñanza y la población objetivo.
- **Tecnología:** Se exponen consideraciones técnicas respecto al desarrollo del eBook interactivo, tales como la plataforma de desarrollo, las tecnologías utilizadas y los tipos de contenido multimedia incluidos, esto último permite clasificar el eBook según la taxonomía de la Sección 2.2
- **Logros:** Se describen los resultados, beneficios, debilidades y logros obtenidos tras la implementación y evaluación del eBook interactivo.

3.2.1 The Development of Interactive Physics EBook in Rigid Body Equilibrium and Rotational Dynamics

- **Descripción general:** esta investigación de Adawiyah, Harjono & Gunawan (2018) expone el desarrollo de un eBook interactivo para educación media, para la temática de equilibrio del cuerpo rígido y dinámica de rotación.
- **Tecnología:** el eBook interactivo está en formato aplicación ejecutable para Windows (.exe) y fue desarrollado con las herramientas Professional Flash CS 5 y Action Script 3. Su contenido multimedia incluye videos, animaciones y audio, lo que hace que corresponda a un eBook multimedia según la taxonomía de la Sección 2.2.
- **Logros:** la implementación del eBook tomó como referencia a estudiantes de educación secundaria, fue probado inicialmente con un grupo focal de 18 estudiantes para determinar su efectividad. Como resultado de esta prueba piloto se obtuvo que el eBook interactivo es adecuado como material de enseñanza para la temática, y genera mejores resultados en comparación con las metodologías clásicas de enseñanza.

3.2.2 New Interactive Books for Control Education

- **Descripción general:** este trabajo de Guzman, Pigué, Dormido, Berenguel & Costa-Castelló (2018) tuvo como objetivo crear eBooks interactivos para la enseñanza de la teoría de control. Los autores seleccionaron un capítulo de un libro clásico del tema (específicamente sistemas de segundo orden), y lo tomaron como base para diseñar una versión interactiva. En cuanto al público objetivo, no se define claramente en el artículo, pero dado el contenido de enseñanza se supone que está orientado a estudiantes de pregrado y posgrado de carreras afines a la temática.
- **Tecnología:** Se trató de un eBook interactivo cuyo contenido está limitado a texto, imágenes e interactividades sobre imágenes, ejecutado en un entorno web, lo que hace que corresponda a un eBook de imágenes según la taxonomía de la Sección 2.2. El texto fue escrito en Látex; las pruebas e interactividades fueron desarrolladas en la herramienta de simulación *sysquake*, y luego fueron convertidas a formatos XHTML y JavaScript y compiladas en el formato epub^[1].
- **Logros:** el artículo no expone resultados de efectividad del eBook, ya que se encuentra en desarrollo; sin embargo, está siendo construido con base en las necesidades de material educativo interactivo para la temática seleccionada, manifestadas por muchos autores.

3.2.3 Design and research potential of interactive textbooks: the case of fractions

- **Descripción general:** esta investigación de Hoch, Reinhold & Werner (2018) describe la implementación y la respectiva sustentación teórica de un eBook interactivo denominado "ALICE: fracciones", correspondiente a un libro de matemáticas interactivo para el aprendizaje de manera didáctica de los números fraccionarios y sus operaciones. Está dirigido a estudiantes de sexto grado.
- **Tecnología:** "ALICE: fracciones" fue desarrollado con la herramienta iBooks Author, contiene encapsulados web denominados widgets accesibles desde el eBook, los cuales fueron desarrollados usando HTML5, JavaScript y CindyJS. Su contenido multimedia es

[1] EPUB es un formato de archivo de libro electrónico que utiliza la extensión de archivo ".epub". El término es un acrónimo de la expresión inglesa electronic publication (publicación electrónica).

principalmente de imágenes e interactividades, lo que hace que corresponda a un eBook interactivo según la taxonomía de la Sección 2.2.

- **Logros:** la prueba piloto consistió en una intervención educativa a 155 estudiantes de sexto grado; como resultado surgió que los eBooks interactivos generan nuevas formas de enseñanza en las aulas, pues permiten incluir tareas interactivas y evaluativas que incrementan la efectividad en la educación.

3.2.4 Using interactive multimedia eBooks for learning blood cell morphology in pediatric hematology

- **Descripción general:** este trabajo de Hsiao, Tiao, & Chen (2018) presenta el desarrollo de un eBook multimedia interactivo para el aprendizaje de la morfología de las células sanguíneas, el cual es comparado con el proceso de aprendizaje tradicional de esta temática mediante el uso de diapositivas. Este eBook está dirigido a estudiantes de medicina.
- **Tecnología:** para el desarrollo del eBook interactivo se utilizó la herramienta SimMagic eBook, correspondiente a un software desarrollado en Taiwán por Hamastar Technology Company. Este eBook es rico en multimedia, incluye imágenes, videos e interactividades evaluativas tipo quiz, lo que hace que corresponda a un eBook interactivo según la taxonomía de la Sección 2.2.
- **Logros:** la validación del software se realizó con estudiantes de la escuela de medicina de Taiwán. Fue evaluado inicialmente con un grupo focal de 51 alumnos, de los cuales 25 fueron instruidos mediante diapositivas clásicas y los restantes 26 emplearon el eBook multimedia interactivo. Como principal conclusión se obtuvo que, tras el proceso de instrucción, se evidenció un 30% de mejor desempeño por parte de los estudiantes instruidos con el eBook interactivo, en las pruebas de medición de los conocimientos adquiridos.

3.2.5 The development of a GameBook for education

- **Descripción general:** esta investigación realizada por Figueiredo & Bidarra (2015) presenta un novedoso eBook interactivo titulado "Aventuras en el río Guadiana", cuyo contenido de enseñanza obedece a la temática de la asignatura *estudios ambientales* de cuarto grado. El eBook se dirigió a niños brasileños de cuarto grado, con edades de 9 a 10 años.
- **Tecnología:** se trata de un eBook interactivo gamificado que está implementado en Unity3D dada su compatibilidad para exportarse a iPhone, iPad, equipos Android, navegadores web y Windows. Su contenido multimedia incluye gráficos en 3D, animaciones, videos e imágenes, lo que hace que corresponda a un eBook interactivo según la taxonomía de la Sección 2.2.
- **Logros:** se realizaron pruebas de usabilidad que identificaron una adecuada experiencia de usuario y un potencial como herramienta de apoyo a la enseñanza-aprendizaje.

3.3 Discusión y conclusiones de la revisión

En la Tabla 17 se presenta un resumen de los resultados más relevantes del análisis realizado en la sección anterior, clasificando las publicaciones por área de estudio, herramienta de desarrollo del eBook asociado, nivel educativo del público objetivo y clasificación según taxonomía de la Sección 2.2.

Tabla 17. Resumen de análisis de las publicaciones

Artículo	Área de Estudio	Herramienta de Desarrollo	Nivel Educativo Objetivo	Clasificación eBook
<i>The Development of Interactive Physics EBook in Rigid Body Equilibrium and Rotational Dynamics</i>	Ciencias Exactas	Professional Flash CS 5	Educación Media	eBook multimedia
<i>New Interactive Books for Control Education</i>	Ingeniería	XHTML	Educación Superior	eBook de imágenes
<i>Design and research potential of interactive textbooks: the case of fractions</i>	Ciencias Exactas	iBooks Author	Educación Media	eBook interactivo
<i>Using interactive multimedia eBooks for learning blood cell morphology in pediatric hematology</i>	Ciencias de la Salud	SimMagic eBook	Educación Superior	eBook interactivo
<i>The development of a GameBook for education</i>	Ciencias Sociales	Unity 3D	Educación Primaria	eBook interactivo

La revisión sistemática de literatura realizada permitió concluir lo siguiente:

- Todos los trabajos seleccionados mencionaron resultados satisfactorios en las pruebas de aprendizaje realizadas con los estudiantes, gracias a la implementación de los eBooks.
- Se encontró que existen trabajos que emplean eBooks interactivos para la enseñanza y el aprendizaje de contenidos educativos, pero no se encontraron eBooks interactivos para la temática específica de comprensión de datos. Lo anterior indica que el desarrollo propuesto es una herramienta innovadora y única en su contenido, constituyéndose en una apuesta de interés para utilizar en los procesos educativos pertinentes.

4. Metodología: análisis, diseño y desarrollo del eBook interactivo

La metodología utilizada para el desarrollo del “eBook Interactivo – Compresión de Datos” está inspirada en el modelo de diseño instruccional ADDIE, cuyo nombre es un acrónimo de los pasos clave que conforman este modelo: Análisis, Diseño, Desarrollo, Implementación y Evaluación (Salas, 2016). En este capítulo, en la Sección 4.1 se describe el análisis, en la Sección 4.2 el diseño y en la Sección 4.3 el desarrollo del eBook interactivo.

4.1 Fase de análisis

Inicialmente es necesario caracterizar la población a la cual estará dirigido el eBook, definir los objetivos de aprendizaje y el contenido de enseñanza, describir el enfoque pedagógico y determinar la herramienta más apropiada para el desarrollo del eBook.

4.1.1 Caracterización del público objetivo

El eBook interactivo planteado será diseñado con base en las características del público específico al que estará dirigido, para así fomentar un aprendizaje direccionado, inclusivo y favorable para el desarrollo instruccional. Esta caracterización se incluye en la Tabla 18 (Instituto Caro y Cuervo, 2016 y Barreiro, 2015).

Tabla 18. Caracterización del público objetivo para el eBook Interactivo.

Perspectiva de Análisis	Caracterización
<i>Perfil demográfico:</i>	Dirigido a jóvenes de los 15 a los 25 años de edad, tanto para hombres como mujeres de habla hispana, con alguna vinculación al sistema educativo público o privado, inicialmente de Colombia.
<i>Perfil cultural:</i>	Debido al contenido netamente educativo, estará dirigido a los jóvenes sin distinción política, social y/o económica, sin embargo, es indispensable el uso de un PC o Smartphone para la visualización del contenido. Está focalizado a alumnos proactivos de primeros semestres de educación superior, con conocimientos informáticos y matemáticos acordes a este nivel de formación.
<i>Perfil de usuario:</i>	El eBook está concebido como un material de estudio para un uso frecuente durante el aprendizaje y como elemento de consulta para posteriores investigaciones. Es necesario el interés del usuario en el aprendizaje y la investigación de nuevos conocimientos.

4.1.2 Definición de los objetivos de aprendizaje

A continuación, se relacionan los objetivos de aprendizaje que se esperan alcanzar con la intervención educativa.

Objetivo general

Comprender los conceptos básicos y los principales métodos de compresión de datos.

Objetivos específicos

- OE 1** Establecer el concepto, funcionamiento y el rol actual de la compresión de datos.
- OE 2** Identificar y diferenciar las medidas de rendimiento en la compresión de datos.
- OE 3** Comprender el objeto de estudio y aplicaciones de la teoría de la información y la compresión de datos.
- OE 4** Identificar los conceptos de información propia y entropía de la información.
- OE 5** Identificar la relación entre los conceptos de la teoría de la información y la compresión de datos.
- OE 6** Construir códigos probabilísticos, Categorizar los diferentes tipos y Evaluar sus aplicaciones.
- OE 7** Comprender los algoritmos de compresión por diccionario LZ77 y LZW, y los algoritmos de compresión basada en contexto RLE, MTF y PPM.
- OE 8** Identificar y diferenciar Los algoritmos de compresión probabilística y por diccionario.
- OE 9** Diferenciar las técnicas de compresión con pérdida y sin pérdida.
- OE 10** Identificar las principales características de la cuantización escalar y la codificación diferencial.
- OE 11** Comprender los procesos básicos de la compresión JPEG y de los esquemas de análisis/síntesis (Cedeño & Vargas, 2021).

4.1.3 Definición del contenido de enseñanza

En la Tabla 19 se establece el contenido de enseñanza de acuerdo con los objetivos de aprendizaje definidos, contenido que corresponde a los aspectos disciplinares del Capítulo 1.

Tabla 19. Contenido de enseñanza y objetivos de aprendizaje

<i>Contenido de enseñanza</i>	<i>Objetivos de aprendizaje</i>
<p><i>Conceptos básicos de la compresión de datos:</i></p> <ul style="list-style-type: none"> • Técnicas de compresión de datos • La compresión de datos y los sistemas de comunicación • Medidas de rendimiento de la compresión. 	<p>OE 1</p> <p>OE 2</p>
<p><i>Introducción a la teoría de la información</i></p> <ul style="list-style-type: none"> • Fuentes de información • Contenido de información de un símbolo • Información promedio o entropía • Codificación de fuente 	<p>OE 3</p> <p>OE 4</p> <p>OE 5</p>
<p><i>Técnicas de compresión sin pérdidas:</i></p> <p>Métodos Estadísticos</p> <ul style="list-style-type: none"> • Codificación Huffman • Códigos aritméticos <p>Compresión basada en contexto</p> <ul style="list-style-type: none"> • Codificación run-length (RLE) • Codificación mover-al-frente (MTF) • Codificación PPM <p>Métodos de diccionario</p> <ul style="list-style-type: none"> • Lempel-Ziv 77 (LZ77) • Lempel-Ziv-Welch (LZW) 	<p>OE 6</p> <p>OE 7</p> <p>OE 8</p>
<p><i>Técnicas de compresión con pérdida:</i></p> <ul style="list-style-type: none"> • Cuantización escalar • Codificación diferencial • Codificación de transformación • Esquemas de análisis/síntesis 	<p>OE 9</p> <p>OE 10</p> <p>OE 11</p>

4.1.4 Descripción del enfoque pedagógico

Debido a sus características particulares (ver Secciones 2.5 y 2.6), el eBook se considera una tecnología que puede apoyar un *enfoque de aprendizaje constructivista*, el cual se basa en la construcción del aprendizaje de manera autónoma y activa, debido a que permite al estudiante explorar actividades de manera libre centradas en sus habilidades y partiendo de sus experiencias con instrucciones guiadas, así mismo la evaluación se centra en el propio aprendizaje de la persona. (Rodríguez, Martínez, y Lozada 2009).

Dentro del marco de la enseñanza-aprendizaje constructivista, una variante es el modelo de *aprendizaje inmersivo*, el cual posibilita el desarrollo de habilidades a través de la interacción en mundos virtuales, promoviendo el uso de sentidos no solo basados en párrafos textuales, sino en la implementación de estímulos a través de audios y señales visuales promoviendo símbolos no verbales que generan una mayor apropiación cognitiva (Toca y Carrillo, 2019).

4.1.5 Selección de la herramienta de desarrollo

El eBook interactivo será una aplicación de software interactiva en 3D, en la que se alojará el contenido de estudio y las respectivas actividades evaluativas de forma guiada y estructurada para facilitar el aprendizaje, por lo que su desarrollo deberá realizarse en herramientas apropiadas para este fin, con la mejor calidad y desempeño posible” (Cedeño & Vargas, 2019). Dada la variedad de herramientas de desarrollo disponibles, se realiza un análisis comparativo de las opciones más relevantes con base en unos criterios de evaluación definidos, con el objetivo de seleccionar la opción más adecuada que se utilizará para el eBook interactivo.

4.1.5.1 Criterios de evaluación.

Los siguientes son los criterios de interés para la evaluación de las herramientas de desarrollo “(Cada criterio tendrá una calificación de 1 a 5 según se indica):

- *Compatibilidad*: Todo desarrollo de software presenta una arquitectura específica que establece las condiciones requeridas para su ejecución y funcionamiento en el entorno de software o sistema operativo (SO). Debido a que se requiere que el eBook sea ejecutado en diversos

equipos tales como PCs, Tablets, Smartphones, etc., la evaluación de este criterio otorgará una calificación según el número de plataformas o SO con las que sea compatible el producto generado, es decir, una calificación igual a 1 si es compatible solo con una plataforma, calificación igual a 2 si es compatible con dos plataformas, etc.

- *Uso de elementos multimedia:* La herramienta de desarrollo debe permitir el uso de todo tipo de multimedia ya que el eBook estará centrado en emplear contenido interactivo de tal forma que incentive el proceso de aprendizaje; por tanto, para la evaluación de este criterio se identificarán cinco elementos multimedia básicos: texto, imágenes, sonidos, videos y elementos 3D, cada uno de los cuales sumará un punto a la calificación de este criterio.
- *Licencia de desarrollo:* Las licencias de software son contratos de condiciones de uso entre el proveedor y el usuario, que en general deben ser aceptadas previamente al primer uso del software y están soportadas por las leyes de cada país. En este caso se evaluarán las condiciones que establece el proveedor de la herramienta de desarrollo, asignando una calificación de 1 si existe un cobro por usar y desarrollar contenido, o un 5 si la herramienta es de uso gratuito.
- *Licencia de publicación de contenidos:* Adicional a las condiciones para desarrollar contenido, pueden existir limitaciones para la publicación de los contenidos desarrollados, por tanto, se asignará una calificación de 1 si existe un costo por publicar los contenidos, un 3 si el contenido presenta marcas de agua, publicidad, o limitaciones en calidad, duración, tamaño, etc., y un 5 si la publicación de los contenidos desarrollados se puede realizar sin restricción alguna (Cedeño & Vargas, 2019).

4.1.5.2 Análisis comparativo de las herramientas para el desarrollo del eBook

Con base en los criterios definidos en la Sección 4.1.5.1, a continuación se realiza un análisis de las opciones de herramientas de desarrollo más relevantes.

- *iBooks Author:* Herramienta propietaria de Apple, disponible de manera gratuita en el Appstore, principalmente diseñada para el desarrollo de libros electrónicos en el formato propietario “ibooks”.

Se enfoca en el desarrollo de material para educación, sin embargo, permite crear un sin número de contenidos que van desde libros de solo texto hasta catálogos interactivos, presentaciones, tutoriales, etc. Las creaciones pueden ser enriquecidas con todo tipo de elementos interactivos tales como imágenes, vídeos, modelos 3D, animaciones HTML5, widgets, entre otros.

iBook Author es una herramienta de gran versatilidad para la creación de libros interactivos, de manera muy amigable e intuitiva, pero sus publicaciones y ejecución solo son posibles en las plataformas de Apple, es decir iPhone, iPad y demás.

- *NeoBook*: Es un programa de Neosoft, que se fundamenta como herramienta de desarrollo multimedia para la creación de aplicaciones interactivas para ser visualizadas en cualquier ordenador.

Los elementos desarrollados son aplicaciones que permiten la interacción del usuario, mediante la lectura, respondiendo cuestionarios, entre otras; estas aplicaciones pueden contener texto, imágenes, audio, vídeo o gifs, mostradas en varias páginas explorables.

Neobook es una herramienta útil para docentes en la elaboración de excelentes elementos para la clase, pero se encuentra limitada en cuanto a que solo permite la ejecución de los contenidos bajo sistemas operativos Windows, imposibilitando su ejecución en smartphones y demás dispositivos móviles.

- *Moglue*: Herramienta gratuita para la creación de libros interactivos compatibles con IOS y Android, mediante la ejecución de Moglue Viewer que es su visualizador de contenidos. Permite crear libros interactivos sin necesidad de tener conocimientos de programación o de diseño gráfico de manera muy intuitiva.

La herramienta contiene una biblioteca de elementos prediseñados, pero permite la inclusión de contenido propio (imágenes y audio) en los desarrollos.

La licencia de desarrollo de Moglue es gratuita, pero su licencia de publicación tiene un costo que oscila los 100 dólares al año.

- *Kotobee Author*: es un software de escritorio para la creación de eBooks de manera intuitiva, permitiendo la inclusión de gran variedad de contenido como videos, audios, imágenes, cuestionarios, widgets de libros, aplicaciones HTML5, archivos adjuntos, objetos 3D entre otros.

La creación y edición del contenido es realizada mediante HTML, JS y CSS, y permite la exportación de contenidos en gran variedad de formatos, lo que extiende su compatibilidad con diversas plataformas tales como exploradores web y Windows, Android, iPhone, Kindle, entre otras.

El uso de Kotobee Author es licenciado por usuario y por cantidad de exportación a plataformas como Android o iPhone, la licencia ronda entre los 150 y 300 USD según se requiera.

- *Unity 3D*: herramienta de desarrollo de videojuegos, que permite la creación de contenido de entretenimiento de alta calidad en entornos 2D y 3D para todo tipo de consolas, dispositivos móviles y PCs de escritorio. Aunque existe la licencia paga, esta herramienta cuenta con la versión *Unity Personal* la cual es gratuita y totalmente funcional.

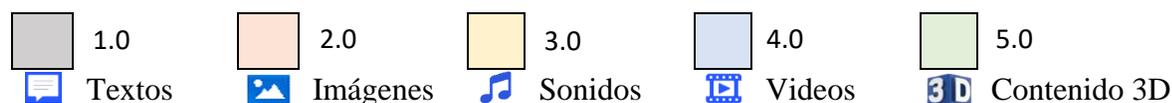
Unity 3D es una herramienta de desarrollo de contenido interactivo de gran versatilidad, consolidándose como una de las más sencillas y potentes del mercado, permite la inclusión y manipulación de contenido multimedia, interactividad y gamificación de los contenidos, y presenta gran compatibilidad al permitir exportar las creaciones a la mayoría de plataformas y sistemas operativos de los dispositivos actuales, tales como Android, IOS, Windows, Xbox, entre otros.

4.1.5.3 Selección de la herramienta de desarrollo

A continuación, en la Tabla 20 se muestra el resultado de la comparación de las herramientas de desarrollo con base en los criterios de evaluación definidos, el color de fondo de cada celda representa el valor de la puntuación asignada para cada criterio según las convenciones indicadas.

Tabla 20. Selección herramienta de desarrollo

	Compatibilidad eBook	Uso de elementos multimedia	Licencia de desarrollo	Licencia publicación	Puntaje
<i>iBook</i>	IOS		Gratuita	Gratuita	4.0
<i>NeoBook</i>	Windows		Gratuita	Gratuita	3.75
<i>Moglue</i>	IOS, Android, con <i>Moglue Viewer</i>		Gratuita	Licencia de Publicación.	3.0
<i>Kotobee Author</i>	Windows, Web, Android, Kindle, IOS, Chromecast		Licencia de Pago Anual	Limitado al Tipo de Licencia	3.25
<i>Unity 3D</i>	Windows, Linux, IOS, Android, Web, PS4, XBOX ONE		Licencia Personal Gratuita	Gratuita	5.0



Tras este análisis comparativo de herramientas de desarrollo, se establece que Unity 3D se ajusta a las necesidades planteadas y obtiene la máxima calificación, por lo cual es seleccionada para la creación y el desarrollo del eBook Interactivo – Compresión de datos (Cedeño & Vargas, 2019).

4.2 Fase de diseño

Con base en los resultados obtenidos en la etapa de análisis, se procede a especificar el diseño del eBook a desarrollar. La fase de diseño consiste en la generación de una guía de desarrollo enriquecida con la dimensión tecnológica y el enfoque pedagógico; incluye la generación de estructuras e interfaces de navegación, y la creación de prototipos o fotomontajes denominados mockups que representan de manera general el eBook interactivo, todo acorde con los recursos tecnológicos seleccionados.

Se realiza el diseño del eBook interactivo tomando como marco de referencia la arquitectura vista en la Sección 2.4.

4.2.1 Vista estática: Diseño Conceptual

El eBook interactivo estará representado en un entorno virtual 3D dentro del cual existirán dos elementos de interacción identificados como el *libro de estudio* y el *PC interactivo*; también existirá una vista central denominada *Perspectiva Inicio* desde la cual es posible visualizar todo el ambiente del eBook como se muestra en la Figura 32.

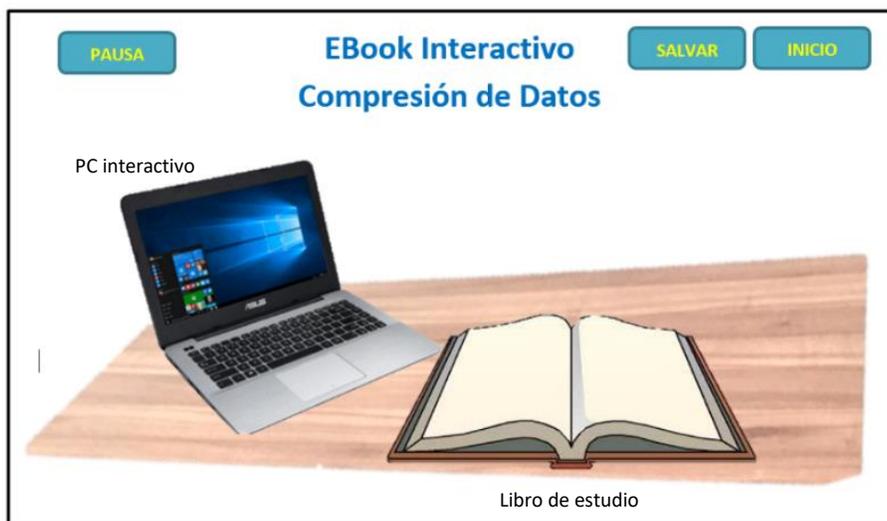


Figura 32. Vista estática. Mockup de la Perspectiva Inicio

En el elemento *libro de estudio* estará plasmado todo el material de instrucción tal como lecturas, diagramas, imágenes y ejemplos interactivos, entre otros, que permitirán al lector consultar y adquirir todo el contenido educativo. Por otro lado, el *PC interactivo* presenta dos perspectivas, en primer lugar, presenta el área donde se desarrollarán las evaluaciones propuestas acerca de la temática de estudio, y en donde se desplegará cada actividad de evaluación según corresponda, ver Figura 33; en segundo lugar, mostrará la pantalla de progreso, en la cual el lector o el estudiante podrá revisar las calificaciones obtenidas en cada prueba como lo muestra la Figura 34 (Cedeño & Vargas, 2019).

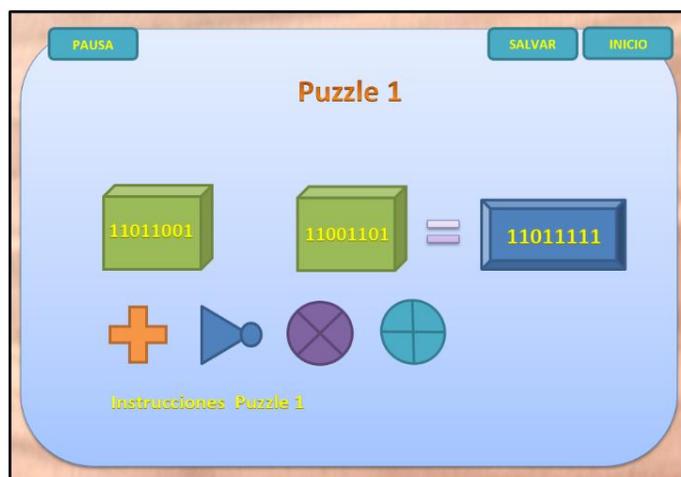


Figura 33. Mockup PC interactivo - actividad de evaluación

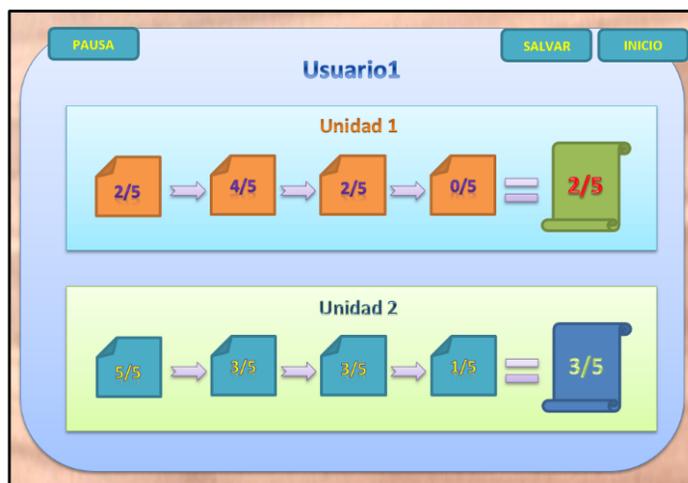


Figura 34. Mockup PC interactivo – pantalla de progreso.

4.2.2 Vista Funcional: Modos de ejecución

En la vista funcional se identifica el elemento principal del eBook interactivo, la aplicación o interfaz de usuario, y se describe paso a paso cómo es su funcionamiento.

El eBook interactivo presentará dos modos de ejecución: el *modo de estudio*, en el cual el usuario debe crear un perfil de ingreso, identificado con un token numérico personal como se muestra en la Figura 35, a través del cual se almacenarán los datos propios del usuario tales como configuraciones, calificaciones y progreso en la lectura del eBook; y el *modo de lectura*, para el que no se requiere la creación o el ingreso mediante el perfil de usuario, por tanto solo permitirá el acceso a la información de consulta sin poder ejecutar las evaluaciones interactivas y no permitirá la opción de almacenar el progreso y configuraciones.

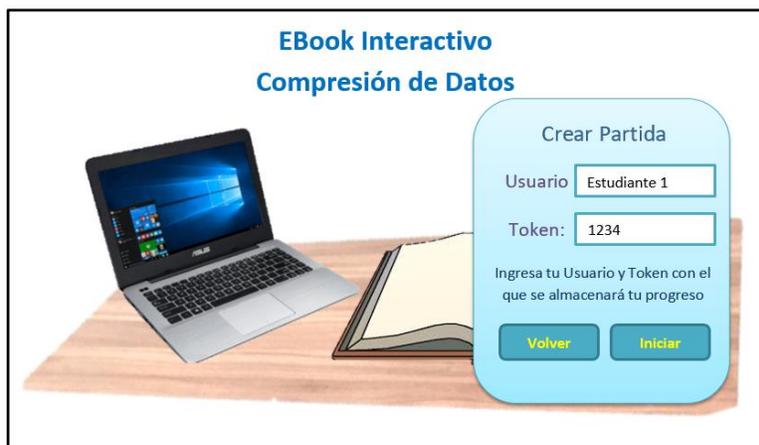


Figura 35. Mockup modo de estudio - creación de perfil con token de usuario.

Adicionalmente, en el *modo de estudio* el lector deberá realizar las actividades de evaluación planteadas de acuerdo a la temática; éstas tendrán una calificación numérica entre 0 y 5 cuyo puntaje será almacenado y sumado en el perfil de usuario, con el fin de determinar la apertura de nuevos contenidos; es decir, cada capítulo presentará un umbral de calificación mínimo para continuar con el siguiente tema, en caso de no contarse con las calificaciones para superar el valor exigido, será necesario revisar las evaluaciones realizadas e intentar mejorar las calificaciones para que el eBook permita continuar con el tema siguiente (Droutsas, Patsilnakos, & Symvonis, Interactive Personalized eBooks for Education, 2018).

4.2.3 Estructuras de navegación

“Las principales estructuras de navegación para el eBook interactivo planteado se identifican continuación:

- *Entorno y Navegabilidad:* El *modo de lectura* únicamente focalizará el libro de estudio dada su funcionalidad limitada a solo consulta, ver Figura 36; al salir de este modo el eBook volverá a la perspectiva de inicio desde donde, a través del menú principal, se podrá seleccionar nuevamente el *modo de lectura* o por el contrario el *modo de estudio*.

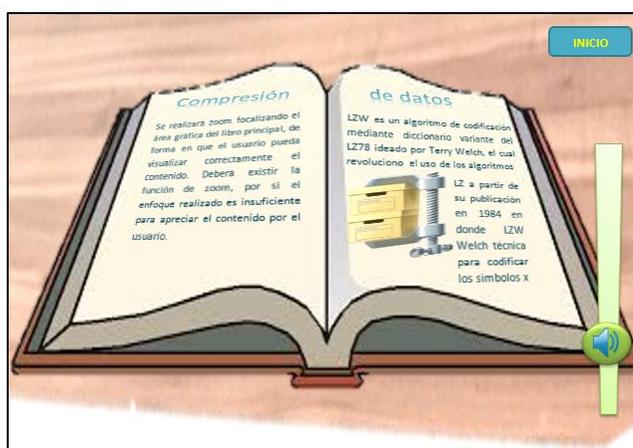


Figura 36. Mockup modo de lectura – focalización en el libro de estudio.

Al seleccionar el *modo de estudio*, es posible focalizar todos los elementos propios del eBook interactivo; desde la perspectiva de inicio solo bastará con interactuar o hacer clic al elemento deseado para desplazarse hacia él; estando en el libro de estudio, es posible navegar entre sus páginas hacia adelante y hacia atrás mediante las flechas de navegación ubicadas en los costados del libro. En la parte alta de la pantalla existirá un botón denominado “Inicio” con el cual es posible regresar a la perspectiva de inicio.

De igual manera, al interactuar o tocar el PC interactivo el eBook desplazará su foco hacia él, y una vez en éste, se desplegará el panel de progreso el cual muestra el avance y las calificaciones para cada actividad de evaluación, y desde el cual es posible ingresar y realizar las actividades para obtener mejores calificaciones o repasar lo aprendido” (Cedeño & Vargas, 2019).

Estas actividades evaluativas también pueden ser accedidas desde el libro de estudio, en el cual en secciones específicas del contenido existirán botones interactivos que permitirán ir al *PC interactivo* abriendo la interactividad correspondiente para su desarrollo, según se muestra en la Figura 37. Una vez en el *PC interactivo*, se podrá presionar el botón “Volver” ubicado en la parte inferior de la pantalla para regresar a la página actual en el *libro de estudio*.



Figura 37. Mockup modo de estudio - botones interactivos en el libro de estudio.

- **Botones de funciones:** el eBook interactivo contará con botones alrededor de la pantalla, accesibles en todo momento de acuerdo al modo de ejecución del eBook, los cuales se aprecian en la Figura 37, y cuya función se describe a continuación:
 - **Botón “Inicio”:** disponible en el *modo de estudio* y en el *modo de lectura*, en el primero regresa el foco de la cámara del eBook a la perspectiva inicial permitiendo interactuar con el PC interactivo o el libro de estudio; en el segundo modo, cierra el *modo de lectura* y regresa al menú principal de la pantalla inicial (ver Figura 38).
 - **Botón “Salvar”:** disponible en todo instante solo en el *modo de estudio* y su función es la de almacenar los datos del progreso actual del lector, tales como página actual, configuración de volumen, luminosidad, calificación actual y demás variables almacenadas en el perfil.

- *Botón “Pausa”*: visible y funcional solo en el *modo de estudio* y permite acceder al menú de pausa.
 - *Botón “Volumen”*: funcional en todas las instancias del eBook y permite controlar el volumen del sonido general del eBook interactivo. Consiste en un botón flotante que al ser presionado habilita la funcionalidad de slider para aumentar o disminuir el volumen, desplazando el botón según corresponda. En caso de solo presionar y soltar el botón, funciona como un control ON/OFF habilitando o deshabilitando el sonido del eBook.
- *Menú Principal*: Corresponde al menú establecido en la pantalla inicial, cuyo mockup se aprecia en la Figura 38, éste contiene las opciones de acceso y usabilidad Crear Usuario, Cargar Progreso, Consulta Rápida, Créditos y Salir, así como los diferentes submenús que permiten la correcta ejecución de estas acciones. La estructura de navegación para sus funciones se muestra en el esquema de la Figura 39.



Figura 38. Mockup pantalla inicial y menú principal.

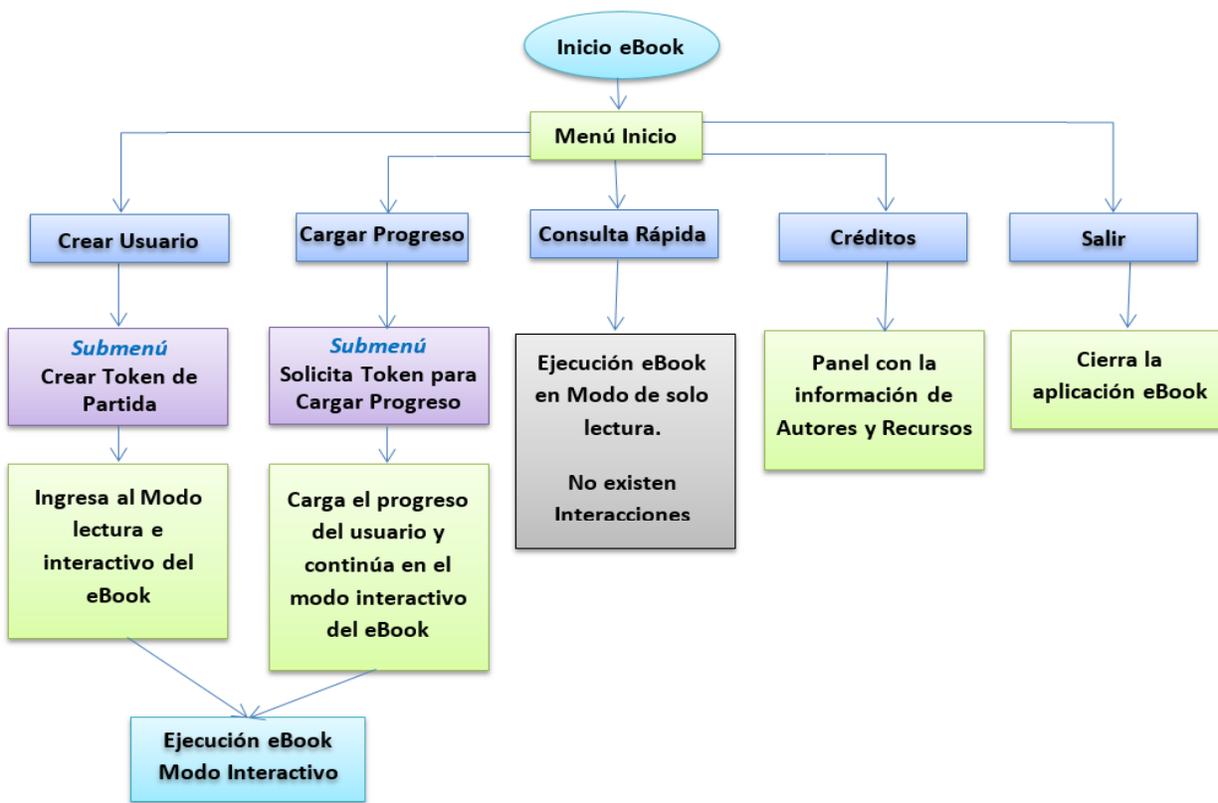


Figura 39. Esquema de navegación Menú Principal

- *Menú de Pausa:* Consiste en un menú de opciones básicas tales como Cambio de Página, Opciones y Salir (opción de regresar al menú principal), accesible desde el *modo de estudio*; como sus nombres lo indican, estas opciones presentan sus correspondientes submenús, a través de los cuales se permitirá respectivamente: ir a la página deseada del eBook; modificar las opciones de luminosidad, volumen y la distancia de lectura de preferencia; e ir al menú principal. La Figura 40 muestra el flujo de navegación de este menú.

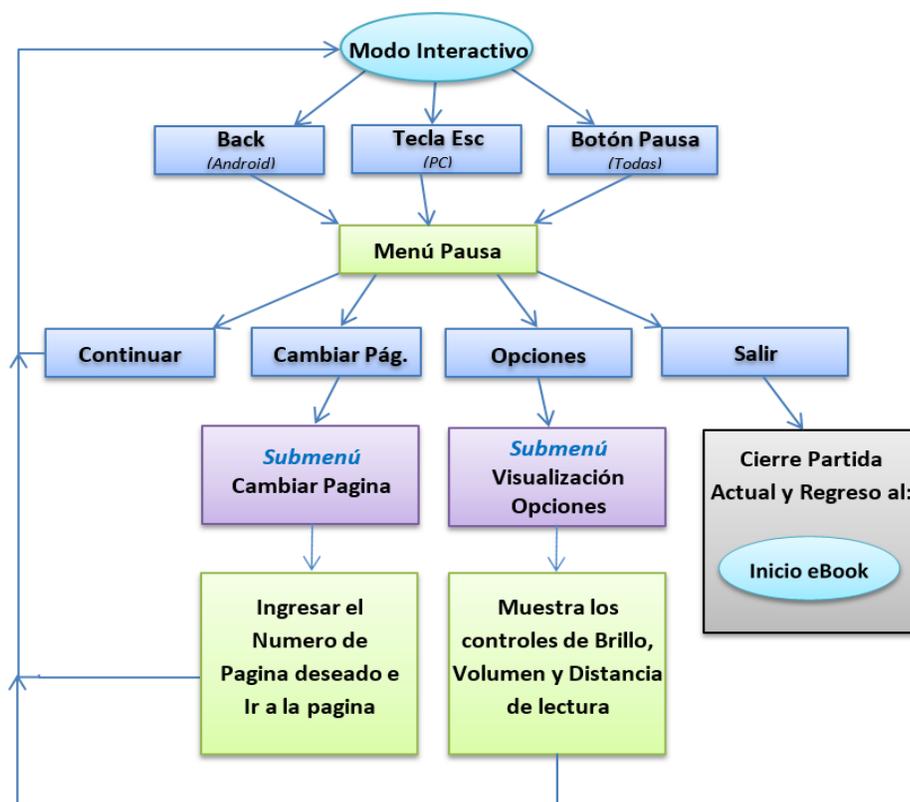


Figura 40. Esquema de navegación Menú de Pausa.

- *Tabla de contenido:* Está ubicada al inicio del *libro de estudio*, su función es establecer una guía de orden acerca del contenido del mismo y permitir el acceso directo a cada uno de los contenidos de estudio; su estructura corresponde al contenido de enseñanza establecido en la Sección 4.1.3.

En el desarrollo final, las temáticas de nivel específico solo serán visibles una vez seleccionada la temática principal, estas serán elementos interactivos que al ser seleccionados direccionarán a la página correspondiente.

4.3 Desarrollo

En esta fase se desarrolló el eBook interactivo considerando los resultados obtenidos en las fases anteriores (análisis y diseño), utilizando la herramienta de software escogida, Unity 3D. Por consiguiente, en esta sección se presenta evidencia del prototipo inicial desarrollado, se exponen las pruebas (funcionalidad - usabilidad) realizadas con un grupo focal a distintas versiones del prototipo, y por último se enumeran las mejoras propuestas para este.

4.3.1 Prototipo del eBook

La construcción del prototipo del eBook interactivo es llevada a cabo en diversas etapas en las que se desarrollan las características indicadas en la fase de diseño (ver Sección 4.2).

4.3.1.1 Entorno y elementos de interacción

Tal como se especificó en el diseño, el eBook interactivo corresponde a un entorno en 3D que está ambientado con objetos propios de un aula educativa y en el que se incluyen los dos elementos principales de interacción planteados:

- *PC interactivo*: corresponde a un modelo 3D de un PC portátil ubicado en el escritorio principal según se muestra en la Figura 41, el cual contendrá todas las interactividades y el control de la puntuación del usuario.

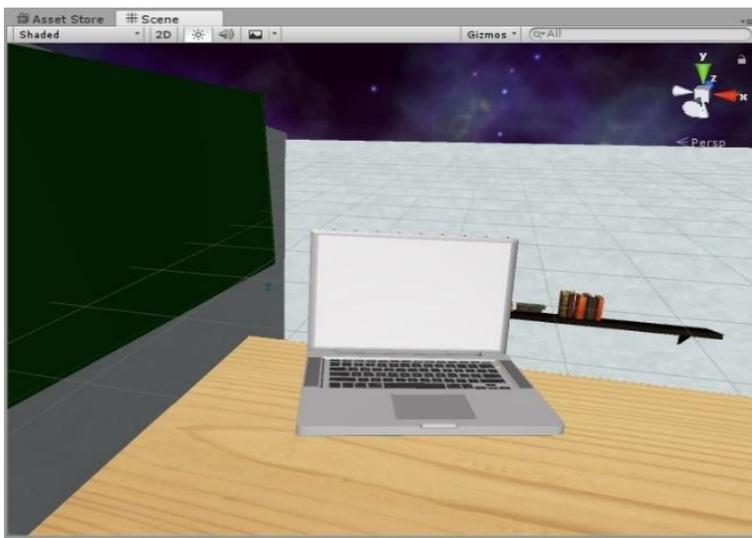


Figura 41. Entorno y elementos de interacción - PC interactivo.

- *Libro de estudio:* Consiste en un elemento 3D con el que se representa un libro tal como se aprecia en la Figura 42, el cual contendrá todo el material educativo y de lectura del eBook; se ha desarrollado usando la librería o Assett MegaBook, que contiene todas las estructuras y funciones para el control y movimiento de las páginas. Esta librería es licenciada y fue adquirida para el desarrollo del presente proyecto.



Figura 42. Entorno y elementos de interacción - Libro de estudio.

- *Interfaz de Usuario:* La UI contiene los elementos interactivos descritos en el diseño, los cuales fueron construidos con las herramientas de UI de Unity que permiten la creación de una estructura tipo panel fijo denominado *Canvas*, ajustado en la parte delantera de la pantalla según se muestra en la Figura 43. En esta estructura son creados los elementos que hacen parte de la UI tales como menú inicio, menú de pausa, botones de cambio de página, botón de control de volumen, etc., cuya habilitación es realizada según sea requerida.

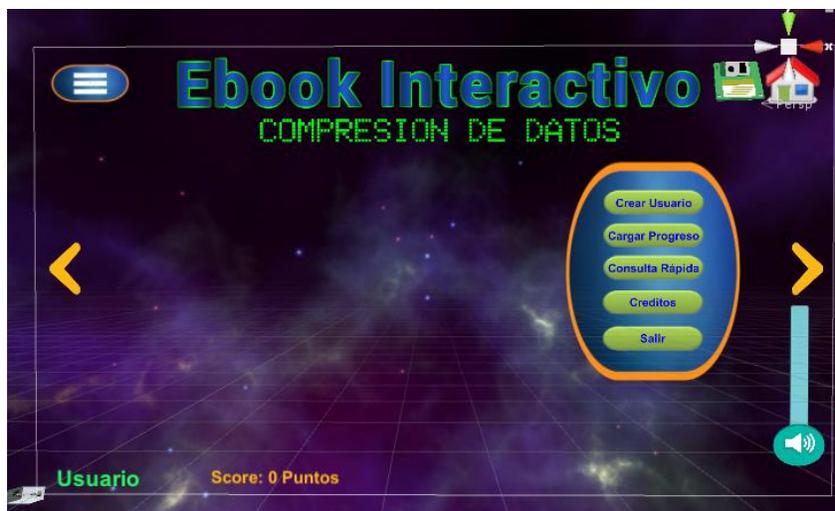


Figura 43. Entorno y elementos de interacción - Interfaz de usuario

4.3.1.2 Cámara y navegabilidad

Debido al desarrollo de contenido tipo 3D y a su creación en un motor de videojuegos, en el eBook interactivo existirá un elemento denominado Cámara Principal (en inglés Main Camera), que corresponde a un marco delimitador de cuya posición y ubicación dependerán los elementos que visualiza el lector o usuario durante la ejecución del eBook (ver Figura 44).



Figura 44. Cámara y Navegabilidad.

Debido a lo anterior, la navegabilidad entre los elementos del eBook interactivo estará netamente ligada a las variables de posición y rotación de la Main Camera, por lo cual se desarrolló un script en C# denominado “CamControl.cs” en el que se incluyen las funcionalidades para obtener el elemento señalado por el usuario y la respectiva función de desplazamiento hasta el punto objetivo (ver Anexo A-1).

4.3.1.3 Scripts de interacciones

El eBook interactivo contiene gran cantidad de elementos que permiten interactuar al usuario interactuar, esto es posible gracias a los scripts de interacción desarrollados, dentro de los cuales cabe destacar (ver Anexo A-2):

- Funciones para reiniciar interactividad - FuntRetry.cs
- Funciones para arrastrar un Objeto - GrabObject.cs

4.3.1.4 Estado, configuración y almacenamiento de perfiles

El estado del eBook, las configuraciones y el progreso del usuario, son evaluados y administrados por el elemento de UI Canvas, que a su vez se encarga de mostrar esta información en pantalla y de la administración de la misma, incluyendo el almacenamiento de esta de manera local en un archivo .dat totalmente accesible por el eBook.

Toda la gestión de las variables mencionadas se realiza mediante las funciones desarrolladas en el script denominado MainCanvas.cs, del cual vale la pena resaltar las rutinas para el almacenamiento y carga de la información de usuario (ver Anexo A-3).

4.4.1.5 Créditos y comunicación

El eBook interactivo presenta un botón de créditos ubicado en el menú principal, que lleva al usuario o estudiante a una pantalla con información acerca del eBook, de las herramientas de desarrollo y de los autores (ver Figura 45).

En esta sección de créditos se alojarán también los datos de contacto y comunicación con los autores del eBook, así como la URL de un foro de discusión en Internet dedicado al eBook, en el que los usuarios puedan informar acerca de errores en el contenido, expresar sus comentarios y realizar preguntas directamente al autor o a otros lectores (Binas, Štancel, Novák, & Michalko, 2012). Para el eBook interactivo desarrollado se habilitó el foro disponible en la URL: https://nowcomment.com/groups/eBook_Interactivo_Compresion_Datos.



Figura 45. Pantalla créditos y comunicación.

4.3.2 Pruebas al prototipo con grupo focal

Las pruebas de usabilidad son un método de alta fiabilidad que permite determinar la experiencia de usuario con el producto; estas pruebas son aplicables a todo tipo de nuevos productos tales como elementos de consumo, nuevos dispositivos y principalmente en todos los desarrollos de tipo software tales como páginas web y aplicaciones en general (Sánchez Morales, Mezura-Godoy, & Benítez-Guerrero, 2018)

Las pruebas de usabilidad comprenden tres técnicas básicas por medio de las cuales se evalúa el nuevo producto, en este caso específico un eBook interactivo, para ello se selecciona un grupo focal de usuarios directamente relacionados con el uso final del producto a evaluar, a quienes se solicitará el empleo del producto y la ejecución de las tareas básicas permitidas por el mismo y sobre quienes se evaluarán las siguientes pruebas de usabilidad:

- *Pruebas a usuarios:* corresponde a pruebas o evaluaciones incluidas en el producto a evaluar las cuales de acuerdo con las calificaciones obtenidas representaran la primera medición de usabilidad del producto.
- *Observación:* Durante el uso y ejecución de las pruebas a usuarios descritas en el ítem anterior, el equipo de diseño y desarrollo observará y tomará nota del flujo de la interacción por parte de los participantes, haciendo énfasis en los errores y dificultades que encuentren los usuarios, con lo cual se establecerá la segunda medición de usabilidad.
- *Preguntar a los usuarios:* comprende la realización de cuestionarios, entrevistas y/o encuestas sobre la opinión y satisfacción tras el uso de la herramienta evaluada; el cuestionario a aplicar puede ser personalizado con base en el tipo de producto a evaluar, o puede tratarse de un test predefinido para este fin.

Para evaluar la usabilidad del eBook interactivo fue necesario realizar una prueba (ver anexo C) con el fin de medir la experiencia del usuario. La prueba de usabilidad tuvo 2 iteraciones en momentos diferentes, con 2 grupos focales distintos:

Iteración 1: Se emplea la prueba de usabilidad de manera virtual con 34 estudiantes del programa de Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia.

Iteración 2: Se emplea la prueba de usabilidad de manera virtual con 37 estudiantes del programa de Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia.

4.3.3 Resultados de las pruebas al prototipo

A continuación, se relacionan algunas observaciones y propuestas de mejora que se encontraron en los resultados de la prueba de usabilidad en ambas iteraciones (ver anexo D).

Las observaciones permitieron reconocer y corregir errores puntuales del eBook interactivo, además de recibir recomendaciones hechas por los usuarios que permitieron la optimización del eBook.

Observaciones positivas

- Los estudiantes comprendieron el manejo del eBook, además las herramientas audiovisuales generaron un reconocimiento intuitivo por parte de los estudiantes.
- Se resalta el lenguaje claro y la sencilla interacción con el eBook.
- La ubicación de los elementos del eBook permitieron la interacción intuitiva con el usuario.
- Se resaltó el adecuado contenido teórico comprendido en el eBook, cantidad de imágenes y herramientas de este.

Propuestas de mejora

- Se sugiere verificar errores ortográficos.
- Se recomienda implementar atajos mediante el uso del teclado.
- Señalización insuficiente para el uso de elementos interactivos.
- Lentitud en el cambio de página.
- Mejorar la aproximación de los valores numéricos resultantes en los ejercicios.

5. Metodología: implementación y evaluación del eBook interactivo

La metodología utilizada para el desarrollo del eBook interactivo está inspirada en el modelo de diseño instruccional ADDIE, cuyo nombre es un acrónimo de los pasos clave o fases que conforman este modelo: Análisis, Diseño, Desarrollo, Implementación y Evaluación. En este capítulo, en la Sección 5.1 se describe la implementación del eBook interactivo, en la Sección 5.2 su evaluación, y en la Sección 5.3 se realiza un análisis de los resultados arrojados por estas dos fases.

5.1 Implementación del eBook interactivo

La metodología para llevar a cabo la implementación y evaluación del eBook es la investigación cuantitativa descriptiva no experimental (Flores et al., 2017), cuya característica principal es la comparación del público objetivo antes y después de la intervención educativa, mediante la realización de pruebas *pre-test* y *post-test* (Bruce, 2012). El público objetivo estuvo conformado por 64 estudiantes que cursaban la asignatura *Teoría de la información y sistemas de comunicación* de la Universidad Nacional de Colombia.

En esta sección se describen las actividades que conforman la fase de implementación del eBook interactivo: aplicación de encuesta de contexto, identificación y mitigación de las amenazas a la validez, diseño de prueba de *pre/post test* y de prueba de usabilidad, y diseño y ejecución de la estrategia instruccional.

5.1.1 Aplicación de encuesta de contexto

Se aplica una encuesta de contexto (Tabla 21) sobre la población objetivo, de modo que se pueda conocer más información de los estudiantes (edad, semestre en curso, pregrado al que pertenece) y el nivel de atracción por el objeto de estudio.

Tabla 21. Encuesta de contexto.

Encuesta de contexto	
Fecha:	
Nombre:	
Código Estudiantil:	
Por favor contestar las siguientes preguntas:	
Pregunta	Respuesta
¿Cuál es su edad?	
¿A qué pregrado pertenece?	
¿Qué semestre cursa actualmente?	
Indique en qué medida se siente atraído por el aprendizaje de los conceptos de la teoría de la información.	1 - En Absoluto, 2 – Poco, 3 - Medianamente, 4 – Algo, 5 - Totalmente

5.1.2 Identificación y mitigación de las amenazas a la validez

Las investigaciones cuantitativas descriptivas están sujetas a posibles sesgos producidos por su propia naturaleza, que involucra el trabajo con grupos conformados de manera aleatoria. Para reducir los sesgos se debe hacer una adecuada identificación de las amenazas a la validez y se deben establecer las estrategias de mitigación correspondientes, de modo que los resultados obtenidos al finalizar las pruebas *pre/post test* se acerquen más a la realidad, para así afirmar o negar las hipótesis planteadas sobre la intervención educativa (Bruce, 2012).

Por lo tanto, con el fin de reducir los sesgos, en la Tabla 22 se describen las posibles amenazas a la validez y las estrategias tomadas para mitigar su impacto (Rodríguez & Vargas, 2020).

Tabla 22. Amenazas a la validez en una investigación cuantitativa descriptiva y estrategias de mitigación.

Amenaza	Descripción	Estrategia de mitigación
Histórica	Evento entre el <i>pre-test</i> y el <i>post-test</i> que aumenta o disminuye los puntajes del <i>post-test</i> comparado al <i>pre-test</i> .	El <i>pre-test</i> se desarrolla al iniciar el periodo de implementación y el <i>post-test</i> al terminar el mismo, de este modo se evita la aparición de eventos educativos que alteren en gran medida los resultados.

Maduración	Surge porque los participantes en el estudio avanzan en edad, se hacen más conocedores o han visto más temas relacionados.	La implementación se ejecuta en un corto periodo de tiempo, evitando así la maduración de los participantes durante la implementación.
Prueba	Surge cuando el <i>pre-test</i> provoca un cambio en los resultados, debido a que los participantes reconocen temas de estudio para aprender por si mismos al notar que no los saben.	Se procura realizar la acción formativa y pruebas en un periodo corto de tiempo, en dos semanas, con el fin de disminuir la posibilidad de estudios complementarios por parte de los estudiantes.
Instrumentación	Ocurre debido al cambio en el instrumento de evaluación de los participantes desde el momento del <i>pre-test</i> hasta el momento del <i>post-test</i> .	Las pruebas de <i>pre-test</i> y <i>post-test</i> contienen las mismas preguntas, de modo que se evalúan los mismos conocimientos tanto al principio como al final de la acción formativa, manteniendo así el mismo instrumento de medida.
Mortalidad	Surge cuando hay un abandono no aleatorio entre el <i>pre-test</i> y el <i>post-test</i> . Por lo tanto, las diferencias en las pruebas de <i>post-test</i> podrían deberse a los varios tipos de abandono y no al eBook interactivo.	Realizar las pruebas durante un periodo corto de tiempo y evitar ejecutarlas al iniciar o finalizar el semestre, donde se suele presentar mayor inasistencia por parte de los estudiantes.

Fuente: Adaptado de la descripción de amenazas de validez de (Rodríguez & Vargas, 2020).

5.1.3 Diseño de prueba *pre-test/post-test* y de prueba de usabilidad

Con base en los objetivos de aprendizaje definidos y en los beneficios de los eBooks interactivos en entornos educativos (Cordón García & Lopes, 2012), se diseñó una prueba *pre/post test* que consta de 15 preguntas, agrupadas de acuerdo con los siguientes módulos de aprendizaje (ver Sección 4.1.3): conceptos básicos de la compresión de datos, introducción a la teoría de la información, técnicas de compresión sin pérdidas y técnicas de compresión con pérdida. La prueba completa de *pre/post test* se encuentra en el Anexo B.

Las preguntas se obtuvieron con base en un examen de prueba de la Universidad de Sydney (Howlett, 2003) y los libros Handbook of Data Compression (Salomon & Motta, 2010) y Introduction to Data Compression (Sayood, 2006). También se contó con el asesoramiento pedagógico de docentes conocedores de la temática de estudio.

La prueba *pre-test/post-test* se diseñó como una evaluación de preguntas de opción múltiple, la cual tuvo en cuenta los criterios de elaboración y estructuración propuestos por Haladyna, Downing & Rodriguez (2002) para este tipo de preguntas.

El diseño de la prueba de usabilidad se describe en la Sección 5.2.2, y la prueba se encuentra en el Anexo C.

5.1.4 Diseño y ejecución de la estrategia instruccional

La estrategia instruccional se lleva a cabo durante seis sesiones: en la primera sesión se identifican los conocimientos iniciales (evaluación *pre-test*), en las siguientes cuatro sesiones se cubre el contenido temático utilizando el eBook interactivo desarrollado (ver Anexo F), y en la última sesión se concluye con la identificación de conocimientos finales (evaluación *post-test*).

Desde la Tabla 23 hasta la Tabla 28 se presenta la estrategia instruccional de cada una de las sesiones.

Tabla 23. Estrategia instruccional sesión 1.

Sesión 1.		Duración de la sesión: 100 min.
Material instruccional: eBook interactivo <i>Compresión de datos, pre-test</i>		
Temas abordados:		
Secuencia de actividades		
No.	Actividad	Tiempo (minutos):
1.	Socialización y descripción de la intervención educativa.	20
2.	Instalación de eBook interactivo.	10
3.	Introducción al uso del eBook interactivo.	10
4.	Identificación de conocimientos iniciales mediante la aplicación del <i>pre-test</i> .	60

Tabla 24. Estrategia instruccional sesión 2.

Sesión 2.		Duración de la sesión: 90 min.
Material instruccional: eBook interactivo <i>Compresión de datos</i> .		
Temas abordados: conceptos básicos de la compresión de datos e introducción a la teoría de la información.		
Secuencia de actividades		
No	Actividad	Tiempo (minutos):
1.	Utilización del eBook interactivo.	90 minutos
2.	Diligenciamiento por parte de los estudiantes de la prueba de usabilidad.	Abierta Online - 1 semana
3.	Envío por parte de los estudiantes de la evidencia del progreso en el eBook interactivo.	Abierta Online - 1 semana

Tabla 25. Estrategia instruccional sesión 3.

Sesión 3.		Duración de la sesión: 90 min.
Material instruccional: eBook interactivo <i>Compresión de datos</i> .		
Temas abordados: técnicas de compresión sin pérdidas		
Secuencia de actividades		
No.	Actividad	Tiempo (minutos):
1.	Utilización del eBook interactivo.	90 minutos
2.	Diligenciamiento por parte de los estudiantes de la prueba de usabilidad.	Abierta Online - 1 semana
3.	Envío por parte de los estudiantes de la evidencia del progreso en el eBook interactivo.	Abierta Online - 1 semana

Tabla 26. Estrategia instruccional sesión 4.

Sesión 4.		Duración de la sesión: 90 min.
Material instruccional: eBook interactivo <i>Compresión de datos</i> .		
Temas abordados: técnicas de compresión sin pérdidas		
Secuencia de actividades		
No.	Actividad	Tiempo (minutos):
1.	Utilización del eBook interactivo.	90 minutos
2.	Diligenciamiento por parte de los estudiantes de la prueba de usabilidad.	Abierta Online - 1 semana
3.	Envío por parte de los estudiantes de la evidencia del progreso en el eBook interactivo.	Abierta Online - 1 semana

Tabla 27. Estrategia instruccional sesión 5.

Sesión 5.		Duración de la sesión: 90 min.
Material instruccional: eBook interactivo <i>Compresión de datos</i> .		
Temas abordados: técnicas de compresión con pérdida.		
Secuencia de actividades		
No.	Actividad	Tiempo (minutos):
1.	Utilización del eBook interactivo.	90 minutos
2.	Diligenciamiento por parte de los estudiantes de la prueba de usabilidad.	Abierta Online - 1 semana
3.	Envío por parte de los estudiantes de la evidencia del progreso en el eBook interactivo.	Abierta Online - 1 semana

Tabla 28. Estrategia instruccional sesión 6.

Sesión 6.		Duración de la sesión: 90 min.
Material instruccional: <i>post-test</i>		
Temas abordados:		
Secuencia de actividades		
No.	Actividad	Tiempo (minutos):
1.	Identificación de conocimientos finales mediante la aplicación del <i>post-test</i> .	90

(Cedeño & Vargas, 2021)

5.2 Evaluación del eBook interactivo

Una vez finalizada la implementación, se cuenta con los elementos necesarios para realizar la evaluación del eBook interactivo en dos aspectos: (1) Evaluación de la efectividad de la acción formativa en el público objetivo, y (2) Evaluación de la usabilidad del eBook interactivo por parte del público objetivo (Cedeño & Vargas, 2021). A continuación, se hace una descripción de estas actividades.

5.2.1 Evaluación de la efectividad de la acción formativa en el público objetivo

Se realizan pruebas estadísticas que comparan los puntajes obtenidos por el público objetivo en el *pre-test* y el *post-test*, utilizando las pruebas *t-student* o *Wilcoxon* ya que son las

apropiadas para el tamaño de la muestra, considerando que el público objetivo está compuesto por más de 50 estudiantes (Flores-Ruiz et al., 2017).

Prueba estadística

La prueba se aplicó a 64 unidades representadas por 64 estudiantes. En la Tabla 29 se muestran los valores de las medias descriptivas obtenidos mediante las pruebas de validez estadística para los datos de la muestra. Cada una de las dos pruebas (*pre-test* y *post-test*) se calificaron de 0 a 5, observando una media de mayor valor en el *post-test*.

Tabla 29. Estadísticos descriptivos del *pre-test* y el *pos-test*.

Valor estadístico		Valores <i>Pre-test</i>	Valores <i>Post-test</i>
Mínimo		0,75	1,50
1er. cuartil		1,50	3,00
Mediana		2,00	4,00
Media		2,00	3,70
3er. cuartil		2,50	4,25
Máximo		3,75	5,00
Nro.	Válidos	64	64
	Perdidos	0	0

En la Figura 46 se muestran los diagramas de caja para cada prueba realizada al público objetivo, encontrando un incremento entre las notas del *pre-test* y el *post-test*.

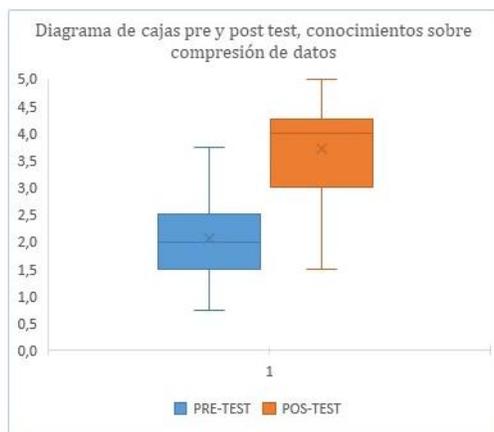


Figura 46. Diagrama de cajas para los puntajes obtenidos en el *pre-test* y el *post-test*.

A continuación, se realizan las inferencias de las mediciones para el público objetivo, de modo que se pueda determinar si mejoró en el *post-test* con respecto a la nota del *pre-test*. Para ello se aplicarán pruebas estadísticas para diferencias de medias con un nivel de confianza del 95%. Esta es una prueba de una cola, ya que para la hipótesis alternativa es de interés saber si los resultados del *post-test* fueron mayores que los del *pre-test*.

Para esto se emplea la prueba estadística *t-student* que permite calcular la significancia de los resultados obtenidos, además esta prueba es indicada para muestras mayores a 50 individuos adaptándose así a la cantidad de estudiantes evaluados en este trabajo. Sin embargo, para poder hacer uso de esta prueba se debe demostrar la normalidad de la muestra. En el caso de no superar dicha prueba se hará uso de la prueba *Wilcoxon*.

Con el fin de comprobar que los datos obtenidos en la muestra se ajustan a una prueba *t-student*, se realiza una prueba de normalidad (*Kolmogórov-Smirnov*). Los resultados obtenidos para la prueba de normalidad son expuestos en la Tabla 30.

Tabla 30. Resultados prueba de normalidad *Kolmogórov-Smirnov*.

<i>Kolmogórov-Smirnov</i>	Valor-P
PRE-TEST	0,002
POST-TEST	0,001

Se puede observar que tanto los datos del *pre-test* como los del *post-test* no siguen una distribución normal, ya que se obtuvieron valores-P de 0,002 y 0,001, respectivamente, y estos son menores a 0,05. Por lo tanto, la hipótesis nula se rechaza. Dado que las variables no cumplen el supuesto para aplicar la prueba *t-student*, se utiliza la prueba alternativa no paramétrica de *Wilcoxon*, para la cual se obtuvieron los resultados mostrados en la Tabla 31.

Tabla 31. Resultados prueba de *Wilcoxon*.

Medida <i>Wilcoxon</i>	Valor
Valor-P	0,000

De acuerdo con los resultados de la Tabla 31, se observa que $P=0$ es decir menor a 0,05, por tanto, se rechaza la Hipótesis nula (H_0) y se acepta la Hipótesis alterna (H_a), lo que significa que la mediana de las diferencias entre el *pre-test* y *post-test* es estadísticamente significativa en la prueba de *Wilcoxon* para muestras relacionadas. En el mismo sentido, se encontraron 59 diferencias positivas, 3 negativas y 2 empates, lo que significa que la diferencia es favorecedora como se puede evidenciar en la Figura 47.



Figura 47. Prueba de rangos de Wilcoxon.

Por consiguiente, se concluye que hay diferencias positivas de las medianas del *pre-test* y el *post-test*, es decir, que la hipótesis de igualdad de medianas se rechaza, por lo cual existe evidencia estadística significativa para decir que las calificaciones del *post-test* fueron mayores que las del *pre-test*. En conclusión, hay evidencia estadística significativa para afirmar que el uso del eBook interactivo fue efectivo en el proceso de intervención pedagógica con los estudiantes (Cedeño & Vargas, 2021).

5.2.2 Evaluación de la usabilidad del eBook interactivo por parte del público objetivo.

La evaluación de usabilidad del eBook interactivo consistió en aplicar una encuesta de 46 preguntas de opción múltiple al grupo de 64 estudiantes, después de haber hecho uso del eBook interactivo. Las preguntas se organizaron en 8 categorías basadas en las heurísticas de Nielsen Holdings (Pribeanu, 2017):

- *Visibilidad del estado del sistema:* El eBook interactivo deberá informar su *estado actual* en todo momento, de ser posible, o permitir el acceso a este de manera sencilla.
- *Uso del lenguaje de los usuarios:* El eBook deberá emplear un lenguaje pertinente para los usuarios a quienes va dirigido, facilitando la comprensión de las funcionalidades, mensajes y contenidos.
- *Control y libertad para el usuario:* Complementando los factores amigables del eBook interactivo, es necesario que permita al usuario las opciones de deshacer los cambios realizados y reiniciar las actividades interactivas, sin necesidad de realizar operaciones complejas o reiniciar el eBook.
- *Consistencia y estándares:* El eBook deberá presentar patrones en su funcionamiento, manteniendo procedimientos definidos para la activación de las diferentes funciones y tareas, evitando de esta manera ambigüedades durante su uso.
- *Prevención de errores:* Prevenir errores es más importante que crear buenos mensajes de error. Las acciones que predisponen a un error deben minimizarse o, en todo caso, preguntarse si el usuario las llegará a realizar.
- *Minimizar la carga de la memoria del usuario:* El sistema debe reducir la información que el usuario necesita retener y mostrarla a través de objetos, acciones u opciones. El usuario no necesita recibir información previa. Las instrucciones para usar el sistema deben ser visibles o accesibles para el usuario si es necesario.
- *Flexibilidad y eficiencia de uso:* Los aceleradores o atajos aumentan la velocidad de interacción para el usuario experto, de modo que el eBook interactivo puede atraer tanto a usuarios novatos como experimentados. Por lo tanto, es pertinente que este permita personalizar acciones para facilitar su uso.
- *Diálogos estéticos y diseño minimalista:* La interfaz no debe incluir información que sea irrelevante o que se use con poca frecuencia, ya que cada pieza adicional de información compite con las unidades de información relevantes en la conversación y reduce su visibilidad relativa.

Las respuestas se basaron en la escala Likert (McLeod, 2019), por lo tanto, las opciones de respuesta posibles fueron: 1- Totalmente en desacuerdo; 2- En desacuerdo; 3- Ni de acuerdo, ni en desacuerdo; 4- De acuerdo; 5- Totalmente de acuerdo.

La Figura 48 presenta un resumen de las respuestas a todas las preguntas de la prueba de usabilidad (en el Anexo C se encuentra el informe de usabilidad detallado). Se observa que hubo una percepción positiva por parte de los estudiantes con respecto a la usabilidad del eBook, ya que el mayor número de respuestas estuvo en las opciones: *Totalmente de acuerdo* y *De acuerdo*.

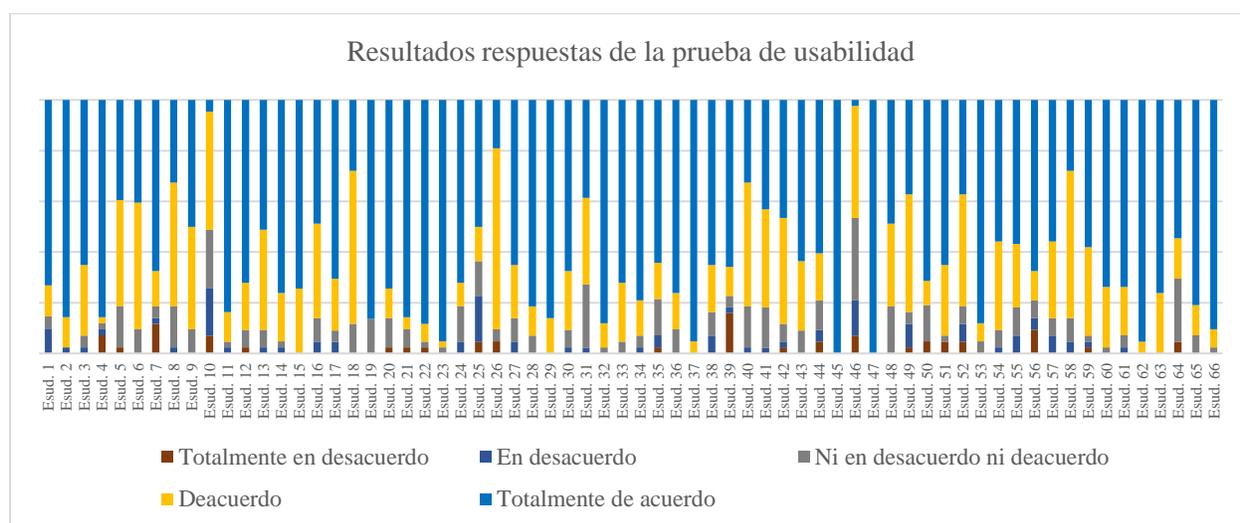


Figura 48. Resultados de evaluación de usabilidad.

En la *Figura 49* se presenta el promedio por categoría de las respuestas a la prueba de usabilidad.

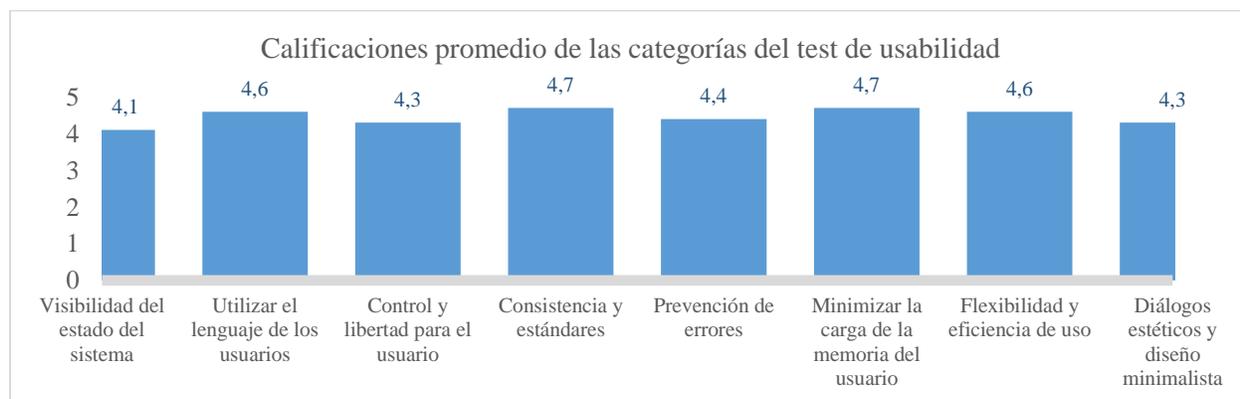


Figura 49. Promedio de las calificaciones por categoría de las preguntas del test de usabilidad.

El promedio de las calificaciones promedio de las categorías de la prueba de usabilidad fue de 4,5, lo que permite afirmar que el eBook genera una percepción de usabilidad positiva (Bravo Paniagua & Valenzuela Gonzáles, 2019).

Las categorías con mayor calificación promedio fueron las siguientes:

- *Uso del lenguaje de los usuarios*: la calificación promedio fue de 4,7, evidenciando que según los estudiantes el eBook empleó un lenguaje pertinente para ellos, facilitando la comprensión de las funcionalidades, mensajes y contenidos.
- *Consistencias y estándares*: la calificación promedio fue de 4,8, evidenciando que para los estudiantes el eBook presenta patrones en su funcionamiento, manteniendo procedimientos definidos para la activación de las diferentes funciones y tareas, evitando de esta manera ambigüedades durante su uso.
- *Minimizar la carga de la memoria del usuario*: la calificación promedio fue de 4,8, evidenciando que según los estudiantes el eBook contiene la mínima información que necesitan retener, y las instrucciones para usar el eBook son visibles o accesibles para ellos.
- *Diálogos estéticos y diseño minimalista*: la calificación promedio fue de 4,7, evidenciando que la interfaz no incluye información irrelevante o de poco uso.

Por otra parte, la categoría *prevención de errores* fue la que obtuvo la puntuación más baja (calificación promedio de 3,9), lo que indica que es necesario minimizar aún más las acciones que predisponen a errores.

En conclusión, el resultado de las calificaciones respecto a la usabilidad del eBook interactivo por parte de los estudiantes fue positivo. No obstante, una recomendación para futuros trabajos es la mejora en la calificación de la categoría de *prevención de errores* (Cedeño & Vargas, 2021).

5.2.3 *Análisis de resultados*

Con respecto a la evaluación de la efectividad de la acción formativa en el público objetivo, las pruebas estadísticas indican que existe evidencia significativa para afirmar que las notas del *post-test* fueron mayores que las del *pre-test*, siendo las del *post-test* lo suficientemente

satisfactorias como para cumplir con los objetivos de aprendizaje planteados. En conclusión, fue efectiva la intervención pedagógica con el apoyo del eBook interactivo. Este resultado respalda la afirmación de que cuando se genera interacción con herramientas tecnológicas, hay un aumento de la motivación y la interiorización del contenido (Grabowski et al., 2020).

Al analizar detalladamente el promedio (redondeado) de respuestas correctas del *post-test* en cada módulo temático (Tabla 32), se aprecia que:

- En la temática *conceptos básicos de la compresión de datos*, los estudiantes lograron duplicar el número de respuestas correctas en el *post-test*.
- En la temática *introducción a la teoría de la información*, los estudiantes alcanzaron un puntaje superior de respuestas correctas en el *post-test*.
- En la temática *técnicas de compresión sin pérdidas – métodos estadísticos y compresión basada en contexto*, se encuentra un aumento representativo en el número de respuestas correctas en el *post-test*.
- En la temática *técnicas de compresión sin pérdidas - métodos de diccionario*, los estudiantes no aumentaron las respuestas correctas en el *post-test*, incluso hubo dos puntos menos en el *post-test*, por lo que se recomienda reforzar los contenidos en este tópico.
- En la temática *técnicas de compresión con pérdida*, los estudiantes logran un mayor número de respuestas correctas en el *post-test*.

Tabla 32. Promedio (redondeado) de respuestas correctas por módulo, en las pruebas *pre-test* y *post-test*.

Módulo temático	Respuestas correctas		Total, preguntas por módulo
	<i>Pre-test</i>	<i>Post-test</i>	
Conceptos básicos de la compresión de datos	29	64	2
Introducción a la teoría de la información	30	53	3
Técnicas de compresión sin pérdidas - métodos estadísticos y Compresión basada en contexto	23	53	6
Técnicas de compresión sin pérdidas - métodos de diccionario	26	24	2
Técnicas de compresión con pérdida	26	41	3

Con respecto a la evaluación de la usabilidad del eBook interactivo por parte del público objetivo, la prueba correspondiente arroja un resultado positivo, especialmente en las categorías de *uso del lenguaje de los usuarios, consistencias y estándares, minimizar la carga de la memoria del usuario y diálogos estéticos y diseño minimalista*. Por otra parte, es necesario fortalecer la categoría *prevención de errores* ya que tuvo la puntuación más baja en comparación con las demás categorías.

Con base en los resultados obtenidos en el presente trabajo, se relacionan a continuación algunas amenazas a la validez que no se consideraron y que pudieron disminuir la efectividad del eBook como herramienta de apoyo al proceso de enseñanza-aprendizaje:

- *Accesibilidad del material instruccional*: El eBook se mantuvo accesible durante el *post-test*, sin embargo, este elemento podría ser excluido para medir con más precisión la capacidad de comprensión y recordación de las temáticas.
- *Formación académica del público objetivo*: Considerando que el público objetivo estuvo conformado por estudiantes del curso de *Teoría de la información y sistemas de comunicación* de la Universidad Nacional de Colombia, ya conocían conceptos básicos de la comprensión de datos y podrían tener un contexto más claro para la comprensión y estudio de las temáticas que se pretendían enseñar.
- *Carencia de expertos*: El apoyo de un asesor pedagógico podría darle mayor calidad al diseño didáctico del eBook interactivo, y con el apoyo de un profesional del diseño gráfico se podría lograr un mayor impacto visual y ahorro de tiempo en el desarrollo de la interfaz gráfica de usuario y en el modelado y animación de los objetos 3D (Cedeño & Vargas, 2021)

6. Conclusiones y recomendaciones.

6.1 Conclusiones

En el estudio de temáticas abstractas como la compresión de datos, los estudiantes encuentran dificultades a la hora de comprender los conceptos y aplicaciones; sin embargo, los eBooks interactivos permiten nuevas prácticas de aprendizaje de mayor interacción (Alshaya & Oyaid, 2017) haciendo uso de las tecnologías de la información y la comunicación presentes en distintos dispositivos y mediante la integración de elementos multimedia (Cedeño & Vargas, 2021).

En la revisión sistemática de literatura efectuada, no se encontraron eBooks interactivos que apoyen el proceso de enseñanza-aprendizaje en el área de la compresión de datos. Por lo tanto, en el presente trabajo se realizó el análisis, diseño, desarrollo, implementación y evaluación de un eBook interactivo para apoyar el proceso de enseñanza-aprendizaje en dicha área.

El eBook interactivo desarrollado está compuesto por módulos focalizados en los siguientes temas: conceptos básicos de la compresión de datos, introducción a la teoría de la información, técnicas de compresión sin pérdidas (métodos estadísticos, compresión basada en contexto y métodos de diccionario) y técnicas de compresión con pérdida.

La herramienta de software que se eligió para el desarrollo fue adecuada: Unity3d permitió crear fácilmente escenas, animaciones e interacciones, generando una aplicación multiplataforma que todos los estudiantes (público objetivo) pudieron ejecutar en los diversos dispositivos y sistemas operativos con que contaban.

“Para la fase de implementación del eBook interactivo se realizaron las siguientes actividades: aplicación de una encuesta de contexto, identificación y mitigación de las amenazas a la validez que podrían afectar la investigación, diseño de una prueba de *pre/post test*, diseño de una prueba de usabilidad, y diseño y ejecución de una estrategia instruccional” (Cedeño & Vargas, 2021).

“La evaluación del eBook interactivo tuvo en cuenta los siguientes aspectos: evaluación de la efectividad de la acción formativa en el público objetivo y evaluación de la usabilidad del eBook interactivo por parte del público objetivo. Con respecto a la evaluación de la efectividad de la acción formativa en el público objetivo, las pruebas estadísticas indicaron que existe evidencia significativa para afirmar que fue efectiva la intervención pedagógica con el eBook interactivo. Al analizar las respuestas correctas por módulos temáticos en el *pre* y *post-test*, se encontró que los temas de *conceptos básicos de la comprensión de datos, introducción a la teoría de la información, técnicas de comprensión sin pérdidas - métodos estadísticos* y *técnicas de comprensión con pérdida*, tuvieron un incremento significativo en el número de las respuestas correctas; sin embargo, en el tema de *técnicas de comprensión sin pérdidas – métodos de diccionario*, se evidenció un decremento en el número de respuestas correctas de los estudiantes” (Cedeño & Vargas, 2021).

“Con respecto a la evaluación de la usabilidad del eBook interactivo por parte del público objetivo, la prueba correspondiente arroja un resultado positivo en las categorías de *uso del lenguaje de los usuarios, consistencias y estándares, minimizar la carga de la memoria del usuario, y diálogos estéticos y diseño minimalista*; no obstante, es necesario fortalecer la categoría *prevención de errores* ya que tuvo la puntuación más baja en comparación con las demás categorías” (Cedeño & Vargas, 2021).

Se identificaron las siguientes amenazas a la validez, que no fueron consideradas y que pudieron disminuir la efectividad del eBook interactivo para apoyar el proceso de enseñanza-aprendizaje: accesibilidad del material instruccional, formación académica del público objetivo y carencia de expertos. No obstante, el eBook interactivo provee a los educadores de un recurso educativo digital que apoya la comprensión de un tema abstracto como lo es la comprensión de datos, permitiendo la estimulación cognitiva y despertando la curiosidad de los estudiantes, lo que promueve un mayor aprendizaje de la información expuesta (Toca & Carrillo, 2019).

6.2 Recomendaciones

Respecto a las especificaciones para realizar la evaluación de la acción formativa, se recomienda restringir la accesibilidad al material instruccional (eBook interactivo) durante la prueba de *post-test*, para medir con más precisión la capacidad de comprensión y recordación de las temáticas.

En búsqueda de mejorar los aspectos pedagógicos e instruccionales del eBook interactivo, se recomienda incluir el apoyo de un profesional en pedagogía de forma que se generen observaciones y ajustes para maximizar la calidad en el diseño didáctico del eBook. Adicionalmente, dado el contexto multimedia del proyecto, se recomienda contar con apoyo de profesionales en diseño gráfico, con quienes se buscaría maximizar los contenidos audiovisuales, mejorando los tiempos en el desarrollo de los contenidos interactivos, recursos 3D y la interfaz de usuario.

Debido al alcance del presente trabajo, el eBook interactivo se diseñó como una herramienta didáctica para el apoyo efectivo en el aprendizaje de los estudiantes, razón por la cual se recomienda que en futuras implementaciones se incluya un diseño y funciones que permitan el trabajo y aprendizaje colaborativo.

Con base en el test de usabilidad aplicado a los estudiantes, se recomienda mejorar la *prevención de errores* con el fin de optimizar la experiencia del usuario.

ANEXOS

A. Anexo: Muestras de scripts de código desarrollados

A continuación, se encuentran muestras de los principales scripts de código desarrollados.

A-1. Scripts de Cámara y Navegabilidad

CamControl.cs

```
//Obtener elemento señalado:
```

```
Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
```

```
RaycastHit hit;
```

```
if (Physics.Raycast(ray, out hit, 100))
```

```
{
```

```
    Clic = hit.transform.gameObject.name;
```

```
    if (Clic == "FocusPage1" || Clic == "FocusPage2" || Clic == "FocusPC")
```

```
    {
```

```
        if (DoubleClic)
```

```
        {
```

```
            if (primerclic == true & timetoclic > 0)
```

```
            {
```

```
                primerclic = false;
```

```
                timetoclic = 0;
```

```
                DClic = Clic;
```

```
                Clic = null;
```

```
                ClicAction();
```

```
            }
```

```
        } else
```

```
        {
```

```
            primerclic = true;
```

```
            timetoclic = DeltaClic;
```

```
            Clic = null;
```

```
        }
```

```
    }
```

```
else
```

```
{
```

```
    primerclic = false;
```

```
    timetoclic = 0;
```

```
    DClic = Clic;
```

```
    Clic = null;
```

```
    ClicAction();
```

```
}
```

```
}
```

```
//Desplazamiento y rotación al sitio indicado
```

```
private bool IrHaciaHito(Vector3 PosDestino, Quaternion TargetRot)
```

```
{
```

```
    if (gameObject.GetComponent<Camera>().fieldOfView <= SetZoom - 4) {
```

```

    gameObject.GetComponent<Camera>().fieldOfView += Time.deltaTime * 60;
} else {
    gameObject.GetComponent<Camera>().fieldOfView = SetZoom;
}

if (ValStep <= 1)
{
    ValStep+= Time.deltaTime * Velmov / (PosDestino - PosInit).magnitude;

    transform.position = Vector3.Lerp(PosInit, PosDestino, ValStep);
    transform.rotation = Quaternion.Lerp(InitRot, TargetRot, ValStep);
}
else
{
    ValStep = 0;
    return true;
}
return false;
}

```

A-2. Scripts de interacciones

```

FuntRetry.cs
void OnEnable()
{
    GOClone = GameObject.FindGameObjectWithTag("GORetry");
    if (!GOClone)
    {
        GOClone = Instantiate(GORetry, GORetry.transform.parent);
        GOClone.tag = "GORetry";
    }
    GORetry.SetActive(false);
}

void OnDisable()
{
    if(ResetOnDisable)
        Destroy(GOClone);
    GORetry.SetActive(true);
}

private void OnMouseUp()
{
    Destroy(GOClone);
    GORetry.SetActive(true);
    GOClone = Instantiate(GORetry, GORetry.transform.parent);
    GOClone.tag = "GORetry";
    GORetry.SetActive(false);
}

```

```

//Funciones para arrastrar un Objeto - GrabObject.cs
void OnMouseDown()
{
    if (Time.timeScale <= 1)
        return;

    screenPoint = Camera.main.WorldToScreenPoint(gameObject.transform.position);
    offset = gameObject.transform.position - Camera.main.ScreenToWorldPoint(new
Vector3(Input.mousePosition.x, Input.mousePosition.y, screenPoint.z));
    if (!PrimerClic) {
        if (UseLocalPos)
            posInit = transform.localPosition;
        else
            posInit = transform.position;
        PrimerClic = true;
    }
    if (Clone)
GOClone = Instantiate (gameObject, gameObject.transform.position, gameObject.transform.rotation,
gameObject.transform.parent);

    if (Particulas)
Particulas.SetActive (true);

    if (ScaletoGrab > 1)
        transform.localScale = transform.localScale * ScaletoGrab;
}

void OnMouseDownDrag()
{
    if (Time.timeScale <= 1)
        return;
    Vector3 curScreenPoint = new Vector3(Input.mousePosition.x, Input.mousePosition.y, screenPoint.z);
    transform.position = Camera.main.ScreenToWorldPoint(curScreenPoint)+offset;
}

```

A-3. Scripts de estado, configuración y almacenamiento de perfiles

```

// Función Guardado
public void SaveToken()
{
    Pagina = ScriptBook.page;
    string Savegamepath= Application.persistentDataPath +"/Savegame_"+Token+".dat";
    BinaryFormatter bf =new BinaryFormatter();
    FileStream file=File.Create(Savegamepath);
    datatosave data = new datatosave(Usuario,Pagina,LuzValue,VolumeValue, MusicVol, CamZoom, Cal);
//Valores a Guardar
    bf.Serialize (file, data);
    file.Close ();
}

```

```
}

```

```
// Función Cargar

```

```
public bool LoadToken()

```

```
{

```

```
    string Loadgamepath= Application.persistentDataPath +"/Savegame_"+Token+".dat";

```

```
    if (File.Exists (Loadgamepath)) {

```

```
        BinaryFormatter bf = new BinaryFormatter ();

```

```
        FileStream file = File.Open (Loadgamepath, FileMode.Open);

```

```
        datatosave data = (datatosave)bf.Deserialize (file);

```

```
        Usuario=data.Usuario;

```

```
        LuzValue = data.LuzValue;

```

```
        VolumeValue=data.VolumeValue;

```

```
        MusicVol = data.MusicVol;

```

```
    CamZoom = data.CamZoom;

```

```
        Cal=data.Cal;

```

```
    if (LoadAllValues)

```

```
    {

```

```
        Pagina = data.Pagina;

```

```
    }

```

```
    bf.Serialize (file, data);

```

```
        file.Close ();

```

```
        return true;

```

```
    }

```

```
else {

```

```
    MsgToken.text = "Token Invalido, No coresponde a una partida valida. Ingresa Nuevamente ";

```

```
    MsgToken.color = Color.red;

```

```
    return false;

```

```
    }

```

```
}

```

```
//Actualizar información en Canvas

```

```
public void UpdateScreen()

```

```
{

```

```
    TxtName.text = Usuario;

```

```
    score = 0;

```

```
if (Cal != null) {
    foreach (float Punt in Cal) {
        if (Punt >= 0) {
            score += Punt;
        }
    }
} else {
    score = 0;
}
TxtScore.text = "Score: "+score+ " Puntos";
Luz.intensity = LuzValue;
GameObject.Find ("SliderVol").GetComponent<Slider> ().value=VolumeValue;
GameObject.Find("SliderVol").SendMessage("UpdateVolMusic", MusicVol);
FuntCam.SendMessage("ActualizarZoom", CamZoom);
ScriptBook.page = Pagina;
}
```

B. Anexo: Prueba de *pre/post-test*

A continuación, se encuentra el cuestionario correspondiente a la prueba de *pre/post-test* aplicada a los estudiantes y su correspondiente solución.

B-1. Cuestionario del *pre/post-test*

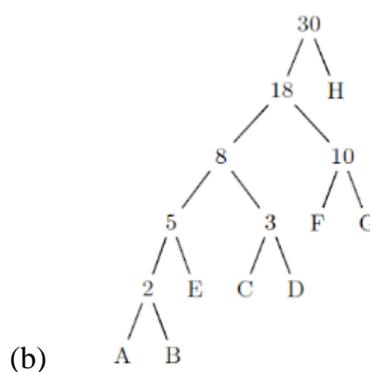
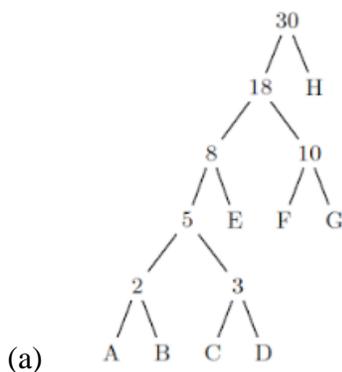
1. Para una DMS X que produce los símbolos $\{s, a, i, m, n\}$ cuyas probabilidades correspondientes son $\{4/11, 3/11, 2/11, 1/11, 1/11\}$, calcule e identifique cual es el símbolo con mayor auto información (Howlett, 2003).

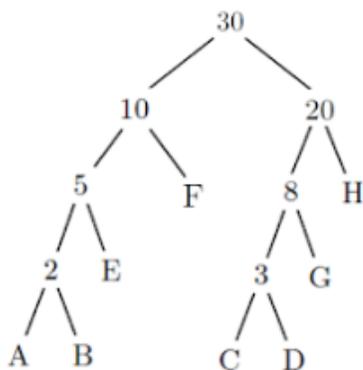
- (a) El símbolo con mayor auto información es "n" con $I=3.45$ bits.
- (b) El símbolo con mayor auto información es "s" con $I=3.45$ bits.
- (c) El símbolo con mayor auto información es "a" con $I=4$ bits.
- (d) El símbolo con mayor auto información es "s" con $I=4$ bits.

2. ¿Cuál es la información promedio de la fuente identificada en el ejercicio anterior?

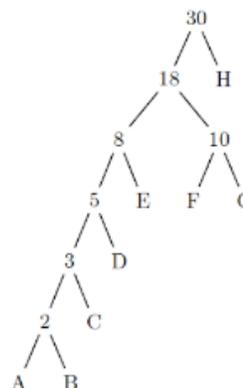
- (a) 4 bits
- (b) 1.8 bits
- (c) 2 bits
- (d) 2.12 bits

3. Dados los ocho símbolos A, B, C, D, E, F, G y H con probabilidades respectivas de $1/30, 1/30, 1/30, 2/30, 3/30, 5/30, 5/30$ y $12/30$. Indique cuál de los siguientes árboles de código genera un código Huffman válido:





(c)



(d)

4. ¿Cuál es la entropía máxima posible H de una fuente X que consiste en N letras diferentes?

- (a) Entropía Máxima $H(X) = -N \cdot \log(1/N)$
- (b) Entropía Máxima $H(X) = (1/N) \cdot \log(1/N)$
- (c) Entropía Máxima $H(X) = -(1/N) \cdot \log(1/N)$
- (d) Entropía Máxima $H(X) = -\log(1/N)$

5. En tal fuente cuya entropía es máxima, ¿cuál es la probabilidad de su letra más probable? ¿Cuál es la probabilidad de su letra menos probable?

- (a) Probabilidad Max=N y Probabilidad Min = 0.
- (b) Probabilidad Max=N y Probabilidad Min = 1/N.
- (c) Probabilidad Max=1/N y Probabilidad Min = 1/N.
- (d) Probabilidad Max=1/N y Probabilidad Min = 0.

6. Para la fuente $X = \{A, B, C, D, E\}$, se tienen los siguientes códigos, identifique cuál NO es un código exclusivamente decodificable.

X	<i>Código</i>
A	000
B	001
C	010
D	011
E	100

(a)

X	<i>Código</i>
A	0
B	01
C	011
D	0111
E	01111

(b)

X	Código
A	1110
B	0
C	1111
D	10
E	110

(c)

X	Código
A	1
B	01
C	010
D	0111
E	0000

(d)

7. ¿Cuál de los siguientes conjuntos de palabras de código podría ser el código de Huffman para un alfabeto fuente de 4 símbolos? (Howlett, 2003)

- (a) 01, 10, 00, 111
- (b) 0, 10, 110, 111
- (c) 1, 01, 10, 001
- (d) 0, 110, 111, 101.

8. Obtenga la codificación Huffman para la DMS X cuyos símbolos y probabilidades se relacionan en la tabla (Sayood, 2006):

Carácter	Probabilidad
a	2/17
b	3/17
c	3/17
d	3/17
e	3/17
f	2/17
g	1/17

- (a) a: 0110; b: 001; c: 010; d: 0011; e: 11; f: 01; g: 001.
- (b) a: 110; b: 011; c: 010; d: 111; e: 000; f: 001; g: 101.
- (c) a: 0000; b: 010; c: 10; d: 011; e: 11; f: 001; g: 0001.
- (d) a: 000; b: 0010; c: 010; d: 0011; e: 11; f: 001; g: 01.

9. Para el código generado en la pregunta anterior, justifique si es o no óptimo:

- (a) NO es óptimo ya que $H=2,82$ bits y $L=3$ bits por símbolo.
- (b) Es óptimo ya que $H=2,32$ bits y $L=2,46$ bits por símbolo.
- (c) Es óptimo ya que $H=2,82$ bits y $L=3$ bits por símbolo.
- (d) Es óptimo ya que $H=2,73$ bits y $L=2,82$ bits por símbolo.

13. Identifique cuál de los siguientes contextos es inválido para una codificación PPM

(a)

<i>Contexto</i>	<i>f</i>	<i>p</i>
<i>as</i> → <i>n</i>	1	1/4
<i>as</i> → <i>s</i>	2	2/4
<i>esc</i>	1	1/4

(b)

<i>Contexto</i>	<i>f</i>	<i>p</i>
<i>as</i> → <i>s</i>	1	1/2
<i>esc</i>	1	1/2

(c)

<i>Contexto</i>	<i>f</i>	<i>p</i>
<i>ss</i> → <i>a</i>	1	1/4
<i>as</i> → <i>s</i>	1	1/4
<i>esc</i>	2	2/4

(d)

<i>Contexto</i>	<i>f</i>	<i>p</i>
<i>a</i> → <i>s</i>	1	1/2
<i>esc</i>	1	1/2

14. Considere la DMS X cuyos símbolos corresponden a la lengua española. Determine la codificación MTF para la frase "pepepeca".

(a) 15 5 1 2 1 9 5 3

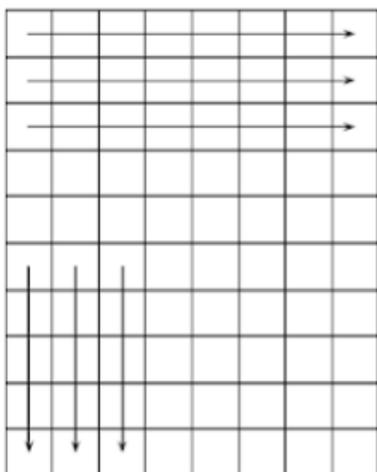
(b) 4 15 0 2 1 9 5 3

(c) 15 1 1 5 1 1 4 3

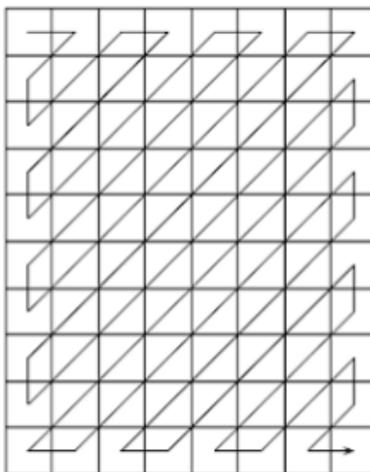
(d) 15 5 1 1 1 1 4 3

15. Indique cuál es el patrón de muestreo o lectura que se emplea en la compresión JPEG con pérdidas (Sayood, 2006).

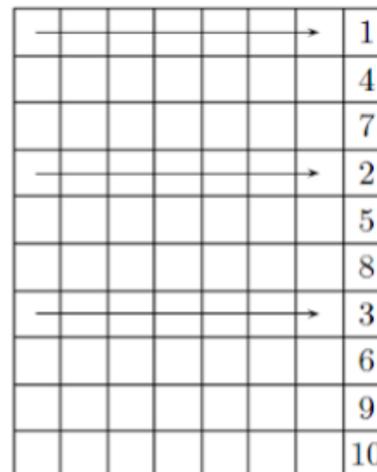
(a)



(b)



(c)



B-2. Solución cuestionario *pre/post-test*

1 La auto información es directamente proporcional a la incertidumbre de que un evento ocurra, por lo que los símbolos {m,n} representan los símbolos con mayor auto información. Adicionalmente se calcula la información propia para el símbolo n con lo que se obtiene:

$$I(x_i) = -\log_b(P(x_i)) \qquad I(n) = -\log_2\left(\frac{1}{11}\right) = 3.45$$

Respuesta correcta:

(a) El símbolo con mayor auto información es "n" con $I=3.45$ bits.

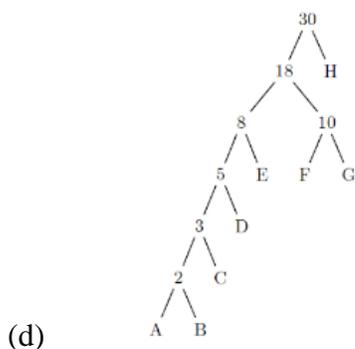
2. Se calcula la entropía para los símbolos con las probabilidades expuestas:

$$\begin{aligned} H(X) &= -\frac{4}{11}\log_2\left(\frac{4}{11}\right) - \frac{3}{11}\log_2\left(\frac{3}{11}\right) - \frac{2}{11}\log_2\left(\frac{2}{11}\right) - \frac{1}{11}\log_2\left(\frac{1}{11}\right) - \frac{1}{11}\log_2\left(\frac{1}{11}\right) \\ &= 2.12 \text{ b/símbolo} \end{aligned}$$

Respuesta correcta:

(d) 2.12 b/símbolo

3 Se realiza la generación del árbol de código Huffman con base en el algoritmo descrito con lo que se obtiene la respuesta correcta:



4. La entropía de una fuente X es máxima cuando sus N letras diferentes tienen la misma probabilidad de ocurrencia, con lo que se reduce su expresión a $H(X) = -\text{Log}(1/N)$.

Respuesta correcta:

(d) Entropía Máxima $H(X) = -\text{Log}(1/N)$

5. La entropía de una fuente X es máxima cuando sus N letras diferentes tienen la misma probabilidad de ocurrencia $p(x)=1/N$.

Respuesta correcta:

(c) Probabilidad Max= $1/N$ y Probabilidad Min = $1/N$.

6 el código de la opción (d) no es exclusivamente decodificable debido a que la palabra de código del símbolo B es prefijo de las palabras de código para C y D, con lo que se generan ambigüedades en la decodificación.

Respuesta correcta:

X	Código
A	1
B	01
C	010
D	0111
E	0000

(d)

7. Un código Huffman se caracteriza por que es generado por un árbol cerrado en el cual todas sus ramas se encuentran completas, esto implica que siempre debe contener 02 símbolos con la mayor longitud de código lo cual permite descartar las opciones de respuesta según se indica:

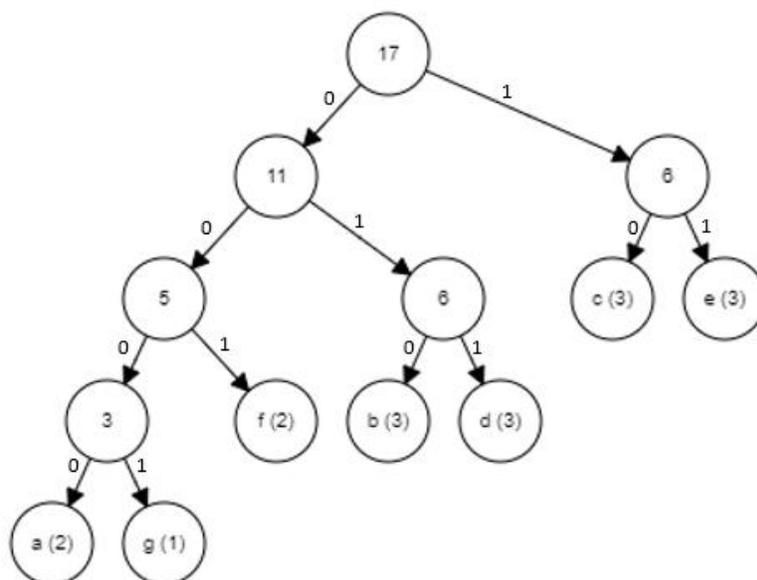
(a) 01, 10, 00, 111 y (c) 1, 01, 10, 001 Existe solo un elemento con L máxima lo que implica la existencia de al menos una rama incompleta.

(d) 0, 110, 111, 101. No es posible la generación de un árbol cerrado con estas palabras de código.

Respuesta correcta:

(b) 0, 10, 110, 111

8. Se genera el árbol de codificación Huffman para la DMS X dada:



Generado de <https://cmps-people.ok.ubc.ca/ylocet/DS/Huffman.html>

Respuesta correcta:

(c) a: 0000; b: 010; c: 10; d: 011; e: 11; f: 001; g: 0001.

9. Se calcula la entropía y la longitud promedio para evaluar la expresión $L \leq H(X) + 1$

$$L = 2.82 \quad H(X) = 2.73 \quad \rightarrow \quad 2.82 \leq 2.73 + 1$$

Respuesta correcta:

(d) Es óptimo ya que $H=2,73$ bits y $L=2,82$ bits por símbolo.

10. Se procede con la subdivisión de los intervalos con base en la distribución de probabilidad acumulada FDA:

<i>Carácter</i>	<i>Probabilidad</i>	<i>FDA</i>
M	0,125	0,125
C	0,375	0,5
O	0,5	1

	<i>Símbolo</i>	<i>C</i>	<i>O</i>	<i>M</i>	<i>O</i>	<i>C</i>	<i>O</i>	<i>C</i>	<i>O</i>
Low	0	0,125	0,3125	0,3125	0,3242	0,3257	0,3279	0,3282	0,3286
High	1	0,5	0,5	0,3359	0,3359	0,3301	0,3301	0,3290	0,3290

Respuesta correcta:

(c) Tag = 0,3289

11. Se calcula la decodificación del Tag indicado:

<i>newtag</i>	<i>Rango</i>	<i>Símbolo</i>
$(0,687-0)/(1-0)= 0,687$	[0,5 - 1)	O
$(0,687-0,5)/(1-0,5)= 0,374$	[0,125 - 0,5)	C
$(0,374-0,125)/(0,5-0,125)= 0,664$	[0,5 - 1)	O
$(0,664-0,5)/(1-0,5)= 0,328$	[0,125 - 0,5)	C
$(0,328-0,125)/(0,5-0,125)= 0,5413$	[0,5 - 1)	O
$(0,5413-0,5)/(1-0,5)= 0,0826$	[0 - 0,125)	M

Respuesta correcta:

(c) OCOCOM

12. se ejecuta la decodificación LZW de la secuencia recibida según se muestra:

RESUMEN DE LA EJECUCIÓN DEL ALGORITMO					
<i>S</i>	<i>Ind.</i>	<i>T</i>	Salida	<i>S</i> ₁	Adicionar al diccionario (<i>S</i> · <i>S</i> ₁)
""	4	<i>p</i>	<i>p</i>	<i>p</i>	
<i>p</i>	1	<i>e</i>	<i>e</i>	<i>e</i>	11 → <i>pe</i>
<i>e</i>	5	<i>r</i>	<i>r</i>	<i>r</i>	12 → <i>er</i>
<i>r</i>	5	<i>r</i>	<i>r</i>	<i>r</i>	13 → <i>rr</i>
<i>r</i>	6	<i>o</i>	<i>o</i>	<i>o</i>	14 → <i>ro</i>
<i>o</i>	3	–	–	–	15 → <i>o</i> –
–	14	<i>ro</i>	<i>ro</i>	<i>r</i>	16 → <i>–r</i>
<i>ro</i>	11	<i>pe</i>	<i>pe</i>	<i>p</i>	17 → <i>rop</i>
<i>pe</i>	14	<i>ro</i>	<i>ro</i>	<i>r</i>	18 → <i>per</i>
<i>ro</i>	16	<i>–r</i>	<i>–r</i>	–	19 → <i>ro</i> –
<i>–r</i>	7	<i>a</i>	<i>a</i>	<i>a</i>	20 → <i>–ra</i>
<i>a</i>	4	<i>p</i>	<i>p</i>	<i>p</i>	21 → <i>ap</i>
<i>p</i>	7	<i>a</i>	<i>a</i>	<i>a</i>	22 → <i>pa</i>
<i>a</i>	3	–	–	–	23 → <i>a</i> –
–	2	<i>l</i>	<i>l</i>	<i>l</i>	24 → <i>–l</i>
<i>l</i>	7	<i>a</i>	<i>a</i>	<i>a</i>	25 → <i>la</i>
<i>a</i>	3	–	–	–	26 → <i>a</i> –
–	22	<i>pa</i>	<i>pa</i>	<i>p</i>	27 → <i>–p</i>
<i>pa</i>	22	<i>pa</i>	<i>pa</i>	<i>p</i>	28 → <i>pap</i>

Respuesta correcta:

(d) PAP

13. En la codificación PPM, la frecuencia para el carácter “esc” corresponde al número de predicciones para cada contexto, cuya condición no se cumple en el contexto que se evidencia en la opción (a)

Respuesta correcta:

(a)

<i>Contexto</i>	<i>f</i>	<i>p</i>
<i>as</i> → <i>n</i>	1	1/4
<i>as</i> → <i>s</i>	2	2/4
<i>esc</i>	1	1/4

14. Se calcula la codificación MTF de la palabra "pepepeca".

Iteración	Lista símbolos indexados																									Salida	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		25
p	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	15
e	p	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	5
p	e	p	a	b	c	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	1
e	p	e	a	b	c	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	1
p	e	p	a	b	c	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	1
e	p	e	a	b	c	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	1
c	e	p	a	b	c	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	4
a	c	e	p	a	b	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	3
	a	c	e	p	b	d	f	g	h	i	j	k	l	m	n	o	q	r	s	t	u	v	w	x	y	z	

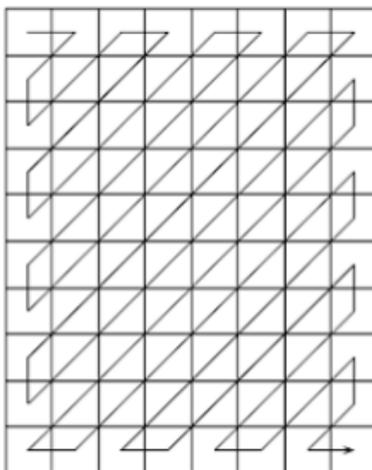
Respuesta correcta:

(d) 15 5 1 1 1 1 4 3

15. El patrón de muestreo o lectura que se emplea en la compresión JPEG con pérdidas corresponde al zigzag, según la opción (b)

Respuesta correcta:

(b)



C. Anexo: Test e informe de usabilidad

C-1. Test de usabilidad

El test de usabilidad tiene como finalidad conocer la experiencia con el uso del eBook interactivo “Compresión de datos”, representado en un entorno 3D. Las preguntas están clasificadas en 8 categorías según se observa en la columna “Categoría” de la siguiente tabla. La información que se recolecte será de uso académico y no se divulgará ningún tipo de datos personales.

Por favor marque la opción con la que más se identifique en una escala de 1 a 5, en donde 1 es “Totalmente en desacuerdo” y 5 es “Totalmente de acuerdo”.

Categoría	No.	Pregunta
<i>Visibilidad del estado del sistema.</i>	1	¿En la ejecución del eBook fue sencillo comprender la funcionalidad del Libro de Estudio?
	2	¿Fue sencillo comprender la funcionalidad del PC interactivo?
	3	¿Fue sencillo comprender la funcionalidad e identificar la perspectiva de inicio en la ejecución del eBook?
	4	¿Considera cómodos y adecuados los desplazamientos entre el Libro de estudio, el PC interactivo y la perspectiva de inicio?
	5	¿Es adecuada e intuitiva la forma de acceder a cada uno de los elementos principales del eBook como lo son el Libro de estudio, el PC interactivo y la perspectiva de inicio?
	6	¿Puede identificar fácilmente, si se está ejecutando el eBook en modo de lectura o modo de estudio?
	7	¿La información mostrada acerca de las calificaciones obtenidas es consistente?
	8	¿El Dashboard del pc interactivo es intuitivo y funciona adecuadamente?
	9	¿Los términos utilizados en cada una de las opciones del menú principal son coherentes y apropiados?

<i>Utilizar el lenguaje de los usuarios.</i>	10	¿Los términos utilizados en las opciones del menú de pausa son coherentes y apropiados?
	11	¿Las instrucciones contenidas en el eBook para la ejecución de las interactividades son pertinentes?
	12	¿La terminología utilizada para explicar la temática del eBook interactivo es adecuada para usted?
<i>Control y libertad para el usuario.</i>	13	¿Fue sencillo e intuitivo restablecer o reiniciar las actividades de evaluación disponibles en el eBook interactivo?
	14	¿Considera adecuado el uso de la tabla de contenido interactiva, para ir a una temática deseada dentro del eBook?
	15	¿Fue sencillo y funcional ir a una página establecida utilizando la opción “ir a página” dentro del eBook Interactivo?
	16	¿Fue fácil e intuitivo salir del modo de estudio, modo de lectura y del aplicativo del eBook interactivo?
<i>Consistencia y estándares.</i>	17	¿El cambio entre el modo de estudio y el modo de lectura se puede realizar de manera fácil e intuitiva?
	18	¿La estructura que presenta la información académica del eBook interactivo es consistente?
	19	¿Las evaluaciones interactivas conservan relación directa al contenido educativo alojado en el libro de estudio?
	20	¿La tabla de contenido funciona de manera coherente y adecuada?
	21	¿Los cuadros de dialogo mostrados por el eBook interactivo acerca de los estados informados o las acciones requeridas presentan mensajes consistentes e intuitivos?
<i>Prevención de errores.</i>	22	¿Considera que el eBook interactivo funciona sin errores?
	23	¿Los mensajes mostrados en las distintas instancias del eBook Interactivo le fueron de utilidad?
	24	¿La validación de los datos ingresados tales como Nombres, Token o Números de página, es realizada de manera precisa e inmediata?

	25	¿Es consistente la información acerca de errores de formato y/o datos inválidos?
Minimizar la carga de la memoria del usuario.	26	¿La información mostrada por el eBook interactivo, tal como progreso del usuario, página y calificaciones, es correcta y pertinente?
	27	¿Las transiciones entre el libro de estudio y el PC interactivo mantienen el flujo de la lectura, retornando en las páginas o actividades pertinentes y conservando el progreso actual?
	28	¿La función de cargar partida le permite continuar con el progreso de estudio desde su ultimo uso?
Flexibilidad y eficiencia de uso.	29	¿El botón deslizable de volumen es pertinente y funcional?
	30	¿El menú “opciones” es de fácil acceso?
	31	¿Son funcionales las opciones de personalización del sistema tales como sonidos, iluminación y profundidad de visión?
	32	Los sonidos de las interacciones son adecuados y pertinentes?
	33	Considera agradable y pertinente la función de música de fondo?
	34	¿Las flechas de desplazamiento del libro de estudio son adecuadas y funcionan correctamente?
	35	¿El menú principal presenta opciones intuitivas y funciona adecuadamente?
	36	¿El dashboard del PC interactivo contiene información pertinente y relevante?
	37	¿El botón Home funciona adecuadamente?
	38	¿El botón Salvar funciona adecuadamente?
	39	La funcionalidad de la fichas interactivas es adecuada y pertinente?
Diálogos estéticos y diseño minimalista.	40	El panel que muestra las calificaciones obtenidas en las actividades interactivas es apropiado y funciona adecuadamente?
	41	El Zoom In/Out funciona adecuadamente?
	42	¿La información mostrada acerca de instrucciones o acciones requeridas, en los diferentes menús y elementos interactivos, es concisa y directa?
	43	¿Los botones e información de la interfaz de usuario son los necesarios y adecuados?

-
- | | |
|----|--|
| 44 | ¿Los mensajes de información que se muestran indican específicamente su significancia y acciones requeridas? |
| 45 | ¿Considera que las temáticas son explicadas de manera suficientemente concisa? |
| 46 | ¿Considera pertinente disminuir la cantidad de texto a cambio de audios que expliquen el contenido? |
-

C-2. Informe de usabilidad

El presente informe tiene como fin describir los hallazgos del análisis a los resultados de la prueba de usabilidad.

Propósito de la prueba de usabilidad

Esta prueba tuvo como propósito identificar y determinar tanto las fortalezas como las debilidades en la usabilidad del eBook interactivo, la identificación de las debilidades permitió realizar los ajustes pertinentes y mejorar las funcionalidades del eBook interactivo.

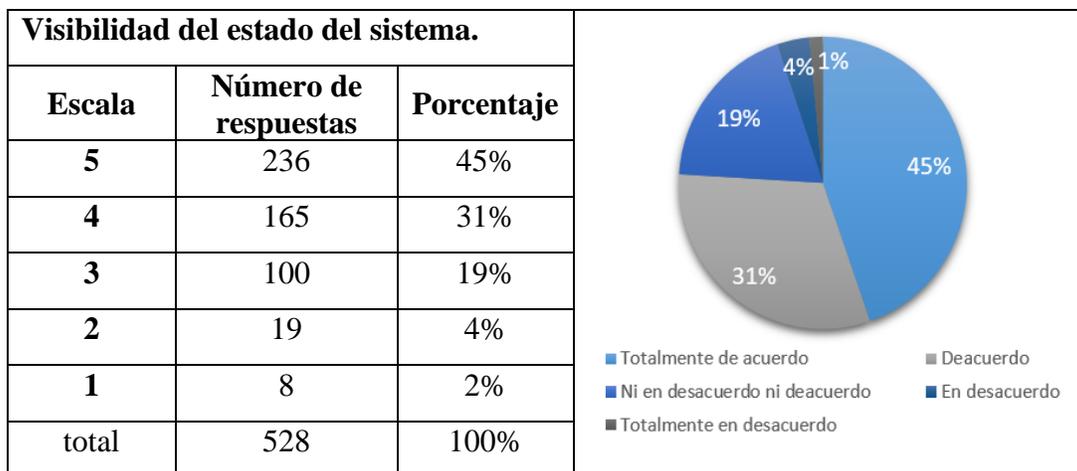
Esta prueba se basó en una puntuación de escala tipo Likert de 1 a 5, en donde 1 es “Totalmente en desacuerdo” y 5 es “Totalmente de acuerdo”.

A continuación, se hace un análisis del número de respuestas en cada opción de la escala para cada categoría de la prueba de usabilidad, considerando el total de 66 estudiantes de la Universidad Nacional de Colombia a quienes les fue aplicada la prueba.

Resultados por categorías

- *Visibilidad del estado del sistema*

En cuanto a la *Visibilidad del estado del sistema*, el 76% de las respuestas de los estudiantes se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

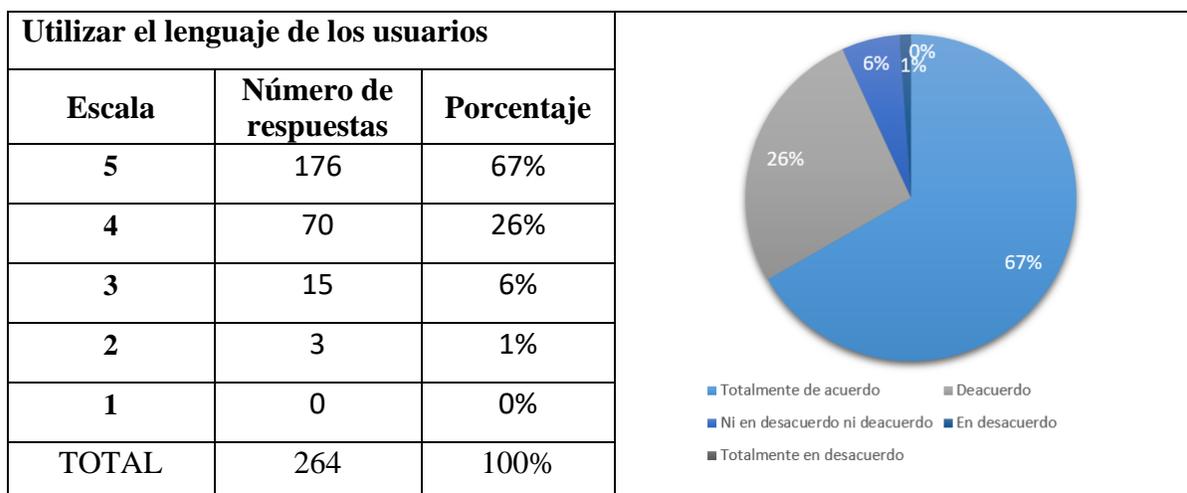


Hallazgos

En esta categoría con la versión de ordenador no se reportaron fallas, sin embargo, hay algunas sugerencias para la versión de Android como las correcciones de errores ortográficos, e inconformidades con la lentitud en el cambio de página.

- Utilizar el lenguaje de los usuarios

En la categoría *Utilizar el lenguaje de los usuarios*, El 93% de las respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

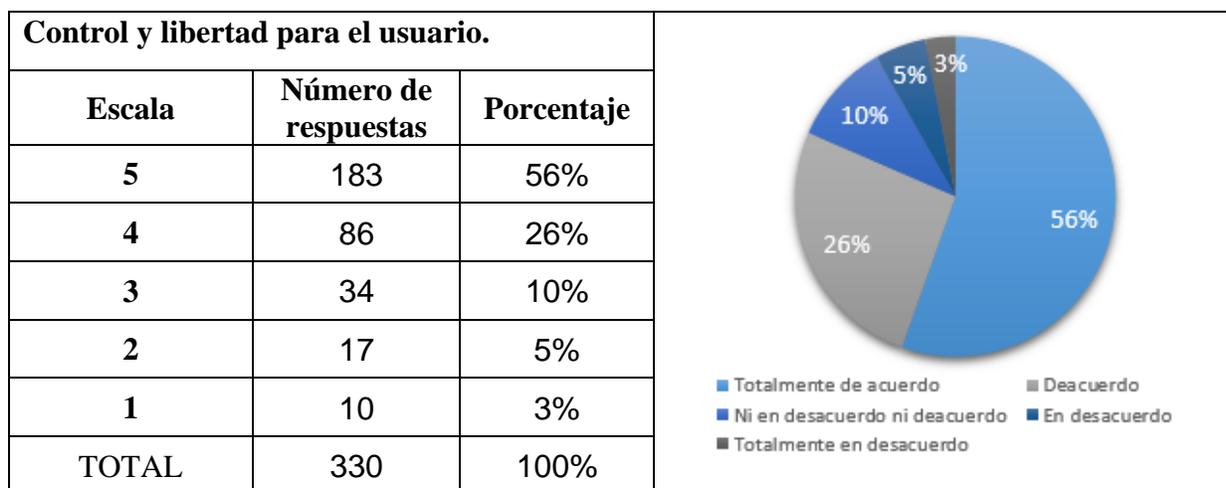


Hallazgos

Para esta categoría se resaltó el uso de lenguaje claro, coherente y de sencilla interacción, sin embargo, se encontraron sugerencias referidas a la inactividad de algunos elementos y la existencia de ejemplos y ejercicios en donde los valores se aproximan, pero en otros no, además se hallaron algunos errores de digitación.

- *Control y libertad para el usuario*

En la categoría *Control y libertad para el usuario*, el 82% de las respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

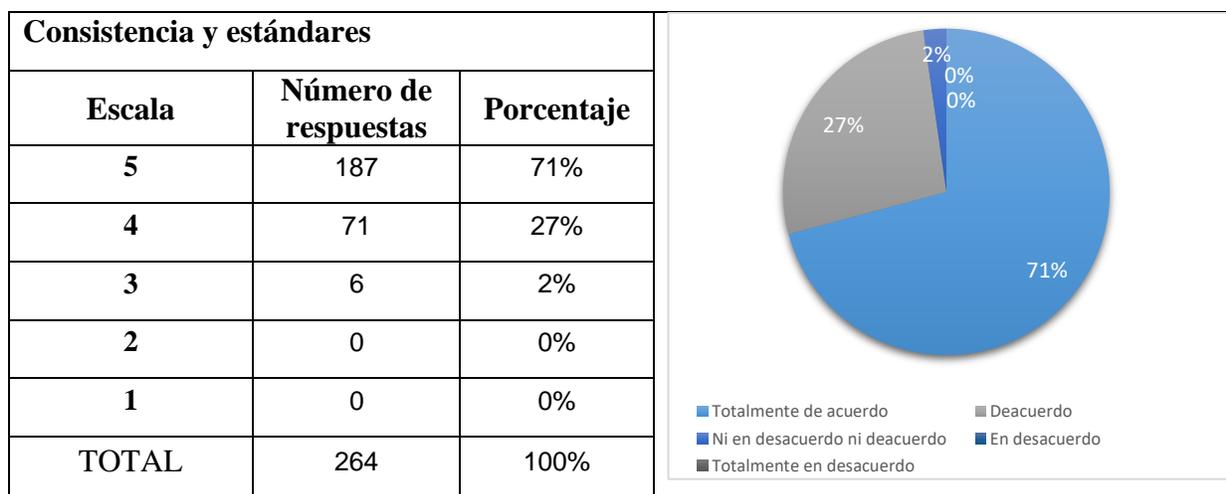


Hallazgos

Se encontró que los temas fueron explicados de manera amena y consistente, además se resalta la forma intuitiva de interactuar con elementos del eBook, en contraste surgieron algunas observaciones como por ejemplo acceder al libro desde la tabla de contenido.

- *Consistencia y estándares*

En la categoría *Consistencia y estándares*, el 98% de respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

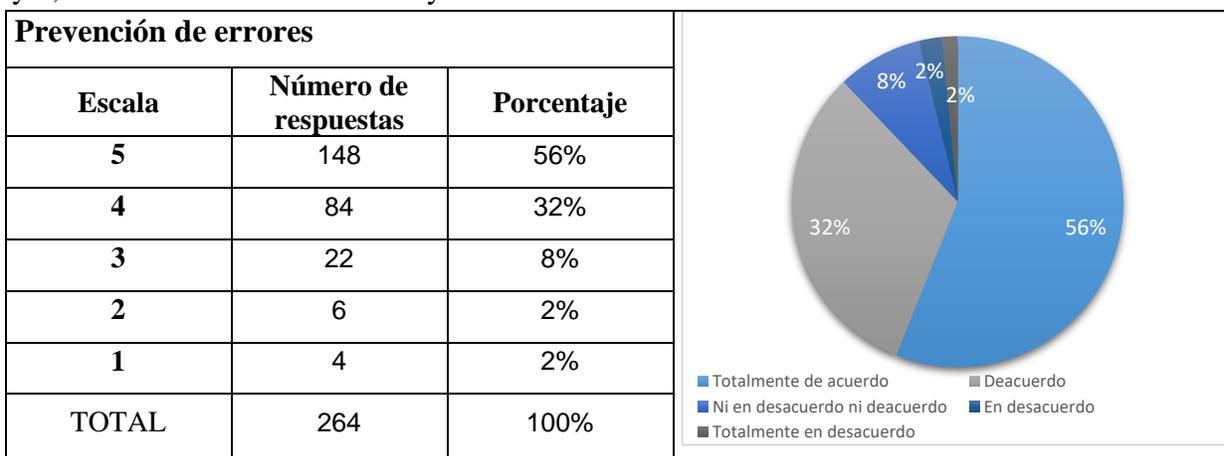


Hallazgos

Se encontraron algunas fortalezas como el adecuado contenido del eBook y apoyo de las ayudas y evaluaciones; en este apartado no surgieron hallazgos adicionales a los mencionados.

- Prevención de errores

En cuanto a la categoría *Prevención de errores*, el 88% de respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

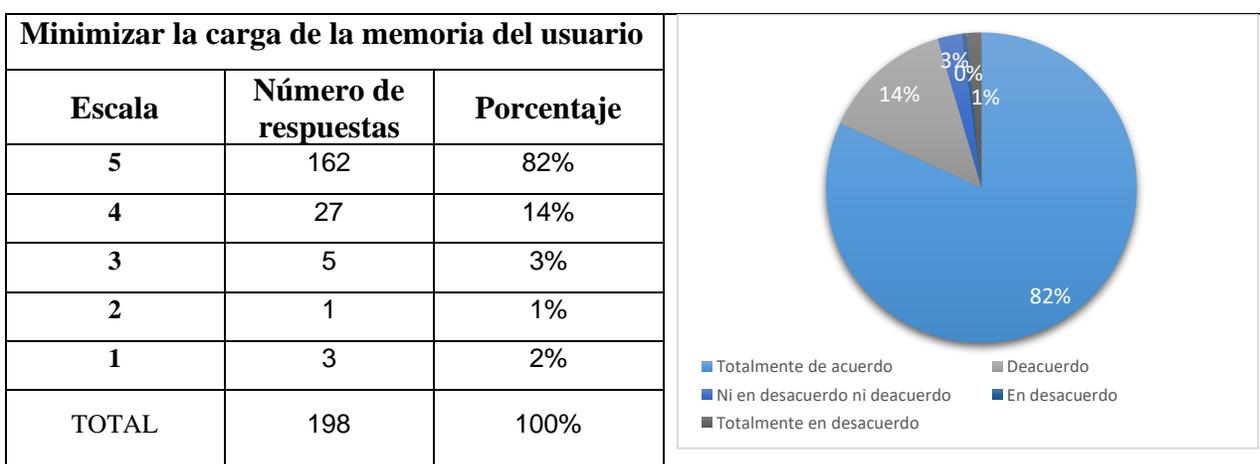


Hallazgos

En esta categoría se encontraron algunos errores al cambiar de página, al guardar información del progreso, algunos errores de digitación, mostrando la necesidad del acceso al teclado físico.

- Minimizar la carga de la memoria del usuario

En la categoría *Minimizar la carga de la memoria del usuario*, el 96% de respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

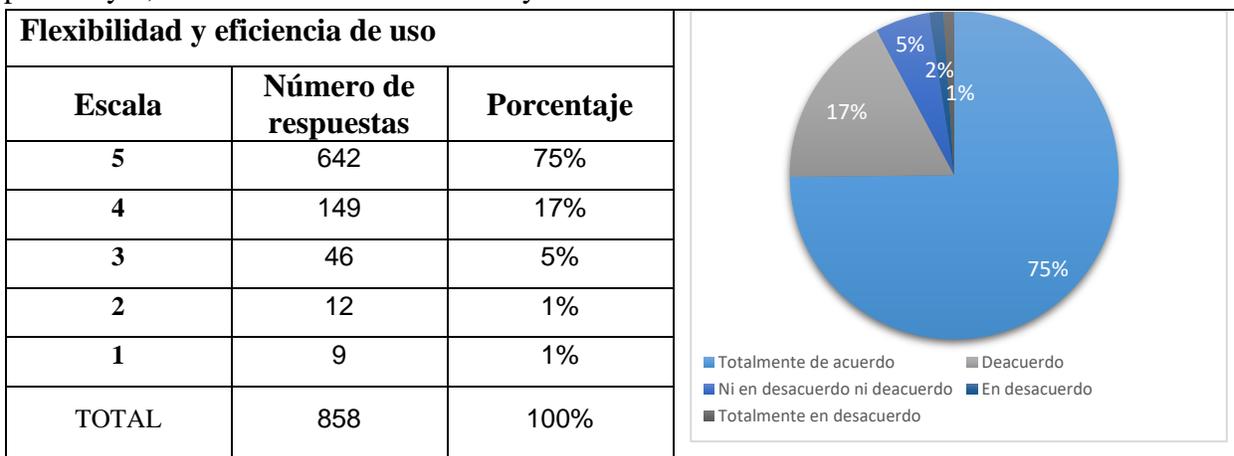


Hallazgos

Se resalta el adecuado funcionamiento del eBook, sin embargo, existieron sugerencias como el guardar la sesión con un usuario y la recomendación de guardar ejercicios mediante “checkpoints”.

- Flexibilidad y eficiencia de uso

En la categoría *Flexibilidad y eficiencia de uso*, el 92% de las respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.

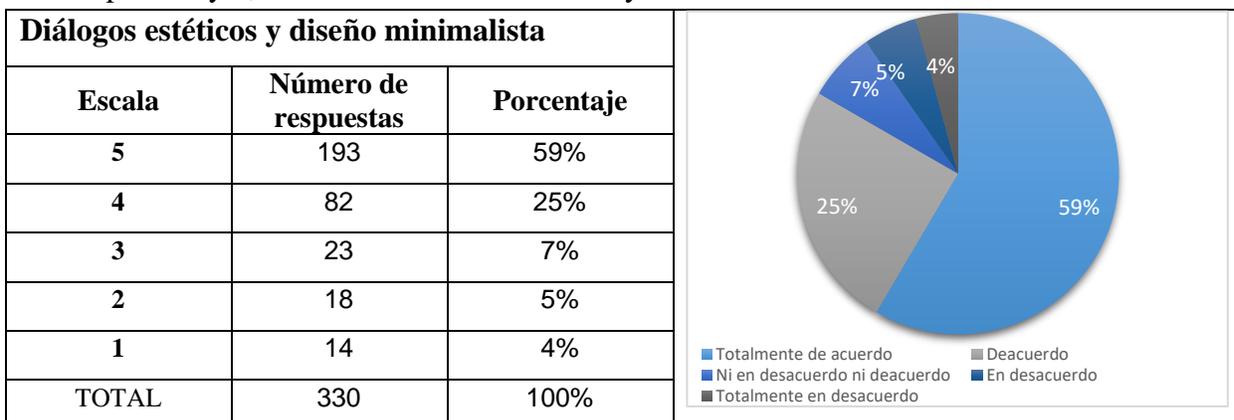


Hallazgos

En esta apartado se resalta la utilidad del eBook como alternativa de aprendizaje, pero también se encontraron algunas disparidades entre opiniones como el uso o no de los videos y de la música de fondo, así como la necesidad del uso del zoom.

- Diálogos estéticos y diseño minimalista

En la categoría *Diálogos estéticos y diseño minimalista*, el 84% de las respuestas se encuentran entre la opción 4 y 5, es decir entre “De acuerdo” y “Totalmente de acuerdo”.



Hallazgos

Se resalta la cantidad de imágenes y herramientas que posee el eBook, así como la suficiencia en cuanto al contenido para un adecuado aprendizaje.

Conclusiones generales de la prueba de usabilidad

La actividad cumplió con su objetivo de identificación de las debilidades y fortalezas en la usabilidad del eBook por parte de los usuarios, quienes participaron de manera activa y propositiva en las respuestas a las preguntas abiertas para realizar sugerencias sobre el eBook.

La puntuación Likert de todas las categorías se ubicó mayormente en las opciones 4 y 5, lo que refleja un funcionamiento adecuado del eBook interactivo; asimismo las respuestas en cada categoría permitieron reconocer y corregir errores puntuales del sistema; además se recibieron las recomendaciones hechas por los usuarios que facilitaron la optimización y actualización del eBook interactivo.

D. Anexo: Solución a los ejercicios del eBook Interactivo

A continuación, se muestra la solución a los ejercicios planteados en el eBook interactivo:

Actividad 1. Medidas de rendimiento

a.

$$FC = \frac{131072}{16384} = 8$$

b.

$$RC = \frac{1}{8} = 0.125$$

c.

$$\text{Reducción Compresión} = \left(1 - \frac{1}{8}\right) * 100 = 87.5\%$$

Actividad 2. Teoría de la información

1. Dados los 27 caracteres equiprobables del alfabeto,

$$H(X) = -27 \left(\frac{1}{27} \log_2 \left(\frac{1}{27} \right) \right)$$

$$H(X) = 4.75 \text{ b/símbolo}$$

2. Para el mensaje M_1 :

$$P(x_1x_2x_1x_3) = (0.4)(0.3)(0.4)(0.2) = 0.0096$$

$$I(x_1x_2x_1x_3) = -\log_2 0.0096 = 6.7 \text{ b/símbolo}$$

Para el mensaje M_2 :

$$P(x_4x_3x_3x_2) = (0.1)(0.2)(0.2)(0.3) = 0.0012$$

$$I(x_4x_3x_3x_2) = -\log_2 0.0012 = 9.7 \text{ b/símbolo}$$

(Hwei, 2003).

3. Con base en la información dada:

$$P(.) = 2P(-)$$

$$P(.) + P(-) = 3P(-) = 1$$

$$P(.) = \frac{2}{3}, P(-) = \frac{1}{3}$$

$$t(.) = 0.2 s \quad t(-) = 0.6 s \quad t(" ") = 0.2 s$$

Se obtiene la entropía de la información

$$H(X) = -\left(\frac{2}{3}\right) \log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right) \log_2\left(\frac{1}{3}\right)$$

$$H(X) = 0.92 \text{ b/símbolo}$$

El tiempo promedio por símbolo es

$$T_s = (P(.)t(.)) + (P(-)t(-)) + t(" ")$$

$$T_s = 0.5333 \text{ s/símbolo}$$

Por tanto, la velocidad r es

$$r = \frac{1}{T_s} = 1.875 \text{ símbolos/s}$$

Velocidad promedio de información del telégrafo es:

$$R = rH(X)$$

$$R = (1.875)(0.92) = 1.725 \text{ b/s}$$

(Hwei, 2003).

4. Se calcula la entropía de la información:

$$I(H) = -\log_2(0.1) = 3.32$$

$$I(o) = -\log_2(0.2) = 2.32$$

$$I(l) = -\log_2(0.1) = 3.32$$

$$I(a) = -\log_2(0.1) = 3.32$$

$$I(_) = -\log_2(0.1) = 3.32$$

$$I(m) = -\log_2(0.1) = 3.32$$

$$I(u) = -\log_2(0.1) = 3.32$$

$$I(n) = -\log_2(0.1) = 3.32$$

$$I(d) = -\log_2(0.1) = 3.32$$

$$H(X) = (0.1 * 3.32) + (0.2 * 2.32) + (0.1 * 3.32)$$

$$+ (0.1 * 3.32) + (0.1 * 3.32)$$

$$+ (0.1 * 3.32) + (0.1 * 3.32)$$

$$+ (0.1 * 3.32) + (0.1 * 3.32)$$

$$H(X) = 3.12 \text{ b/símbolo}$$

Actividad 3. Codificación de fuente

ACTIVIDAD 3. CODIFICACIÓN DE FUENTE

Arrastre la estampilla de "código libre de prefijo" o "código no libre de prefijo" sobre cada uno de los códigos mostrados e ingrese la longitud promedio para cada caso:

Código	$P(x_i)$
a	10 1/8
b	01 1/8
c	5/8
d	010 1/8

Código	$P(x_i)$
a	10 1/16
b	01 1/8
c	1 3/8
d	101 7/16

Código	$P(x_i)$
a	1 1/5
b	01 2/5
c	1 3/10
d	011 1/10

Código	$P(x_i)$
a	000 3/10
b	00 1/10
c	2/5
d	1 1/5

Código	$P(x_i)$
a	00 3/8
b	01 2/5
c	1/8
d	11 1/10

Código	$P(x_i)$
a	01 1/2
b	00 1/5
c	01 1/5
d	0000 1/10

Introduce los valores de L aproximados a 2 decimales.

Volver
Reintentar

Actividad 4. Codificación Huffman

Parte 1:

ACTIVIDAD 4. CODIFICACIÓN HUFFMAN

Para el fragmento de texto mostrado, obtenga el árbol del código Huffman correspondiente: *LAS_MAQUINAS_NO_PUEDEN_PENSAR*

X	P(x _i)
L	1/29 ✓
M	1/29 ✓
Q	1/29 ✓
R	1/29 ✓
I	1/29 ✓
O	1/29 ✓
D	1/29 ✓
U	2/29 ✓
P	2/29 ✓
S	3/29 ✓
E	3/29 ✓
A	4/29 ✓
N	4/29 ✓
-	4/29 ✓

Los valores mostrados en los nodos corresponden al numerador de la suma de las probabilidades.

Parte 2:

ACTIVIDAD 4. CODIFICACIÓN HUFFMAN

Con el árbol del código Huffman generado, seleccione la palabra que decodifica correctamente las secuencias binarias dadas:

10000001110

SON LAS QUE

110101111000110111110

SOMOS SALIR SOLOS

111111100001010000010

PUERRO PUEDEN PESAS

111100001011000110011

PENSAR PESAN PERAS

Los valores mostrados en los nodos corresponden al numerador de la suma de las probabilidades.

Volver Reintentar

Actividad 5. Códigos Aritméticos

Dado el texto inicial, se procede a obtener las probabilidades de cada símbolo x_i y el valor de su respectiva FDA:

x_i	$P(x_i)$	$F_X(x_i)$
O	0.341772152	0.341772152
_	0.202531646	0.544303797
C	0.202531646	0.746835443
P	0.075949367	0.82278481
M	0.063291139	0.886075949
R	0.037974684	0.924050633
Y	0.025316456	0.949367089
E	0.012658228	0.962025316
N	0.012658228	0.974683544
Q	0.012658228	0.987341772
U	0.012658228	1

Partiendo de $low = 0$ y $high = 1$, se procede a encontrar los sub-intervalos para cada uno de los símbolos de la secuencia:

Símbolo	Cálculo low y high	
C	Low	$0 + (1 - 0) (0.544303797) = 0.544303797$
	High	$0 + (1 - 0) (0.746835443) = 0.746835443$
O	Low	$0.544303797 + (0.746835443 - 0.544303797) (0) = 0.544303797$
	High	$0.544303797 + (0.746835443 - 0.544303797) (0.341772152) = 0.613523474$
M	Low	$0.544303797 + (0.613523474 - 0.544303797) (0.82278481) = 0.601256696$
	High	$0.544303797 + (0.613523474 - 0.544303797) (0.886075949) = 0.605637688$
P	Low	$0.601256696 + (0.605637688 - 0.601256696) (0.746835443) = 0.604528576$
	High	$0.601256696 + (0.605637688 - 0.601256696) (0.82278481) = 0.60486131$
R	Low	$0.604528576 + (0.60486131 - 0.604528576) (0.886075949) = 0.604823403$
	High	$0.604528576 + (0.60486131 - 0.604528576) (0.924050633) = 0.604836039$
O	Low	$0.604823403 + (0.604836039 - 0.604823403) (0) = 0.604823403$
	High	$0.604823403 + (0.604836039 - 0.604823403) (0.341772152) = 0.604827722$

Como resultado, se genera la etiqueta o *tag* = **0. 604824**.

Actividad 6. Compresión basada en contexto

1. Respuestas conceptos:

- RLE consiste en “Agrupar secuencias de caracteres idénticos en términos de su frecuencia”.
- El caracter virtual “escape” se envía “Para las no coincidencias de contexto en el orden actual”
- La codificación que reorganiza el orden de los símbolos de mayor a menor frecuencia es “MTF”.
- La codificación RLE para la secuencia “aaabbccccddd” es: 3a2b4c4d.
- Las secuencias de código cortas como las generadas por RLE pueden ser comprimidas mediante “Codificación Huffman” debido a su buen desempeño para este tipo de secuencias.

2. Para la cadena “sshtth_ii_e” se asume el alfabeto fuente $A=\{_,e,h,i,s,t\}$ al cual se asigna un orden inicial y se procede con la codificación MTF (Sayood, 2017):

Iteración	Lista símbolos indexados						Salida
	0	1	2	3	4	5	
sshtth_ii_e	_	e	h	i	s	t	4
sshtth_ii_e	s	_	e	h	i	t	0
sshtth_ii_e	s	_	e	h	i	t	3
sshtth_ii_e	h	s	_	e	i	t	5
sshtth_ii_e	t	h	s	_	e	i	0
sshtth_ii_e	t	h	s	_	e	i	1
sshtth_ii_e	h	t	s	_	e	i	3
sshtth_ii_e	_	h	t	s	e	i	5
sshtth_ii_e	i	_	h	t	s	e	0
sshtth_ii_e	i	_	h	t	s	e	1
sshtth_ii_e	_	i	h	t	s	e	5
sshtth_ii_e	Codificación MTF final:						40350135015

3. La “d” no existe en el contexto 0, por lo cual se añade con frecuencia 1; con este nuevo símbolo se incrementa la frecuencia de “escape” de 5 a 6, lo que aumenta las frecuencias totales a 21 (Salomon y Motta, 2010).

<i>Símbolo</i>	<i>f</i>	<i>p</i>
a	4	4/21
s	6	6/21
n	1	1/21
i	2	2/21
m	1	1/21
d	1	1/21
escape	6	6/21

4. Al recibirse el símbolo “d” el codificador inicia con la búsqueda de coincidencia sin éxito en los contextos existentes desde el orden 2 al orden 0, con lo cual envía 3 “escape” con las probabilidades de cada contexto, seguido de la codificación del símbolo “d” del orden -1. En la tabla siguiente se visualizan los bits requeridos (Salomon y Motta, 2010):

<i>Símbolo</i>	<i>Orden</i>	<i>p</i>	<i>Bits requeridos (I)</i>
escape	2	1/2	1
escape	1	2/5	1,32
escape	0	5/19	1,93
d	-1	1/28	4,81

Actividad 7. Lempel Ziv

1. LZ77

ACTIVIDAD 7. LEMPEL-ZIV

1. Para el texto que se muestra, seleccione la tripla correspondiente a la codificación LZ77 en cada iteración, considerando que tanto el *diccionario* como el *buffer de lectura anticipada* son de tamaño 10:

PEPE_PECAS_ PICA_PAPAS

Iteración 1	<0,P,0>	<0,0,PE>	<0,0,P> ✓	Iteración 6	<0,0,S> ✓	<0,1,A>	<1,1,S>
Iteración 2	<0,0,E> ✓	<1,0,E>	<0,2,PE>	Iteración 7	<6,0,P>	<6,2,I> ✓	<0,2,P>
Iteración 3	<2,0,P>	<2,2,_> ✓	<2,2,EP>	Iteración 8	<2,2,C>	<4,2,S>	<6,2,_> ✓
Iteración 4	<2,3,CA>	<3,2,C> ✓	<2,C,3>	Iteración 9	<5,1,A> ✓	<5,2,P>	<5,2,A>
Iteración 5	<2,3,A>	<0,1,A>	<0,0,A> ✓	Iteración 10	<3,1,P>	<2,1,A>	<2,2,S> ✓

Volver

➡
Reintentar

2. A continuación, detallan las iteraciones de la codificación LZW para la cadena:
PEPE_PECAS_PICA_PAPAS

S	x	S · x	S · x en diccionario	Salida	Adicionar al diccionario
""	P	P	SI		
P	E	PE	NO	5	8->PE
E	P	EP	NO	3	9->EP
P	E	PE	SI		
PE	_	PE_	NO	8	10->PE_
_	P	_P	NO	7	11->_P
P	E	PE	SI		
PE	C	PEC	NO	8	12->PEC
C	A	CA	NO	2	13->CA
A	S	AS	NO	1	14->AS
S	_	S_	NO	6	15->S_
_	P	_P	SI		
_P	I	_PI	NO	11	16->_PI
I	C	IC	NO	4	17->IC
C	A	CA	SI		
CA	_	CA_	NO	13	18->CA_
_	P	_P	SI		
_P	A	_PA	NO	11	19->_PA
A	P	AP	NO	1	20->AP
P	A	PA	NO	5	21->PA
A	S	AS	SI		

Al llegar al final de la cadena se escribe a la salida el índice del buffer $S \cdot x$ almacenado, obteniendo entonces el código LZW de la cadena de entrada:

Salida= 5-3-8-7-8-2-1-6-11-4-13-11-1-5-14.

Actividad 8. Cuantización

Mapeo de digitalización para la codificación (A/D) – reconstrucción D/A de 3 bits para la función $f(t) = 4 \cos(2\pi t)$ (Sayood, 2017):

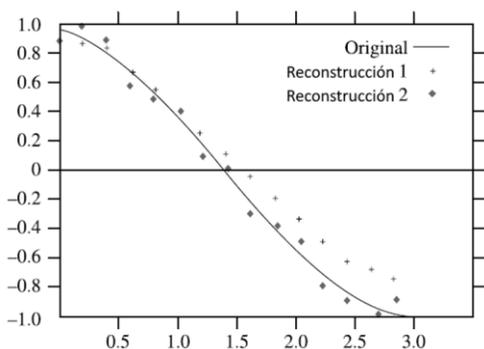
t	$4 \cos(2\pi t)$	A/D Output	D/A Output	Error
0.05	3.804	111	3.5	0.304
0.10	3.236	111	3.5	-0.264
0.15	2.351	110	2.5	-0.149
0.20	1.236	101	1.5	-0.264

Actividad 9. Codificación diferencial

1. A continuación, se aprecian los rangos de valores y error para la senoide de ejemplo (Sayood, 2017):

	Rango dinámico de valores	Tamaño de paso para un cuantizador uniforme de 4 niveles	Rango del error de cuantización
Senoide	[-1, 1]	0.5	[-0.25, 0.25]
Diferencias muestra a muestra (1-21)	[-0.2, 0.2]	0.1	[-0.05, 0.05]
Diferencias muestra a muestra (25)	[-0.4, 0.4]	0.2	[-0.1, 0.1]

2. Las expresiones para las reconstrucciones de la sinusoidal son (Sayood, 2017):



$$\text{Reconstrucción 1(1-21): } d_n = x_n - x_{n-1}$$

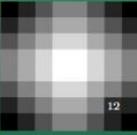
$$\text{Reconstrucción 1(1-25): } d_n = x_n - \hat{x}_{n-1}$$

Actividad 10. DCT - Cuantización - IDCT

Se calcula la IDCT de la matriz cuantizada (c) con lo cual se obtiene:

ACTIVIDAD 10. DCT - CUANTIZACIÓN - IDCT

Rendimiento de la DCT cuando se aplica a una imagen de tonos continuos.
 Se comienza con el patrón altamente correlacionado de la figura (a). Los 64 coeficientes de la DCT de este patrón se muestran en la figura (b) (unos pocos coeficientes dominantes). La figura (c) muestra los coeficientes después de haber sido toscamente cuantizados, por lo que solo permanecen 4 coeficientes no nulos. Halle los resultados de aplicar la IDCT a estos coeficientes cuantizados para completar la matriz de la figura (d).



(a)

239	1.19	-89.76	-0.28	1.00	-1.39	-5.03	-0.79
1.18	-1.39	0.64	0.32	-1.18	1.63	-1.54	0.92
-89.76	0.64	-0.29	-0.15	0.54	-0.75	0.71	-0.43
-0.28	0.32	-0.15	-0.08	0.28	-0.38	0.36	-0.22
1.00	-1.18	0.54	0.28	-1.00	1.39	-1.31	0.79
-1.39	1.63	-0.75	-0.38	1.39	-1.92	1.81	-1.09
-5.03	-1.54	0.71	0.36	-1.31	1.81	-1.71	1.03
-0.79	0.92	-0.43	-0.22	0.79	-1.09	1.03	-0.62

(b)

239	1	90	0	0	0	0	0
0	0	0	0	0	0	0	0
-90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c)

0	9	21	29	29	21	8	0
9	17	29	38	38	29	17	8
21	30	42	50	50	41	29	21
30	38	50	59	59	50	38	29
30	38	50	59	59	50	38	29
21	30	42	50	50	41	29	21
9	17	29	38	38	29	17	8
0	9	21	29	29	21	8	0

(d)

Volver
Introduzca los valores aproximados al entero inferior más cercano.
Reintentar

Actividad 11. Compresión fractal de imágenes:

1. Tras aplicar la transformación indicada:
 - a. Los puntos transformados son: “(1, 0), (3, 0), (0, 1) y (2, 1)”.
 - b. Los que representan un: “Paralelogramo”
2. Para la imagen del ejercicio, se calculan los coeficientes a , b , c , d , e y f de la transformación efectuada.

Sistema de ecuaciones:

$$\begin{aligned} 7a + 8b + e &= 4 \\ 4a + 2b + e &= 1.3 \\ 7a + 3b + e &= 2 \end{aligned}$$

$$\begin{aligned} 7c + 8d + f &= 8 \\ 4c + 2d + f &= 9.6 \\ 7c + 3d + f &= 8 \end{aligned}$$

Solución:

$$\begin{aligned} a &= 1/10 \\ b &= 2/5 \\ e &= 1/10 \end{aligned}$$

$$\begin{aligned} c &= -8/15 \\ d &= 0 \\ f &= 176/15 \end{aligned}$$

E. Anexo: Productos académicos relacionados con el trabajo final

Anexo E.1: Ponencia en ELE 2019 – Premio Mejor Trabajo de Sección

Durante la fase de diseño del presente trabajo, se generó como producto el capítulo de libro “Diseño de un eBook interactivo para la enseñanza y el aprendizaje de la comprensión de datos” (Cedeño & Vargas, 2019), el cual contiene los resultados obtenidos en cuanto a los aspectos disciplinares, aspectos didácticos, estado del arte y metodología de diseño. Fue expuesto en el *Encuentro Latinoamericano De Educación - ELE 2019*, en cuya ponencia fue otorgado el premio como mejor trabajo de sección. Los resultados se encuentran publicados en el libro “Revolución en la Formación y la Capacitación para el Siglo XXI - Edición 2, Vol. II”, ISBN: 978-958-52333-3-1. A continuación, se adjunta el respectivo certificado:



ELE 2019

CONGRESO

ENCUENTRO LATINOAMERICANO DE EDUCACIÓN

CONCEDE EL PREMIO MEJOR TRABAJO DE SESIÓN A

YULIAN CEDEÑO O.

POR LA PRESENTACIÓN DE LA PONENCIA

Diseño de un eBook interactivo para la enseñanza y el aprendizaje de la comprensión de datos



© 2019 Instituto Antioqueño de Investigación


Director ELE
Octubre 11 de 2019

Seguidamente, se incluye la primera pagina del capítulo de libro publicado:

Diseño de un eBook interactivo para la enseñanza y aprendizaje de la compresión de datos

Yulian Cedeño O.¹

Edgar Miguel Vargas C.²

Universidad Nacional de Colombia – Colombia

La teoría de la información se ha consolidado como objeto de estudio en pregrados y posgrados de carreras afines a su contenido, es la disciplina que rige los fundamentos teóricos de la transmisión y el procesamiento de la información permitiendo su medición y representación matemática; como una rama de esta ciencia se encuentra la compresión de datos cuyo fin es el de interpretar la información contenida en un conjunto de elementos o símbolos y a partir de ello obtener una representación más compacta que contenga una menor cantidad de datos. El estudio de los conceptos de estas disciplinas presenta un grado de dificultad considerable, por tal razón, en este trabajo se plantea el diseño de un eBook interactivo como herramienta de aprendizaje, correspondiente a un ambiente virtual en el que se explicaran interactivamente los conceptos de la teoría de la información referentes a la compresión de datos, mediante el uso de material multimedia como texto, imágenes, animaciones entre otros, con lo que se espera mejorar la eficacia del proceso de enseñanza y aprendizaje.

1. INTRODUCCIÓN

El uso de las matemáticas para el análisis de la información fue introducido como área de estudio por Claude Shannon (Yeung, 2002), tras la publicación de su trabajo en la década de los 40 (Shannon, 1948), momento desde el cual la información se define matemáticamente (Stone, 2015), con lo que se establece la base teórica para la comprensión y el diseño de todo sistema que incluya el transporte de datos. En la actualidad la teoría de la información se ha consolidado como objeto de estudio en pregrados y posgrados de las carreras afines a su contenido. La teoría de la información es la disciplina que rige la transmisión y el procesamiento de la información y se ocupa de la medición y de la representación matemática de la misma. Para ello se tiene como principio que la información aportada por un evento es directamente proporcional a la incertidumbre que se tiene sobre este, dando paso a la inclusión de la probabilidad como la herramienta base para el análisis de la información. De esta manera se genera la expresión mostrada en la ecuación (1), que cuantifica la información de los símbolos de una fuente de información F que genera los símbolos $\{x_1, x_2, \dots, x_n\}$ con probabilidades $P = \{p_1, p_2, \dots, p_n\}$ respectivamente, formulada por Claude Shannon en 1948 (Shannon, 1948):

$$I(x_i) = -\text{Log}(p_i) \quad (1)$$

Shannon también determinó la información promedio generada por la fuente denominada Entropía H , mediante la expresión de la ecuación (2) mostrada y que representa el *ideal de compresión de datos* hasta el momento inalcanzable:

$$H(F) = -\sum_{i=1}^n p_i * \log(p_i) \quad (2)$$

La compresión de datos es la codificación de forma eficiente de los datos, de tal forma que los datos resultantes presenten un menor tamaño (Fowler y Yagel, 1994). Esto se logra mediante diferentes estrategias, propias de cada algoritmo de compresión, pero en general la mayoría buscan algún tipo de redundancia en los datos originales, la cual tratan de eliminar con el fin de reducir la cantidad de los mismos. Teniendo en cuenta este aspecto, es necesario evaluar el tipo de información a comprimir, y si ésta presenta redundancia, por tanto, no existe un algoritmo de compresión universal, puesto que la distribución de los datos a comprimir depende directamente del tipo de información contenida, bien sea texto, imágenes, etc. (Salomón, 2002).

Los algoritmos de compresión son medidos con base en su *relación de compresión*, la cual está definida como el tamaño de los datos originales dividido por el tamaño de los datos tras la compresión, de esta forma, si 10 bytes son comprimidos en 1 byte, tenemos una tasa de compresión de 10, es decir que el archivo original se redujo en 10 veces su tamaño (Deng y Yang, 2010).

El término eBook (*electronic book*) o libro electrónico se utiliza para denominar los libros digitales o la versión electrónica de un libro físico, estos eBooks pueden ser visualizados en un sin número de dispositivos tales como PC, smartphone, PDAs, e-readers como el Kindle Amazon, entre otros dispositivos electrónicos de bolsillo (Mutalib et al., 2012; Sepúlveda y Quiceno, 2015). La historia de los eBook se remonta a 1972 con el proyecto Gutenberg, cuyo objetivo central era la digitalización de libros físicos para ser distribuidos libremente, lo cual no tuvo gran crecimiento debido al limitado acceso a los equipos de cómputo en la época. En 1998 con la salida al mercado del Rocket eBook como el primer equipo especializado para la lectura de eBooks, *e-readers*, inicia el auge de los eBooks con un crecimiento exponencial impulsado los frecuentes avances tecnológicos en estos dispositivos que incrementan la atraktividad de los eBooks (Jin, 2014).

¹ ycedenoo@unal.edu.co

² emvargasc@unal.edu.co

Anexo E.2: Ponencia en ELE 2021 – Premio Mejor Trabajo de Sección

Tras la implementación del “eBook Interactivo - Compresión de Datos” se generó como producto el capítulo de libro “Implementacion y evaluación de un e-book interactivo para la enseñanza-aprendizaje de la compresión de datos” (Cedeño & Vargas, 2021). Este producto fue expuesto vía virtual durante la pasada pandemia del COVID-19 y fue premiada como mejor trabajo de sección. Los resultados se encuentran publicados en el libro “Revolución en la Formación y la Capacitación para el Siglo XXI - Edición 4, Vol. II”, ISBN: 978-958-53278-8-7. A continuación, se adjunta el respectivo certificado:



Seguidamente, se incluye la primera página del capítulo de libro publicado:

Implementación y evaluación de un *eBook* interactivo para la enseñanza-aprendizaje de la comprensión de datos

Yulian Cedeño Ocampo¹
Edgar Miguel Vargas Chaparro²
Universidad Nacional de Colombia
Colombia

En el estudio de temáticas abstractas, como es el caso de la comprensión de datos, los estudiantes encuentran dificultad en su aprendizaje debido a la ausencia de recursos educativos digitales interactivos (Echeveste y Martínez, 2016). Los eBooks interactivos, mediante la integración de elementos multimedia, permiten nuevas prácticas de aprendizaje de mayor interacción. Este trabajo hace parte de una investigación cuyo objetivo es el análisis, diseño, desarrollo, implementación y evaluación de un eBook interactivo para la enseñanza-aprendizaje de la comprensión de datos. En este capítulo se presenta la implementación del eBook interactivo sobre el público objetivo y su posterior evaluación, para lo cual se hace uso de la metodología de investigación cuantitativa descriptiva aplicada a los estudiantes del curso de Teoría de la información y sistemas de comunicación de la Universidad Nacional de Colombia. Para la implementación se realiza una encuesta de contexto al público objetivo, se identifican y mitigan las amenazas a la validez que podrían afectar la investigación, se realiza el diseño de una prueba de *pre/post test* y de una prueba de usabilidad, y se diseña y ejecuta una estrategia instruccional en la que se aplica el *pre-test* al público objetivo, se realiza la intervención educativa utilizando el eBook interactivo, y se aplica el *post-test* y la prueba de usabilidad al público objetivo. Finalmente, se evalúa la efectividad de la acción formativa en el público objetivo, analizando estadísticamente los resultados obtenidos en las pruebas *pre-test* y *post-test*, y se evalúa la usabilidad del eBook, es decir, la calidad de la experiencia que tiene el usuario al interactuar con él.

¹ Ingeniero Electrónico, Estudiante de Maestría en Ingeniería.

Contacto: ycedenoo@unal.edu.co

² Ingeniero de Sistemas, Especialista en Ingeniería de Telecomunicaciones, Especialista en Ciencias Físicas y Magíster en Ciencias.

Contacto: emvargasc@unal.edu.co

Anexo E.3: Ponencia en LACCEI 2022

Abordando el detalle de las técnicas de compresión de datos, se generó como producto el capítulo de libro “Diseño y Desarrollo de un Software Interactivo para la Enseñanza - Aprendizaje de la Compresión Fractal de Imágenes” (Vargas & Cedeño, 2022), presentado vía virtual en Latin American and Caribbean Consortium of Engineering Institutions - LACCEI 2022. Los resultados se encuentran publicados en el libro “Education, Research and Leadership in Post-pandemic Engineering: Resilient, Inclusive and Sustainable Actions”, ISBN: 978-628-95207-0-5. A continuación, se adjunta el respectivo certificado:



Seguidamente, se incluye la primera página del capítulo de libro publicado:

Design and Development of an Interactive Software for the Teaching-Learning of Fractal Compression of Images

Edgar Miguel Vargas Chaparro, M.Sc.¹, Yulian Cedeno Ocampo, M.Sc. (c).¹
¹Universidad Nacional de Colombia, Colombia, emvargasc@unal.edu.co, ycedenoo@unal.edu.co

Abstract– In the study of courses with complex topics, as is the case of data compression, students experience difficulty in their learning due to the abstract nature of these theories and the absence of interactive digital resources. Fractal image compression seeks to deduce a set of functions, called iterated function system (IFS), which by iterations reconstructs the image to be compressed; In this way, compression levels are achieved that are not achieved using other techniques, because the image is considered compressed in the description of the functions that generate it, this is what is stored, and to recover the image, the functions are simply iterated. This article describes the design and development of interactive software that supports the teaching-learning of the fundamental concepts of fractal compression of images: contractive affine transformations, iterated function systems (IFS), IFS encoding (compression) and decoding (decompression) IFS.

Keywords — Fractal compression, images, interactive software, teaching, learning.

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2022.1.1.429>

ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

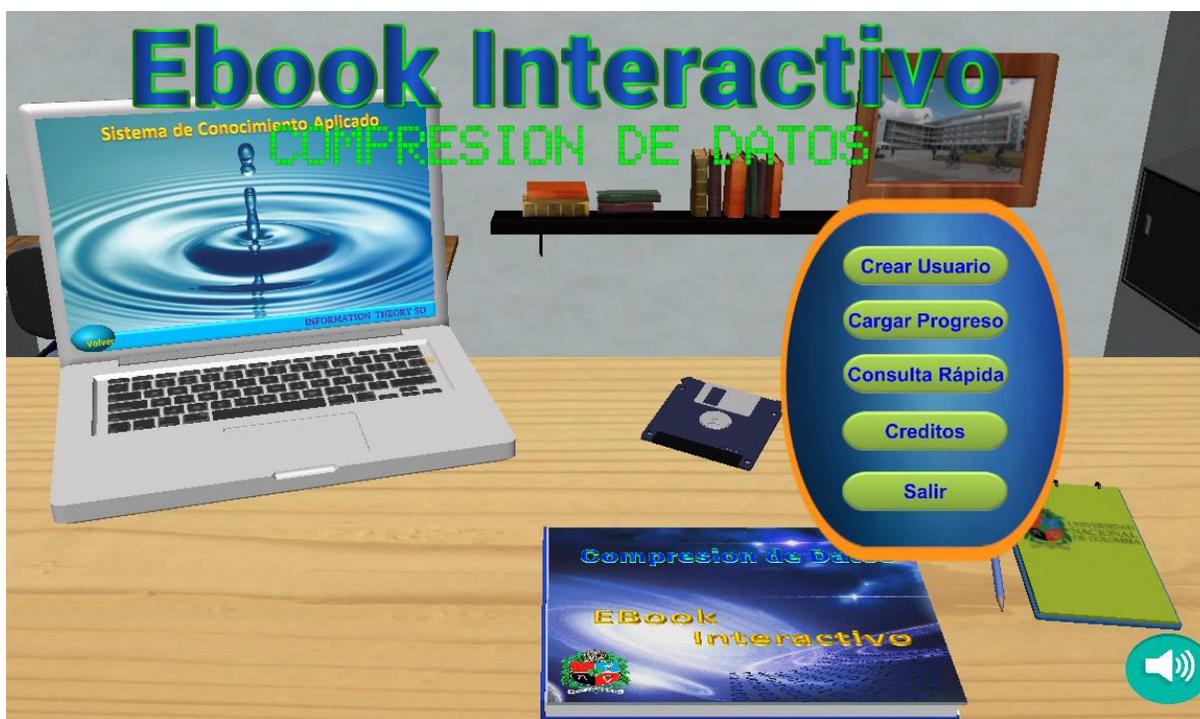
F. Anexo: Software eBook Interactivo Compresión de Datos.

El eBook interactivo diseñado y desarrollado en el presente trabajo puede ser descargado desde el enlace de Drive compartido, en el que reposan las versiones actualizadas del eBook para las plataformas:

<https://drive.google.com/drive/u/1/folders/1MYzBNGHm8aOgo1WIB6RG4svrVojULHvk>

A continuación, se muestran algunas imágenes del funcionamiento del eBook interactivo Compresión de Datos.

Vista inicio eBook Interactivo:



Dashboard - PC interactivo:

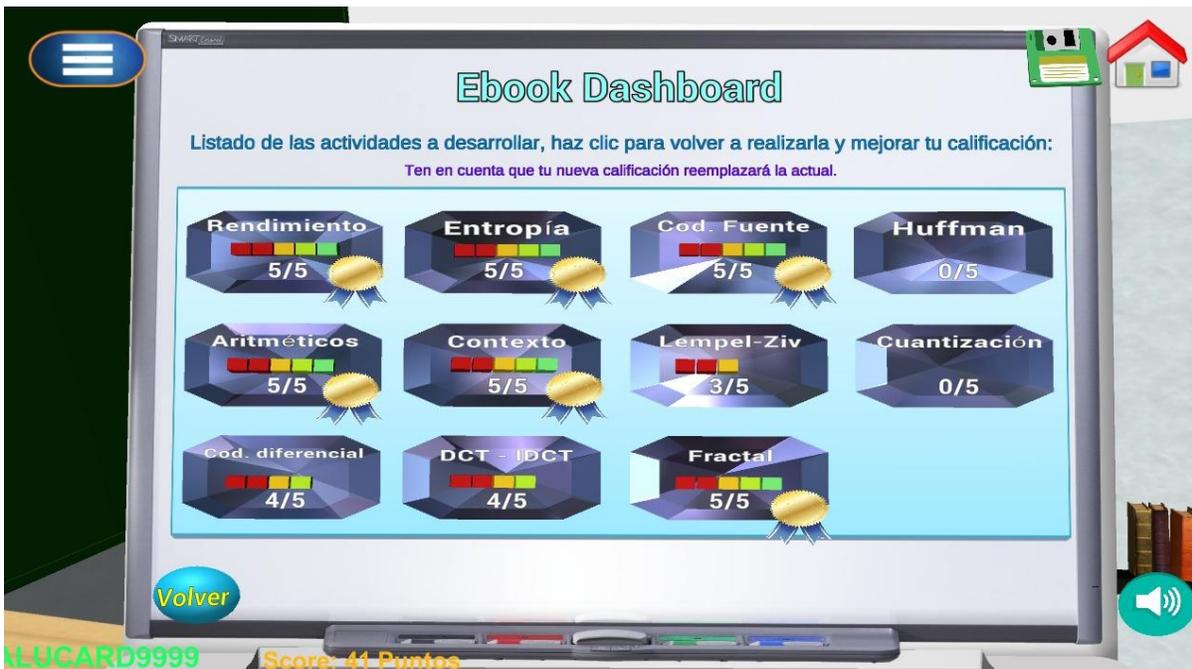
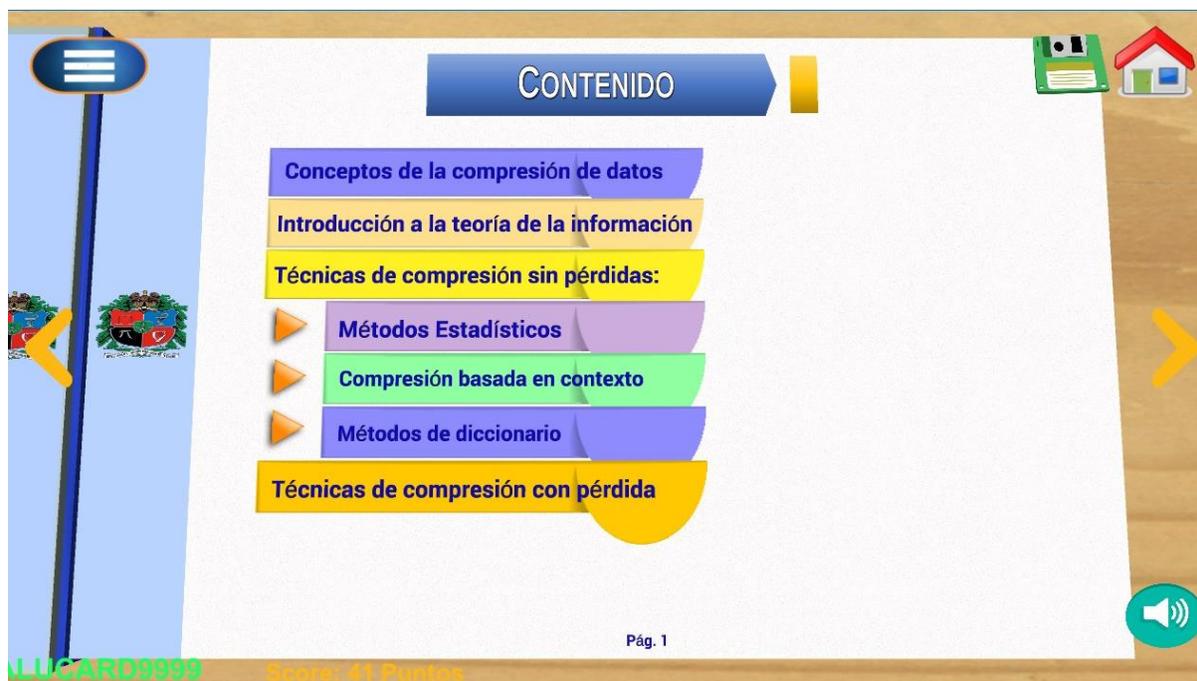


Tabla de contenido interactiva:



Desarrollo de contenido de enseñanza:

CÓDIGOS ARITMÉTICOS

Iteraciones paso 3:

n=1
Secuencia CA
n=3
n=4

Tras obtener el intervalo anterior $[0, 0.25)$, se procede con la siguiente subdivisión para la secuencia CA:

$$newlow = 0 + (0.25 - 0) \cdot 0.25 = \mathbf{0.0625}$$

$$newhigh = 0 + (0.25 - 0) \cdot 0.75 = \mathbf{0.1875}$$

El intervalo obtenido para la secuencia CA es $[0.0625, 0.1875)$:

[Volver](#)

LUCARD9999 Score: 41 Puntos

DECODIFICACIÓN ARITMÉTICA

Ejemplo:
Continuando con el ejemplo anterior, para la secuencia CASA se obtuvo el $tag = 0.17$, para el cual a continuación se realizará el proceso de decodificación.

Se calcula el valor de $newtag$.

$$newtag = \frac{(0.86 - 0.75)}{(1 - 0.75)} = 0.44$$

Se obtiene el símbolo x_i con base en el rango de la FDA en el que se encuentra $newtag$:

Secuencia decodificada: **CASA**

Tras realizar las iteraciones partiendo desde el $tag=0.17$, se obtiene la secuencia decodificada **CASA**.

$n = 4$

Pág. 23

LUCARD9999 Score: 41 Puntos

Referencias bibliográficas

- Adawiyah, R., Harjono, A., & Gunawan, G. (2018). The Development of Interactive Physics E-Book in Rigid Body Equilibrium and Rotational Dynamics. *Journal of Research & Method in Education* (8)2, pp. 29-33.
- Armañanzas, E. (2010). *El libro electrónico, una gran herramienta para la cultura y la educación*, Universidad del País Vasco, Lejona, España
- Allen, W. (2006) Overview and Evolution of the ADDIE Training System, *Adv. Dev. Hum. Resour.* 8.
- Alshaya, H., & Oyaid, A. (2017). *Journal of Education and Practice* www.iiste.org ISSN. 8(8). www.iiste.org
- Barnsley, M. F., (2012), *Fractals Everywhere: Third Edition*, Dover Publications, New York.
- Barnsley, M. F., & Sloan, Alan D. (1988). A Better Way to Compress Images: Mathematics is providing a novel technique for achieving compression ratios of 10000 to 1 - and higher.
- Blelloch, G. (2013). *Introduction to Data Compression*, Pittsburgh PA: Carnegie Mellon University
- Binas, M., Štancel, P., Novák, M., y Michalko, M., (2012). *Interactive eBook as a Supporting Tool for Education Process*, Košice, Slovakia: Technical university of Košice.
- Bravo Paniagua, T., & Valenzuela Gonzáles, M. R. (2019). Desarrollo de instrumentos de evaluación: cuestionarios. Centro de Medición MIDE UC, Instituto Nacional para la Evaluación de la Educación INEE, 5. <https://www.inee.edu.mx/wp-content/uploads/2019/08/P2A355.pdf>
- Bruce, A. T. (2012). Quasi-Experimental Research Designs. En *Etica e Politica* (Vol. 15, Número 1). <https://doi.org/10.1093/acprof>
- Camargo, J. M. (2008). *El libro electrónico: La industria editorial en la era de la revolución digital*, Bogotá: Pontificia Universidad Javeriana.
- Cedeño, Y., & Vargas, E. M. (2019). Diseño de un eBook interactivo para la enseñanza y aprendizaje de la compresión de datos. En *Revolución en la formación y la capacitación para el siglo XXI* (2a ed., Vol. II, pp. 779–788). Instituto Antioqueño de Investigación.
- Cedeño, Y., & Vargas, E. M. (2021). Implementacion y evaluación de un e-book interactivo para la enseñanza-aprendizaje de la compresión de datos. En *Revolución en la formación y la capacitación para el siglo XXI* (4a ed., Vol. II, pp. 290–301). Instituto Antioqueño de Investigación.
- Cordón García, J. A., & Lopes, C. A. (2012). The eBook: invariances and transformations. *Profesional De La Informacion*, 21(1), 83–90. <https://doi.org/10.3145/epi.2012.ene.11>
- Droutsas, S., Patsilnakos, P., & Symvonis, A. (2018). Interactive Personalized eBooks for Education. *IEEE*.
- Elton, J. A., (1988), *Simultaneously Contractive Remetrization Theorem for Iterated Function Systems*. Georgia Institute of Technology Preprint.
- Flores-Ruiz, E., Miranda-Navales, M., & Villasís-Keever, M. (2017). El protocolo de investigación VI: cómo elegir la prueba estadística adecuada. *Estadística inferencial. Rev Alergia Mexico*, 64(3), 364–370. <http://www.revistaalergia.mx>
- Figueiredo, M., Bidarra, J., y C. Natalio. (2014). *Designing eBook Interaction for Mobile and Contextual Learning*, Portugal.

- Figueiredo, M., y Bidarra J. (2015). The development of a Gamebook for education. *ScienceDirect*. (67) pp. 322-331.
- García-Valcárcel, A. (2016). *Recursos digitales para la mejora de la enseñanza-aprendizaje*. [https://gredos.usal.es/bitstream/handle/10366/131421/Recursos digitales.pdf](https://gredos.usal.es/bitstream/handle/10366/131421/Recursos%20digitales.pdf)
- Gama R. M. (2002) El libro electrónico: del papel a la pantalla, Bogotá: Revista Nueva Epoca.
- Guzman, P., Dormido, B., y Costa-Castelló A. (2018) New Interactive Books for Control Education. *ScienceDirect*. 51(4) pp. 190-195
- Grabowski, E., Alten, F., Termühlen, J., Heiduschka, P., Brücher, V., Eter, N., & Clemens, C. R. (2020). Analysis of the importance of e-learning in ophthalmology and evaluation of an e-learning app. *Ophthalmologie*, 117(12), 1218–1224. <https://doi.org/10.1007/s00347-020-01100-x>
- Guerrero, F. G., & Pérez, L. A. (2008). A Software for Learning Information Theory basics with Emphasis on Entropy of Spanish. *Energía Y Computación*, 16(1), 59–65.
- Gallego, F., Molina R., y Llorens, F. (2014). Gamificar una propuesta docente, *Diseñando experiencias positivas de aprendizaje*, JENUI vol. X.
- Haladyna, T. M., Downing, S. M., & Rodriguez, M. C. (2002). A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment. *Applied Measurement in Education*, 15(3), 309–333. https://doi.org/10.1207/s15324818ame1503_5
- Hamada, M., & Hassan, M. (2017). A multimedia learning environment for information theory. *Proceedings of 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2016*, December, 55–61. <https://doi.org/10.1109/TALE.2016.7851770>
- Hernández, G., & Niño, L. (1992). *Geometría Fractal en Compresión de Imágenes Digitales*. Santafé de Bogotá: Universidad Nacional de Colombia, Facultad de Ingeniería, Departamento de Sistemas.
- Hoch, S., Reinhold, F., Werner, B. et al. (2018) Potencial de diseño e investigación de libros de texto interactivos: el caso de las fracciones. *ZDM Mathematics Education* 50, pp 839–848. <https://doi.org/10.1007/s11858-018-0971-z>
- Hwei, H., (2003). *Schaum's Outline of Theory and Problems of Analog and Digital Communications*. McGraw-Hill. United States of America.
- Jin, C. (2014). Adoption of eBook among college students: The perspective of an integrated TAM. *Computers in Human Behavior*.
- Kitaboo, Hurix Digital, 23 Octubre 2018. [En línea]. Available: <https://kitaboo.com/importance-of-ebooks-in-education/>.
- Lebert, M (2009). Una corta historia del ebook, Universidad de Toronto.
- McLeod, S. (2019). Likert Scale Definition, Examples and Analysis. Recuperado: <https://www.simplypsychology.org/likert-scale.html>.
- Morales, M. (2003). *Notas sobre Compresión de Datos*, San Andrés Cholula: INAOE.
- Mutalib, A., Noor, A. M., Hashim, H. M., & Mahari, R. (2012) “EBooks as textbooks in the classroom”. *Procedia - Social and Behavioral Sciences*.
- Pribeanu, C. (2017). A Revised Set of Usability Heuristics for the Evaluation of Interactive Systems. *Informatica Economica*, 21(3/2017), 31–38. <https://doi.org/10.12948/issn14531305/21.3.2017.03>
- Rodriguez Grimaldo, A. C., & Vargas Chaparro, E. M. (2019). Diseño y Desarrollo de un Sistema de Realidad Mixta para la Enseñanza-Aprendizaje de la Física de Agujeros Negros. En *Revolución en la Formación y la Capacitación para el Siglo XXI (3a ed., pp. 340–350)*. Instituto Antioqueño de Investigación.

- Sayood, K. (2017). Introduction to Data compression. En ACM Transactions on Multimedia Computing, Communications, and Applications. San Francisco CA. Elsevier Inc.
- Sánchez Morales, G., Mezura-Godoy, C., & Benítez-Guerrero, E. (2018). Proceso de Pruebas de Usabilidad de Software. *Tecnología Educativa Revista CONAIC*, (1).
- Salas, R. (2016). Diseño y análisis de un sistema web educativo considerando los estilos de aprendizaje. Editorial Areas de innovación y Desarrollo S.L. Primera Edición. ISBN: 978-84-945785-4-0
- Salomon, D., y Motta, G. (2010) Handbook of Data Compression. New York, EEUU; Springer. Fifth Edition.
- Suman C, Anjali, P, Sonal P. (2008). Study of LZ77 and LZ78 Data Compression Techniques
- Tejedor, F. (1981). Validez interna y externa en los diseños experimentales. *Revista española de pedagogía*. (39)151 pp. 15-39.
- Toca, C., y Carrillo, J., (2019). Los entornos de aprendizaje inmersivo y la enseñanza a ciber-generaciones. *Educação e Pesquisa* (45) 1-20 <https://doi.org/10.1590/s1678-4634201945187369>
- Rodríguez F., Jesús L., Martínez, Nerwis, y Lozada, Joan Manuel (2009). Las TIC como recursos para un aprendizaje constructivista. *Revista de Artes y Humanidades UNICA*. Disponible en: <https://www.redalyc.org/articulo.oa?id=1701/170118863007>
- Vargas, E. M., & Cedeño, Y. (2022). Diseño y Desarrollo de un Software Interactivo para la Enseñanza - Aprendizaje de la Compresión Fractal de Imágenes. En *Education, Research and Leadership in Post-pandemic Engineering: Resilient, Inclusive and Sustainable Actions*. Fundación LACCEI. Disponible en: https://laccei.org/LACCEI2022-BocaRaton/full_papers/FP429.pdf
- Welch, T, (1984). *Technique for High-Performance Data Compression*, St Louis.
- Wu, H. K., Lee, S. W. Y., Chang, H. Y., & Liang, J. C. (2013). Current status, opportunities and challenges of augmented reality in education. *Computers and Education*, 62, 41–49. <https://doi.org/10.1016/j.compedu.2012.10.024>