



UNIVERSIDAD NACIONAL DE COLOMBIA

Método de evaluación de usabilidad aplicada a productos de software que facilitan el acceso a herramientas informáticas de personas con enfermedades que afectan la motricidad

**Usability evaluation method applied to software products that provide access to software
tools for people with conditions affecting the motor functions**

Ing. Jhon Fernando Sánchez Álvarez

Universidad Nacional de Colombia

Facultad de Minas

Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

Junio de 2015



UNIVERSIDAD NACIONAL DE COLOMBIA

Método de evaluación de usabilidad aplicada a productos de software que facilitan el acceso a herramientas informáticas de personas con enfermedades que afectan la motricidad

Usability evaluation method applied to software products that provide access to software tools for people with conditions affecting the motor functions

Jhon Fernando Sánchez Álvarez

Tesis de investigación presentada como requisito parcial para optar al título de:
Magíster en Ingeniería de Sistemas

Línea de investigación:
Calidad de software

Director:

Ph.D. Jovani Alberto Jimenez Builes

Co-Director:

Ph.D. John Willian Branch Bedoya

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

Junio de 2015

“Si tu cuerpo te limita no dejes que tu mente también lo haga”

Stephen Hawking

AGRADECIMIENTOS

Agradezco a Jovani Alberto Builes quien con paciencia y ejemplo logró formar de mí un investigador. Y antes de ser mi director es mi amigo. También deseo agradecer al profesor John Willian Branch por su apoyo a lo largo de toda mi formación profesional.

A mi familia por ser parte fundamental de mi vida; sin su ayuda nada de esto sería posible.

A lo largo de esta aventura conocí personas espectaculares. Unas aún están. Otras se alejaron. Pero queda el cariño por todas... Como diría el protagonista de mi libro favorito "Viajé, amé, perdí, confié y me traicionaron". Es increíble pensar que en un libro de fantasía allá tanta verdad... En fin. Gracias totales.

RESUMEN

La computación es parte intrínseca de nuestro día a día. No obstante, existen enfermedades que imposibilitan su uso de manera eficiente. Por esto, numerosas organizaciones y centros de investigación desarrollan software para facilitar el acceso al uso de los computadores a las personas que padecen de alguna enfermedad que altera de manera significativa su motricidad. La evaluación de usabilidad del software a menudo se realiza de forma empírica. Esta tesis de maestría tiene por objetivo mostrar un método ponderado para evaluar la usabilidad de software diseñado para facilitar el acceso a la computación a personas con deficiencias motoras usando un enfoque heurístico.

La evaluación heurística tiene un gran potencial de adaptabilidad, de manera que puede ser usado para encontrar problemas de usabilidad en estos software. El propósito fue desarrollar un nuevo conjunto de heurísticas que pueden ser utilizados por los diseñadores para evaluar las maquetas y prototipos funcionales.

Palabras claves: heurísticas, usabilidad, software especializado en personas con discapacidad, interacción persona-ordenador, discapacidad motriz.

ABSTRACT:

The computer is an intrinsic part of our daily life. However, there are conditions that preclude its use efficiently. Thus, many institutions and research centers develop software to facilitate access to computers for people who suffer from a disease that significantly alters their motor skills. The usability evaluation of software is often done empirically. The article aims to show a weighted method for evaluating the usability of software designed to facilitate access to computing for people with mobility impairments using a heuristic approach.

Heuristic evaluation has great potential for adaptability, so that can be used to find usability problems in these software. The objective of this research was to develop a new set of heuristics that can be used by designers to evaluate the models and functional prototypes.

Keywords: heuristic, usability, specialized software for people with disabilities, human computer interaction, motor disabilities.

Tabla de contenido

| | | |
|--------|--|----|
| 1. | Introducción..... | 1 |
| 1.1. | Justificación..... | 1 |
| 1.2. | Planteamiento del problema..... | 2 |
| 1.3. | Objetivo general | 3 |
| 1.4. | Objetivos específicos | 3 |
| 1.5. | Método usado para la elaboración de la tesis | 3 |
| 1.5.1. | Definición de los objetivos | 3 |
| 1.5.2. | Diseño o elección del metamodelo | 3 |
| 1.5.3. | Caracterización del concepto a ser medido | 3 |
| 1.5.4. | Validación | 4 |
| 1.6. | Alcance del trabajo..... | 4 |
| 1.7. | Estructura de la tesis..... | 4 |
| 2. | Marco teórico | 5 |
| 2.1. | Calidad de software..... | 5 |
| 2.2. | Principios de calidad de software | 5 |
| 2.3. | Usabilidad | 5 |
| 2.3.1. | Atributos de usabilidad | 6 |
| 2.4. | Evaluación de Usabilidad | 8 |
| 2.4.1. | Inspección | 8 |
| 2.4.2. | Indagación | 9 |
| 2.4.3. | Test..... | 9 |
| 2.4.4. | Evaluación heurística de usabilidad..... | 10 |
| 2.4.5. | Escala de Usabilidad del Sistema..... | 12 |
| 2.5. | Interacción persona- ordenador | 13 |

| | | |
|--------|---|----|
| 2.6. | Enfermedades Neuromotoras | 13 |
| 2.6.1. | Esclerosis lateral amiotrófica | 14 |
| 2.6.2. | Parálisis Bulbar Progresiva..... | 14 |
| 2.6.3. | Distrofia muscular | 14 |
| 2.6.4. | Atrofia muscular espinal | 14 |
| 2.7. | Método Delphi | 15 |
| 3. | Antecedentes | 17 |
| 3.1. | Diseño de software para personas en condición de discapacidad..... | 17 |
| 3.2. | Adaptabilidad de la evaluación heurística a casos particulares | 18 |
| 3.3. | Evaluación de productos de software para personas con discapacidad motora..... | 20 |
| 4. | Solución propuesta | 22 |
| 4.1. | ¿Por qué es la evaluación heurística el mejor método para examinar software que facilitan el acceso a la computación de personas con enfermedades que alteran la motricidad?..... | 23 |
| 4.2. | Adquisición de características heurísticas..... | 24 |
| 4.2.1 | Principio de acción mínima | 27 |
| 4.2.2 | Prevención de posibles errores cuando se ejecuta una tarea..... | 27 |
| 4.2.3 | Affordance..... | 27 |
| 4.3. | Asignación de pesos a las heurísticas | 28 |
| 4.3.1. | Marco legal para la realización de la prueba | 30 |
| 4.3.2. | Cantidad de usuarios requeridos para la prueba | 32 |
| 4.3.3. | Nivel educativo de los usuarios | 34 |
| 4.4. | Evaluación por expertos del software..... | 37 |
| 5. | Validación..... | 41 |
| 5.1. | Usuarios | 42 |

| | |
|---------------------------------------|----|
| 5.2. Localización | 42 |
| 5.3. Procedimiento | 42 |
| 6. Conclusiones y trabajo futuro..... | 48 |
| 6.1. Conclusiones..... | 48 |
| 6.2. Trabajo futuro..... | 48 |
| Apéndice A..... | 49 |
| Apéndice B..... | 51 |
| 7. Referencias..... | 55 |

Listado de figuras

| | |
|--|----|
| Figura 1: Atributos de la usabilidad | 7 |
| Figura 2: Clasificación de los métodos de evaluación de la usabilidad. | 8 |
| Figura 3: Ventajas y desventajas de la evaluación heurística [31] | 12 |
| Figura 4: Proceso del Método Delphi | 16 |
| Figura 5: Categorías de Usabilidad | 25 |
| Figura 6: Diferencia entre pesos por heurística..... | 38 |
| Figura 7: Grafico de dispersión de escala de usabilidad del sistema, tiempo medio usado en la prueba y errores promedio en la prueba contra los software. | 45 |
| Figura 8: Grafico de Relación entre variables del software 1 | 46 |
| Figura 9: Grafico de Relación entre variables del software 2 | 46 |
| Figura 10: Validación de supuestos (residuales)..... | 51 |
| Figura 11: Normalidad..... | 51 |
| Figura 12: Validación de supuestos (residuales)..... | 52 |
| Figura 13: Normalidad..... | 52 |
| Figura 14: Validación de supuestos (residuales)..... | 53 |
| Figura 15: Normalidad..... | 53 |
| Figura 16: Validación de supuestos (residuales)..... | 54 |
| Figura 17: Normalidad..... | 54 |

Listado de tablas

| | |
|--|----|
| Tabla 1: Evaluación heurística..... | 26 |
| Tabla 2: Encuesta de asignación de pesos heurísticos..... | 28 |
| Tabla 3: Usuarios con enfermedades neurológicas que alteran la motricidad..... | 34 |
| Tabla 4: Pesos dados por los usuarios..... | 36 |
| Tabla 5: Pesos dados por los expertos..... | 37 |
| Tabla 6: Varianza en pesos heurísticos..... | 38 |
| Tabla 7: Puntuación de usabilidad de los diez software..... | 40 |
| Tabla 8: Modelo de regresión..... | 47 |

1. Introducción

1.1. Justificación

Según la Organización Internacional de Normalización (ISO por sus siglas en inglés) en el estándar 9241-11, define la usabilidad como *“el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso”* [1]. La usabilidad es un aspecto importante al momento de valorar la calidad de un software [2].

En la evaluación de la usabilidad normalmente se miden los comportamientos, gestos y el número de clics que realizan los usuarios al completar una actividad, pero este tipo de medición no aplica a las personas con requisitos especiales, debido a que poseen necesidades foráneas a las etapas de diseño que propone la ingeniería de software [3].

La medición de calidad de un software, se compone entre otros ítems, por la usabilidad del mismo. La usabilidad es un atributo que posee el software. Para afirmar que un software es usable, se debe realizar una medición de los parámetros de él. Existen métodos desarrollados para la evaluación de la usabilidad [4]–[9], que proponen métodos empíricos basados en la experiencia de un experto. No obstante, existe una necesidad creciente de un modelo que mida cuantitativamente la capacidad del producto de software en general, mientras que da un mayor nivel de importancia a ciertas heurísticas categorizadas por el usuario [10].

En un gran número, los desarrolladores ignoran este atributo de calidad, debido a la complejidad en su medida y a las numerosas publicaciones que proponen los diferentes métodos [11]. Esto genera un serio problema teniendo en cuenta que a un software se le invierte una cantidad considerable de recursos, y más aún, si tiene como objetivo facilitar la interacción persona-computador a seres humanos que no

pueden hacer uso de las nuevas tecnologías. O aun peor, desarrollan métodos para medir el producto desarrollado por ellos y así obtienen una “buena” calificación de su software.

1.2. Planteamiento del problema

Stucki *et al.* [12] aseguran que el 10% de la población mundial posee algún tipo de discapacidad. De este porcentaje un 10% posee una discapacidad motora. Alrededor de 720 millones de personas padece alguna discapacidad y de esta cifra, 72 millones tienen una discapacidad motriz.

El uso de los computadores está creciendo como parte de nuestra vida cotidiana. En las labores, en la educación y para acceder a entretenimiento e información. Sin embargo, los usuarios con deficiencias motrices les resulta dificultoso o imposible hacer uso de aplicaciones de software comunes [13]. Mientras, algunos expertos argumentan que los requisitos para que estas personas accedan a la tecnología esta solventada por medio de dispositivos de apoyo especializados. Estos poseen tres grandes defectos: su costo, complejidad y la alta demanda de mantenimiento son excesivos por lo que se abandonan tiempo después de su creación [14]–[16]. A menudo se percibe que diseñar artefactos exclusivos para las personas con discapacidad motriz es inviable económicamente debido a que es una población reducida [17]. El segundo defecto, es que estas se diseñan para el usuario promedio. Son inmutables, por ende, los usuarios con deficiencias motrices deben adaptarse a ellos [18]. Otros expertos justifican que estas limitaciones de hardware deben ser solventadas por medio de software que posibilite una mejor usabilidad de los periféricos de entrada [19].

Diferentes entidades y centros de investigación desarrollan hardware y software que facilitan a las personas con enfermedades que afectan su motricidad, hacer uso de las TIC. No obstante, la medición del software se realiza de forma empírica y subjetiva, sin tener en cuenta las heurísticas de los usuarios finales.

1.3. Objetivo general

Diseñar un método de evaluación de usabilidad aplicada a productos de software que facilitan el acceso a herramientas informáticas de personas con enfermedades que afectan la motricidad.

1.4. Objetivos específicos

- Elaborar y aplicar un cuestionario a expertos en usabilidad utilizando el método Delphi.
- Seleccionar los ítems para las pruebas de usabilidad utilizando los resultados del cuestionario.
- Elaborar y aplicar los ítems de las pruebas de usabilidad a los usuarios finales.
- Construir un modelo de regresión para la validación de los ítems de las pruebas de usabilidad.

1.5. Método usado para la elaboración de la tesis

El método usado para realizar esta tesis de maestría contemplan cuatro etapas propuestas por Jacquet *et al* [20] para el diseño de un método de medición, a saber:

1.5.1. Definición de los objetivos

La primera etapa del proceso es la que se fundamenta el trabajo. Consiste básicamente en obtener el marco conceptual y realizar una adecuada revisión de la literatura existente en el tema de la tesis. Y con ello, caracterizar el concepto que debe ser medido.

1.5.2. Diseño o elección del metamodelo

La segunda etapa propuesta es seleccionar el metamodelo. Consiste en establecer la relación entre el concepto y el software. Lo anterior, se logra con base en la revisión de literatura del paso preliminar.

1.5.3. Caracterización del concepto a ser medido

En la tercera etapa del desarrollo del método el concepto a medir debe ser claro. Es importante asignar las reglas de asignación numérica para las características establecidas en la anterior etapa.

1.5.4. Validación

En la cuarta y última etapa se ratifican los datos obtenidos por medio de un modelo regresión.

1.6. Alcance del trabajo

Un método de evaluación de usabilidad permite medir el “*grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso*” [1]. El método brinda a los desarrolladores de software lineamientos para obtener una medición de cuan usable es un software especializado.

El método de evaluación se valida mediante un modelo de regresión con los datos obtenidos de los criterios de los usuarios finales.

1.7. Estructura de la tesis

La presente tesis de maestría está estructurada de la siguiente manera: en el capítulo dos se presenta el marco teórico; en el capítulo tres se presentan los antecedentes; en el capítulo cuatro se presenta el método de evaluación de software propuesto; en el capítulo cinco se muestran los resultados de la validación y, finalmente, en el capítulo seis se presentan las conclusiones y el trabajo futuro que se deriva de este trabajo.

2. Marco teórico

2.1. Calidad de software

La Organización Internacional de Normalización (ISO por sus siglas en Inglés), creó un proyecto llamado SQuaRE (Software Product Quality Requirements and Evaluation). El cual tiene como objetivo unificar procesos de requisitos de la calidad de software y su evaluación. Definen la calidad de software como el grado en que un producto satisface las necesidades implícitas cuando se utiliza en condiciones especificadas [21].

2.2. Principios de calidad de software

En la calidad de software existen seis principios planteados por Watts Humphrey [22]:

- Un cliente que no demanda calidad, posiblemente no la obtendrá.
- Para obtener calidad de un producto, se debe gestionar de manera continua.
- Se debe medir la calidad en todo el proceso.
- La calidad de un producto es directamente proporcional al proceso usado para construirlo.
- Las pruebas de calidad deben estar presentes para asegurar la misma.
- La motivación de los profesionales que desarrollan es factor clave a la hora de asegurar la calidad de un producto.

2.3. Usabilidad

ISO 25010:2011 define la usabilidad como el grado en que un producto o sistema puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico.

Hay gran cantidad de publicaciones que tratan de abordar el tema de medir y cuantificar la usabilidad, un ejemplo de esto es el estándar ISO 9241-11 (1998) que define la usabilidad en términos de eficiencia, eficacia y satisfacción del usuario [11].

Nielsen, [23] especifica cinco descomposiciones que definen el termino usabilidad:

- Facilidad de aprendizaje.
- Eficiencia.
- Recuerdo en el tiempo.
- Tasa de errores.
- Satisfacción.

El proyecto ISO 25010:2011: SQuaRE, plantea seis subcategorías de la usabilidad:

- Conveniencia reconocible.
- Aprendibilidad.
- Operatibilidad.
- Predicción de errores.
- Interfaz estética.
- Accesibilidad.

2.3.1. Atributos de usabilidad

Para Nielsen[24], la usabilidad es una cualidad abstracta del software para ser cuantificada claramente. Para medirla se requiere realizar una descomposición. Nielsen especifico cinco atributos, a saber:

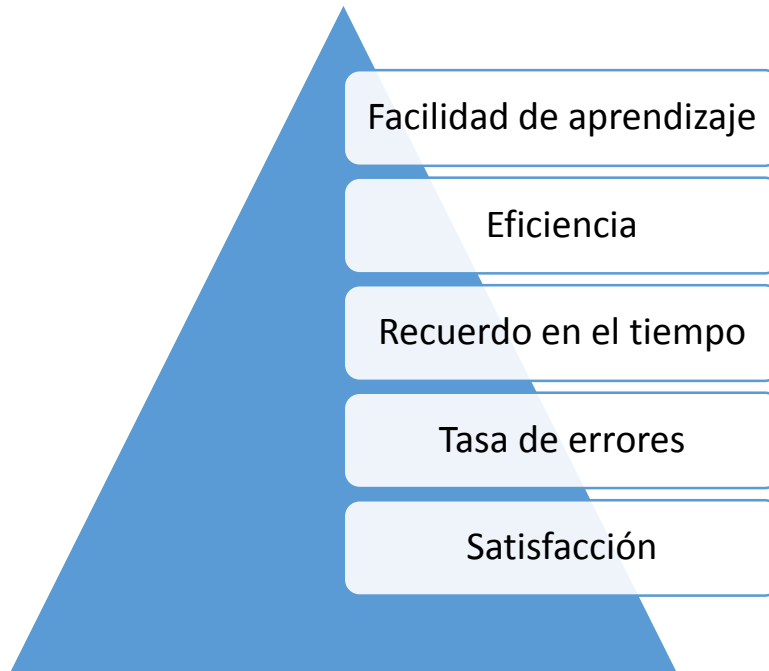


Figura 1: Atributos de la usabilidad

- Facilidad de aprendizaje. Se refiere a la facilidad de aprender la funcionalidad básica del software, para que el usuario realice la tarea correctamente.
- Eficiencia. Es la cantidad de operaciones realizadas por unidad de tiempo usando el software. Si la usabilidad del software es alta, más rápido es el usuario al utilizarlo. Esto no implica que la eficiencia del usuario sea alto debido a una alta eficiencia del software.
- Recuerdo en el tiempo. Es importante para los usuarios que no usan el software con regularidad que recuerden el funcionamiento del mismo. Este atributo refleja el recuerdo dejado por el software en el usuario, tras un tiempo de no usarlo.
- Tasa de errores. Este atributo se refiere a la cantidad de errores cometidos por el usuario. Entre más alto sea este atributo menor usabilidad posee un software determinado.
- Satisfacción. Atributo subjetivo. Explica la impresión del usuario del software [25].

2.4. Evaluación de Usabilidad

Las pruebas o evaluación de usabilidad son la práctica más conocida. Consiste en mostrar al usuario una serie de tareas a realizar, y pedirle que las realice con el prototipo del sistema. Las acciones y comentarios de usuario se recopilan para un análisis posterior [25].

Existen diversos métodos para evaluar la usabilidad y se dividen en tres clasificaciones, a saber: inspección indagación y test [26]:

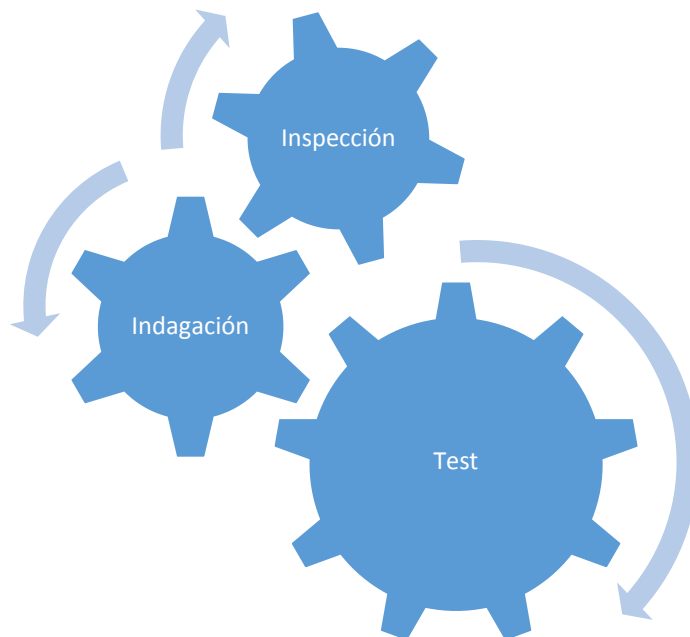


Figura 2: Clasificación de los métodos de evaluación de la usabilidad.

2.4.1. Inspección

Es el nombre dado a un conjunto de métodos basado en juicios y opiniones de expertos. Esta categoría se fundamenta principalmente en la interfaz como factor fundamental de la evaluación de usabilidad [26].

Los métodos más importantes de inspección son:

- Evaluación heurística.
- Recorrido de la usabilidad plural.
- Recorridos cognitivos.

- Inspección de estándares.

2.4.2. Indagación

Estos métodos están focalizado en las preferencias y necesidades del usuario. Es importante para estos métodos captar la mayor cantidad de información acerca de la usabilidad que permita desarrollar un software de calidad. Se realiza por medio de encuestas, hablando con los usuarios o mediante la observación de los mismos [26].

Los métodos más importantes de indagación son:

- Observación de campo.
- Grupos de discusión dirigidos (*Focus groups*).
- Entrevistas.
- Grabación del uso (*Logging*).
- Estudio de campo proactivo.
- Cuestionarios.

2.4.3. Test

Son los métodos en los que generalmente el usuario final realiza tareas específicas en el software, y los evaluadores analizan los resultados para ver como el sistema soporta a los usuarios con sus tareas [26].

Tipos de métodos:

- Medida de prestaciones.
- Test remoto.
- Pensando en voz alta.
- Interacción constructiva.
- Test retrospectivo.
- Método del conductor.

2.4.4. Evaluación heurística de usabilidad

La evaluación heurística [27][28] es un método de evaluación de interfaces de usuario que permite hallar los problemas de usabilidad de un software. Su aplicación consiste en establecer un conjunto de evaluadores que inspeccionan la interfaz con respecto a un pequeño conjunto de principios de usabilidad bastante amplios, que se conocen como la "heurística". El conjunto original de heurísticas de usabilidad utilizados por varios de los primeros estudios fue desarrollado con el propósito de hacer el método fácil de explicar y entender [29], debido a que es un aspecto importante de la ingeniería de usabilidad que los métodos pueden ser ampliamente utilizados y ser fáciles de transferir a otras organizaciones [30].

Así mismo, Nielsen [30] propuso siete factores para contabilizar la usabilidad de un producto de software. De estos factores se desglosan características que ayudan a evaluar la usabilidad de un producto de software (más información ver apéndice A):

- Factor 1: Visibilidad del estado del sistema.
- Factor 2: Relación entre sistema y mundo real.
- Factor 3: Control de usuario y libertad.
- Factor 4: Consistencia y estándares.
- Factor 5: Prevención de errores.
- Factor 6: Reconocimiento en lugar de recuerdo.
- Factor 7: Flexibilidad y eficiencia de uso.

2.4.4.1. Diseño de la prueba

La evaluación heurística está fundamentada en una serie de normas que describen el dominio en las interfaces, denominadas heurísticas. Estas heurísticas tienen como características que deben ser reconocidas en el campo de investigación o consideradas por el grupo de evaluadores que pueden ser importantes para cualquier elemento específico de la interfaz. Para Alva [31] cada heurística es presentada de una manera estructurada, con uno o más de los siguientes elementos:

- Preguntas de conformidad: que debe hacer el sistema/usuario para satisfacer las heurísticas.
- Evidencia de conformidad: que aspectos de diseño deben considerarse, que indiquen satisfacción o infracción de la heurística.
- Motivación: captura aspectos no conformes a las heurísticas (defectos) en un informe donde los evaluadores describen el problema, su severidad, y sugieren como arreglarlo.

2.4.4.2. Cantidad de evaluadores

Con relación a la cantidad de expertos en usabilidad evaluadores existen diferentes alternativas. Nielsen señala que el número de evaluadores debe estar entre tres y cinco [4], fundamentando que una mayor cantidad de evaluadores reduce el beneficio drásticamente y que la proporción de rentabilidad es más alta cuando se emplean tres o cuatro evaluadores. No obstante, diferentes autores como [32] cuestionan esta propuesta, señalando que esto funciona cuando se aplica la evaluación en condiciones ideales, pero en casos como evaluaciones Web no responde a las necesidades de evaluación.

2.4.4.3. Ventajas y desventajas de la evaluación heurística

Ventajas



- Fácil de conducir.
- Útil en las etapas iniciales de desarrollo para obtener referencia informal, de la percepción de las actitudes del usuario hacia el sistema.
- Los evaluadores no necesitan ser miembros del grupo de usuarios de la aplicación.
- Identifica las limitaciones de la interfaz de usuario-Rápida y económica

Desventajas



- Sólo colecciona datos de un solo paso, por lo que algunos datos importantes pueden ser perdidos.
- No hay datos visuales permanentes.
- Sólo identifican las heurísticas infringidas.

Figura 3: Ventajas y desventajas de la evaluación heurística [31]

2.4.5. Escala de Usabilidad del Sistema (System Usability Scale, SUS).

La Escala de Usabilidad del Sistema (SUS) fue desarrollada por Brooke en 1996, debido a una fuerte necesidad en la comunidad de usabilidad para obtener una herramienta que pueda recolectar rápida y fácilmente la calificación subjetiva de un usuario respecto con la facilidad de uso de un producto.

El método SUS se usa usualmente después de que un usuario ha tenido la oportunidad de utilizar un sistema. La escala SUS es una escala de estilo Likert (generando puntuaciones de 1 a 5) que produce un único número, constituyendo una medida compuesta de la usabilidad del sistema global sometido a estudio. Está compuesto por 10 preguntas [33].

Luego de que el usuario haya respondido las preguntas, se realiza una sumatoria (diferente en cada pregunta). Se multiplica por 2.5 para obtener la escala de usabilidad del sistema [33].

2.5. Interacción persona- ordenador

La Association for Computer Machinery (ACM), define la IPO como “*la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado*” [26]. En el contexto español se denomina Interacción persona-ordenador y en el latinoamericano se reemplaza la palabra ordenador por computador.

Según [34] para desarrollar sistemas que propendan por una adecuada interacción, se hace necesario:

- 1) Comprender los factores tales como psicológicos, ergonómicos, organizativos y sociales, que determinan como la gente trabaja y hace uso de los computadores.
- 2) Desarrollar herramientas y técnicas que ayuden a los diseñadores a conseguir que los sistemas de información sean los eficaces según las actividades a las cuales se quieran aplicar.
- 3) Conseguir una interacción eficiente, efectiva y segura, tanto a nivel individual como de grupo.

2.6. Enfermedades Neuromotoras

Las enfermedades neuromotoras (Motor Neuron Disease, MND por sus siglas en inglés) se refieren al conjunto de desórdenes progresivos resultante de la degeneración de las células motoras de las partes que componen la motricidad humana. Estas células controlan la actividad muscular voluntaria como hablar, caminar, respirar y tragar. [35].

Uno de cada 50.000 personas padece este tipo de enfermedades. Es claro además que el riesgo de padecerlas aumenta con la edad y la prevalencia de las MND es casi del doble en hombres que mujeres [36].

El sitio principal de degeneración de la neurona motora clasifica a los trastornos.

Las MND más comunes son:

2.6.1. Esclerosis lateral amiotrófica

Comúnmente conocida como ELA (Amyotrophic Lateral Sclerosis, ALS), Es una enfermedad que progresa a gran velocidad. Ataca las neuronas, que son los agentes que controlan los músculos voluntarios que podemos controlar (músculos de los brazos, piernas, y cara). Al final de la enfermedad la persona pierde la capacidad de controlar la capacidad de respirar y requiere asistencia mecánica. La muerte sobre llega generalmente de tres a cinco años desde el inicio de la enfermedad, la mayoría por insuficiencia respiratoria [37].

2.6.2. Parálisis Bulbar Progresiva

Afecta el tallo cerebral, es la zona que interviene en las neuronas motoras inferiores necesarias para tragar, hablar, masticar y otras funciones. Los síntomas incluyen debilidad muscular faríngea, músculos mandibulares y faciales débiles, pérdida progresiva del habla, y atrofia muscular lingual. La debilidad en miembros superiores existe pero no es predominante. Al final, las personas que la padecen se vuelven incapaces de tragar y hablar, se encuentran en grave riesgo de asfixia y neumonía por aspiración.

2.6.3. Distrofia muscular

Existe un grupo de enfermedades denominados distrofia muscular. Es de curso progresivo, más lento que la ELA, las cuales tienen como característica principal la debilidad de los músculos estriados, encargados de la función de generar los movimientos voluntarios. Además, esta enfermedad es de origen genético, es transmitida por el cromosoma X, es decir, solo afecta a los hombres, mientras las mujeres conservan el gen de forma recesiva [38].

2.6.4. Atrofia muscular espinal

Como la mayoría de las MND, actúa de forma progresiva en las neuronas encargadas del movimiento. Degeneran las células del asta anterior de la medula, afectando las funciones motoras de la parte inferior y provoca debilidad y el consumo de los músculos esqueléticos. En la mayoría de los casos la debilidad es mayor en las piernas que en los brazos [39].

2.7. Método Delphi

El método Delphi fue desarrollado a mediados del siglo XX por RAND Corporation. Fue hecho para obtener un consenso de un grupo de expertos sobre un tema en particular [40]. Otros investigadores desarrollaron variaciones en el método. Linstone y Turoff [41] lo definen como: un método para la estructuración de un proceso de comunicación de grupo de manera que el proceso es eficaz en permitir que un grupo de individuos, como un todo, hagan frente a un problema complejo. Para llevar a cabo esta comunicación estructurada, se proporciona: una retroalimentación de las contribuciones individuales de la información y el conocimiento; alguna valoración de la sentencia de grupo; alguna oportunidad a las personas para revisar puntos de vista; y cierto grado de anonimato de las respuestas individuales.

El método Delphi es usado en investigaciones que requieren información de opinión [40]. En su estado sin variaciones, normalmente se realiza por medio de encuestas, donde se garantiza el anonimato de los participantes.

Para realizar la aplicación y posterior análisis del método Delphi, Astigarraga [42], propone cuatro etapas:

- Formulación del problema.
- Elección de expertos.
- Elaboración y lanzamiento de cuestionario.
- Desarrollo práctico y explotación de resultados.

En la Figura 4 se muestra el proceso del método Delphi, descrito por [41].

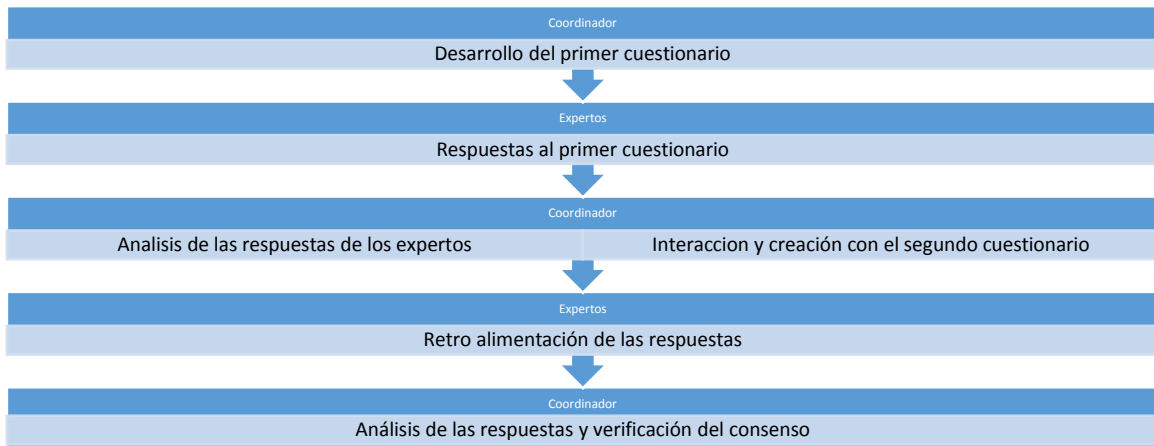


Figura 4: Proceso del Método Delphi

3. Antecedentes

Este capítulo está motivado por la necesidad de analizar y discutir el trabajo realizado sobre la evaluación de software. La primera parte “Diseño de software para personas en condición de discapacidad” es importante debido a que se introduce los principios de la interacción persona-ordenador en el caso de personas con enfermedades que alteran la motricidad.

La segunda parte recibe como título “Adaptabilidad de la evaluación heurística a casos particulares”, en la cual se exponen casos donde se ha modificado el método para evaluar prototipos de software en contextos específicos.

Y por último, la tercera parte “Evaluación de productos de software para personas con discapacidad motora”, donde se explora los diferentes métodos usados hasta ahora para este tipo de valoración.

3.1. Diseño de software para personas en condición de discapacidad

Dado que las tecnologías electrónicas se integran más a la educación y al empleo, la capacidad de acceder y usar estas nuevas tecnologías se convierte en una herramienta fundamental para las personas con discapacidad logrando ser capaces de participar en estos ambientes [43]. El diseño para las personas con discapacidad se está convirtiendo en un tema cada vez más importante por una variedad de razones, sobre todo debido a la reciente legislación en muchos países que promueve los derechos de las personas con discapacidad que obliga a muchas empresas a desarrollar sitios web accesibles. Muchos desarrolladores se preocupan por la necesidad de producir un sistema que será usable para todos los usuarios, independiente de su capacidad y pérdida gradual de movilidad. Además, se inquietan debido al hecho de que tengan que comprometer seriamente su diseño general para lograr este objetivo. Es evidente que esto no estaría en el interés de

nadie, y con la creciente capacidad de personalizar las interfaces para satisfacer las necesidades de los diferentes usuarios, esto no es necesario [44].

Existen numerosos conjuntos de pautas para ayudar a los desarrolladores a producir sistemas que sean accesibles y utilizables por personas con discapacidad. Estos incluyen directrices muy generales, como los producidos por el Centro de Diseño Universal [45], herramienta de autor [46] y los desarrolladores de contenido [47]. Sin embargo, no está claro si proporcionar pautas es un método eficaz para garantizar diseños utilizables.

Para Helen et al, [48], para tratar con diseños para las personas con discapacidad, los desarrolladores necesitan tener un marco conceptual en el que deben situar las directrices relacionadas con la discapacidad, que a menudo no tienen debido a la falta de experiencia con personas con discapacidad y sus tecnologías.

Uno de los muchos desafíos de diseño para las personas con discapacidad es el hecho de que hay muchas posibilidades de padecer discapacidades, de los sistemas sensoriales, físicas y cognitivas, y que éstos pueden ocurrir en las combinaciones, en vez de individualmente. Esto es particularmente importante ya que a medida de que la edad avanza, es probable que las personas adquieran múltiples discapacidades. Y aunque una discapacidad por separado puede tener un efecto relativamente menor, sus efectos combinados pueden ser importantes.

3.2. Adaptabilidad de la evaluación heurística a casos particulares

La evaluación heurística tiene el potencial de ser una herramienta de evaluación útil para examinar los prototipos, debido a que no hace suposiciones sobre las tareas y el propósito del software. La evaluación heurística es una técnica de inspección que le da la libertad a los evaluadores en la forma en que llevan a cabo la evaluación [4].

Jakob Nielsen desarrolló factores heurísticos basados en aplicaciones de escritorio [30]; estos se refieren a los conceptos de interfaz de usuarios comunes. Tales como deshacer, rehacer y prevención de errores. Sin embargo, muchas de estas ideas dan un significado limitado en el contexto de los software que facilitan el acceso de

personas con enfermedades que limitan su motricidad a la tecnología. Si bien pueden ser útiles en el descubrimiento de algunos problemas de usabilidad en estos tipos de productos, no abordan varios problemas de usabilidad importantes, tales como el requerimiento de adquirir un hardware especial para su uso o que el software posea el principio de acción mínima.

Dykstra [49], sugiere que las heurísticas pueden ser desarrolladas para categorías específicas de software mediante la evaluación de los productos existentes, y mediante el desarrollo de principios que describen los problemas de usabilidad encontrados. Sin embargo, dada la gran variabilidad de problemas de usabilidad que esperaba hallar en este tipo de software la realización de las evaluaciones hechas por nosotros mismos no era el práctico. Esperaba hallar una variación significativa en problemas de usabilidad encontrados en las diferentes gamas de este tipo de software. Por lo tanto, necesitaba una amplia cobertura del espacio de diseño, y que teníamos que incluir un gran número de software.

La evaluación heurística puede ser adaptada para que pueda ser usado para encontrar problemas de usabilidad importantes en todo tipo de prototipos [50]. Un ejemplo de esto es el trabajo realizado por Lynch [10] y Pinelle [50], en los cuales desarrollaron un conjunto de heurísticas específicas para la evaluación de diferentes productos de software. Lynch et al, propuso un medida heurística ponderada para la evaluación de páginas web por personas de edad avanzada. Creando 32 heurísticas basadas en criterios de expertos, e interrogando sobre la importancia de las características heurísticas relativa a los usuarios finales.

La importancia de este estudio radica en que hay varios modelos desarrollados para la evaluación de la usabilidad web [51]–[54]. Sin embargo, existe una necesidad creciente de un modelo que mide cuantitativamente la usabilidad de un sitio mientras que da un mayor nivel de importancia a ciertas heurísticas calificadas por el usuario final [10].

Pinelle [50], crearon un método heurístico para la evaluación de video juegos, obteniendo diez heurísticas. En este caso, los problemas de usabilidad se obtuvieron de una revista especializada en juegos, donde escriben profesionales en

esta área. Problemas de usabilidad, como el mapeo de control y la comprensibilidad de las representaciones visuales, también están cubiertos en la mayoría de opiniones y el análisis de los juegos permite identificar problemas de usabilidad comunes.

Para crear las reglas heurísticas siguieron el siguiente método:

1. Identificación de los problemas de usabilidad del mundo real que proporcionan amplitud y profundidad de cobertura de los juegos.
2. Desarrollo de un conjunto de categorías que agrupan similares problemas de usabilidad.
3. Creación de heurísticas en las categorías de problemas y que describan cómo los problemas comunes de usabilidad se pueden evitar.

La adaptabilidad de este método permite obtener una evaluación de usabilidad de todo tipo de software a un bajo costo.

3.3. Evaluación de productos de software para personas con discapacidad motora

Lo primero que cabe destacar es la ausencia de estudios elaborados en este contexto. No obstante, lo hecho por Rowan et al, [55], es de gran importancia. Ellos aseguran que ningún método puede transmitir satisfactoriamente a los desarrolladores todos los temas de accesibilidad y usabilidad. Por lo tanto consideran un número de métodos que podría ser utilizado para elevar los niveles de accesibilidad de los sitios. Es valioso desde el punto de vista que incorpora un metamodelo para la evaluación de sitios web para personas con discapacidad que proporcionaría una norma para descubrir la mayoría de los problemas de accesibilidad asociados a un sitio web. Concluyen que la accesibilidad de las personas con discapacidad a sitios web es un grave problema debido a que los proveedores de información no comprenden los requisitos de uso de estas personas.

El estudio elaborado por Mankoff et al [3], se refiere a simular la discapacidad usando la técnica denominada Evaluación de accesibilidad mediante simulación de la experiencia del usuario (EASE por su siglas en ingles). A primera vista esto resulta muy útil debido a que las pruebas de usuario con poblaciones especiales a menudo requieren un mayor esfuerzo, tiempo y compensaciones monetarias por parte de los diseñadores y desarrolladores que las pruebas de usuario con la población general. Además, puede ser difícil encontrar grupos homogéneos de participantes o para controlar cuidadosamente ciertos factores. No obstante, los síntomas que manifiestan las personas con deficiencias motoras bajas son extremadamente complejos, debido en gran parte a la variedad de formas en que esas discapacidades se usan para interactuar. Esta gran complejidad hace que sea difícil de controlar y evaluar el impacto de estas discapacidades en la experiencia de usar un computador.

Para An et al [56], el concepto de comunicación aumentativa y alternativa (AAC por sus siglas en inglés) puede compensar los problemas de comunicación relacionados con la discapacidad, contribuyendo así a una mayor calidad de vida y de independencia. Usando este concepto desarrollaron un teclado virtual que usa bioseñales. Lo importante de este trabajo es la primera prueba de usabilidad que se encontró en la revisión de literatura de esta tesis. Seis personas con diferente grado de pérdida de movilidad se usaron para realizar la evaluación, con cinco métricas: Apariencia del sistema, grado de satisfacción, grado de rendimiento, intención de compra y tiempo para realizar una actividad. El problema de esta prueba radica es que usa características subjetivas e incluye un parámetro que puede resultar comprometedor para el usuario final.

4. Solución propuesta

De acuerdo con los resultados obtenidos de la revisión de literatura en el capítulo de antecedentes, se identificaron las particularidades de los métodos usados en la evaluación heurística y se justificó la creación del método propuesto en esta tesis. En el capítulo actual se presenta el Método de evaluación de usabilidad aplicada a productos de software que facilitan el acceso a herramientas informáticas de personas con enfermedades que afectan la motricidad.

Previamente en esta tesis se han desarrollado un conjunto de heurísticas para completar dos roles importantes en el proceso de diseño de estos tipos de software. En primer lugar, las heurísticas que sirven como un grupo de nociones de diseño que se pueden utilizar durante las etapas formativas de diseño y desarrollo; y en segundo lugar, pueden ser utilizadas para llevar a cabo inspecciones de usabilidad donde los expertos las utilizan para examinar su diseño.

El método usado para la definición de heurísticas de usabilidad fue motivada inicialmente por el enfoque utilizado por Nielsen [30], cuando definió un conjunto de heurísticas de propósito general. Nielsen en 1994, evaluó varios conjuntos existentes de heurística para determinar cuáles proporcionaban la "*cobertura explicativa más amplia*" de un conjunto de 249 problemas de usabilidad. Sin embargo, ya que la literatura en el tema de la heurística en usabilidad para los aplicativos de software que facilitan el acceso al uso de los computadores a personas con enfermedades que alteran su motricidad es limitada, y debido a los resultados de los estudios de problemas de usabilidad de estos software; no se ha informado anteriormente en detalle, las consideraciones que fueron algo diferente.

El proceso de desarrollo de las heurísticas se elaboró en cuatro pasos. El primero de ellos es identificar los problemas de Usabilidad, este es el paso más importante de todo el proceso. Sin embargo, debido a la ausencia de literatura en este tema se consultó a expertos en usabilidad y accesibilidad. El segundo paso es crear las

reglas heurísticas de acuerdo a la información obtenida en el paso anterior. El tercero es la asignación de pesos para cada heurística. Para probar el método se evaluaron diez aplicativos de software de distribución libre. El cuarto paso, es la evaluación por dos expertos de las heurísticas presentes en estos tipos de software. Pero antes de explicar los pasos se hace necesario explicar la elección de la evaluación heurística.

4.1. ¿Por qué es la evaluación heurística el mejor método para examinar software que facilitan el acceso a la computación de personas con enfermedades que alteran la motricidad?

La evaluación heurística fue creada alrededor de 1990, en un momento en que era costoso conseguir el acceso a los usuarios. Era común que las personas tenían que ser entrenadas para usar la tecnología que está siendo evaluada antes de que pudieran sentarse en un laboratorio de usabilidad para realizar algunas tareas. Todo el concepto de ni siquiera tener una interfaz para los usuarios finales era bastante nuevo.

Las pruebas de usabilidad han existido por lo menos desde la década de 1980, pero comenzó a ser ampliamente practicado casi al mismo tiempo que Nielsen y Molich publicaron su método de evaluación heurística [27] [57]. Mientras que las pruebas de usabilidad, probablemente necesitan una renovación como método, el proceso básico sigue funcionando bien. Es bastante barato para tener acceso a los usuarios. Las interfaces de usuario para el uso de la tecnología están en todas partes. Para la mayoría de las aplicaciones de la tecnología que prueban, los usuarios no necesitan un entrenamiento especial.

En el caso específico de evaluar la usabilidad de un software determinado con personas con problemas de motricidad resulta en gran medida costoso debido a que el equipo encargado de la evaluación debe asumir los costos asociados al desplazamiento de estas personas al laboratorio de usabilidad. Un estudio realizado por Petrie propone solventar por medio de la evaluación remota este problema. Sin embargo presenta otras dificultades relacionadas con la interacción de la personas con tecnología de asistencia [48].

Aun así, la evaluación heurística usada en esta tesis usa un método pesos relativos asignados por los usuarios generando una disyuntiva entre el juicio de los expertos y lo que los usuarios finales consideran más importante.

La evaluación heurística puede ayudar a un equipo cuando la interfaz del usuario cumple con las directrices de otra persona. Pero la observación de las personas que utilizan un diseño en un test de usabilidad da un dato al equipo principal para la toma de decisiones en el diseño, para sus usuarios utilizando su diseño; sobre todo en un mundo evolucionado mucho más allá de la entrada de línea de comandos e interfaces de usuarios simples para opciones como pantallas táctiles, medios sociales y la conectividad ubicua. Por separado y en combinación, estas y otras decisiones de diseño presentan complejos problemas sutiles, de usabilidad. La observación de las personas que utilizan un diseño siempre triunfará sobre una inspección o auditoría para obtener evidencia sólida que determina una dirección de diseño.

4.2. Adquisición de características heurísticas

El paso inicial para el desarrollo del nuevo método de evaluación fue definir los parámetros para ser usados como heurísticas. Como se anotó anteriormente, no existen trabajos relacionados con este tipo de investigación. Por esta razón, se hizo necesario realizar un estudio Delphi con leves modificaciones. Los expertos en usabilidad fueron seleccionados por medio de la herramienta web Scopus, según el número de publicaciones realizadas sobre el tema. Fueron contactados vía correo electrónico, debido a que la gran mayoría de estos están ubicados en la península ibérica.

Para obtener las características heurísticas se les pidió responder una pequeña encuesta y además, dada su experiencia, se les solicitó que anotaran tres particularidades que consideran importantes en el diseño de estos tipos de aplicativos de software. Con esto se obtuvo una lista de trece especificaciones. Luego de recolectar este cuestionario y concatenarlo, se procedió a enviar una segunda encuesta para determinar los pesos que ellos les daban a las

características, excluyendo las dadas por el experto al que se le enviaba el cuestionario (brindando una calificación de 1 a 10).

Las características fueron agrupadas en cuatro grupos propuestos por diversos autores [58]–[61]. Estos son:



Figura 5: Categorías de Usabilidad

En la tabla 1, se muestra un formato en blanco de la evaluación heurística con pesos y presencia. Los pesos de cada heurística son dados en el siguiente capítulo y el concepto de presencia se define en el capítulo 4.4.

Tabla 1: Evaluación heurística

| Eficiencia | ¿Presencia? | Peso | Puntuación |
|--|-------------|------|------------|
| Principio de acción mínima | | | |
| Eficiencia al llevar a cabo una tarea específica | | | |
| Prevención de posibles errores cuando se ejecuta una tarea | | | |
| Tiempo de respuesta entre acción y respuesta corto | | | |
| Puntuación total de eficiencia | | | |

| Eficacia | ¿Presencia? | Peso | Puntuación |
|---|-------------|------|------------|
| Esfuerzo mínimo para realizar una tarea | | | |
| Puntuación total de eficacia | | | |

| Aprendibilidad | ¿Presencia? | Peso | Puntuación |
|--|-------------|------|------------|
| Ayudas mientras el usuario se entrena | | | |
| Proporcionar buena información de orientación (para que el usuario sepa que pasa dentro de un proceso) | | | |
| Tiempo de entrenamiento breve | | | |
| Puntuación total de aprendibilidad | | | |

| Satisfacción | ¿Presencia? | Peso | Puntuación |
|--|-------------|------|------------|
| Adaptabilidad a los usuarios | | | |
| Autonomía en las tareas hechas a cabo | | | |
| Capacidad del software que posibilita al usuario predecir el resultado de sus acciones de forma intuitiva. | | | |
| Affordance | | | |
| No requerir de aditamento adicional para ejecutar el software | | | |
| Puntuación total de satisfacción | | | |

A continuación se describen las heurísticas que pueden resultar confusas:

4.2.1 Principio de acción mínima. El concepto de principio de acción mínimo se deriva del principio seis: bajo esfuerzo físico propuesto en “*Maximizing usability: The principles of Universal Design*” [62]. Este principio define cuatro directrices:

- a. Permitir al usuario mantener una posición corporal neutra.
- b. Utilizar fuerzas de funcionamiento razonables.
- c. Minimizar las acciones repetitivas.
- d. Minimizar el esfuerzo físico sostenido.

Las directrices c y d se pueden abstraer como “Principio de acción mínima”.

4.2.2 Prevención de posibles errores cuando se ejecuta una tarea. El estándar ISO 25010:2011 definió este término como el “*grado en que un sistema protege a los usuarios contra cometer errores*” [21].

4.2.3 Affordance Si bien no existe una palabra específica en español para Affordance, Donald Norman [63] usó el término en el contexto de la IPO para representar las posibilidades de acción que son inmediatamente percibidas por el usuario.

En su libro “*The Design of Everyday Things*”, Norman establece el concepto Affordances no solo dentro de las capacidades físicas del usuario, sino también en la capacidad de éste de nutrirse de experiencias pasadas, metas, planes, estimaciones, comparando otro tipo de vivencias, entre otros.

Norman sustenta el término mediante un ejemplo: si una persona entra a una habitación en la que hay una pelota y un sillón; la persona podría lanzar el sillón y sentarse sobre la pelota porque es objetivamente posible.

Para Norman, la Affordance entre objetos y la persona que interactúa con ellos hace muy probable que el sujeto se siente en el sillón y lance la pelota. Esta persona acumula experiencias, como la forma de los objetos, textura, entre otros. La pelota tiene unas características perfectas para ser lanzada y el sujeto lo ha visto hacer en otras ocasiones

4.3. Asignación de pesos a las heurísticas

Con el fin de decidir el valor de los pesos para cada heurística, se realizó una encuesta a diez personas con enfermedades que afectan la motricidad. La encuesta se distribuyó de manera presencial. Las personas participaron bajo consentimiento informado y no se les pagaba para la realización de dicha encuesta. La encuesta enumeraba las trece heurísticas. Se les pidió a los sujetos para valorar la importancia de cada elemento para aumentar la usabilidad de los aplicativos de software en una escala de 1 (no importante) a 10 (muy importante). El valor de respuesta medio fue seleccionado como el peso de la heurística correspondiente. Este peso se aplicaría en el cálculo de una puntuación final para la facilidad de uso del software.

Para que los usuarios comprendan de una forma adecuada las heurísticas sabiendo que ningún usuario posee conocimiento técnico relacionado con el uso del computador, se elaboró un cuestionario con un nivel técnico más cercano al usuario diario:

Tabla 2/parte 1: Encuesta de asignación de pesos heurísticos

| | Nada importante | | | | | | | | | | Muy importante | | | | | | | | | |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1. ¿Considera usted importante que el software no requiera una cantidad excesiva de movimientos suyos para ser ejecutado.? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. ¿La rapidez y la facilidad es un atributo importante para usted? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. ¿Que el software detecte y corrija posibles errores es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Un tiempo corto entre su acción y la respuesta del pc, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Un mínimo esfuerzo para hacer funcionar el software, ¿es importante para usted? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Tabla 2/parte 2: Encuesta de asignación de pesos heurísticos

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 6. ¿Considera usted importante que el software proporcione ayudas visuales mientras se entrena? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. ¿Considera usted importante que el software proporcione en todo momento información del estado del sistema? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. Un corto tiempo de entrenamiento, ¿es importante para usted? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. Que otros usuarios, con diferentes tipos de enfermedades puedan hacer uso del software, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. El software brinde autonomía total desde el inicio hasta apagar el equipo, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 11. Que el software funcione de forma intuitiva, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 12. Que el usuario perciba las posibles acciones que ofrece el software sean notadas por usted de forma inmediata, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 13. Que no sea necesario añadir un hardware adicional para que el software funcione, ¿es importante? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

4.3.1. Marco legal para la realización de la prueba

Como lo sugiere Lynch en su estudio. Antes de iniciar la prueba se les solicito a los usuarios que leyeran y firmaran un documento denominado “*Consentimiento informado*” donde se explica la finalidad del estudio, se garantiza además el anonimato de las personas involucradas en él y se expresa que se hace de forma voluntaria. Si las condiciones físicas del usuario no le permitían firmar el consentimiento a la persona que lo acompañaba, previa aceptación por parte de la persona afectada.

A continuación se presenta el texto usado como consentimiento voluntario:

“Consentimiento Informado para la realización de la Prueba de Usabilidad en personas con necesidades especiales

El propósito de esta ficha de consentimiento es proveer a los participantes en esta investigación con una clara explicación de la naturaleza de la misma, así como de su rol en ella como participantes.

La presente investigación es conducida por Jhon Fernando Sánchez Álvarez de la Universidad Nacional de Colombia Sede Medellín

La meta de este estudio es establecer un método para la evaluación de software diseñados exclusivamente para personas con enfermedades que alteran su motricidad.

Las pruebas que se realizarán se detallan a continuación:

- Cuestionario sobre heurísticas.
- Pruebas con dos software (tiempo estimado 30-40 minutos)

La participación en este estudio es estrictamente voluntaria. La información que se recoja será confidencial y no se usará para ningún otro propósito fuera de los objetivos de esta investigación. Los valores obtenidos en las diferentes pruebas serán codificados usando un número de identificación y por lo tanto, serán anónimas.

La realización de las pruebas puede provocarle fatiga general y otros síntomas provocados por el esfuerzo en realizar las pruebas de software.

Si tiene alguna duda sobre este proyecto, puede hacer preguntas en cualquier momento durante su participación en él. Igualmente, puede retirarse del proyecto en cualquier momento sin que eso lo perjudique de ninguna forma. Si nota incomodidad, cansancio, fatiga o cualquier otro tipo de molestia durante la realización de alguna de las pruebas, tiene usted el derecho de hacérselo saber al investigador o de no realizarlas.

Desde ya le agradecemos su participación.

Acepto participar voluntariamente en esta investigación, conducida por Jhon Fernando Sánchez Álvarez. He sido informado de la meta de este estudio es crear un método para la evaluación de software diseñados exclusivamente para personas con enfermedades que alteran su motricidad.

Reconozco que la información que yo provea en el curso de esta investigación es estrictamente confidencial y no será usada para ningún otro propósito fuera de los de este estudio sin mi consentimiento.

He sido informado de que puedo hacer preguntas sobre el proyecto en cualquier momento y que puedo retirarme del mismo cuando así lo decida, sin que esto acarree perjuicio alguno para mi persona. De tener preguntas sobre mi participación en este estudio, puedo contactar al profesor Jovani Alberto Jiménez Builes, Ph. D. al teléfono 4255222.

Entiendo que una copia de esta ficha de consentimiento me será entregada, y que puedo pedir información sobre los resultados de este estudio cuando éste haya concluido. Para esto, puedo contactar al profesor Jovani Alberto Jiménez Builes, Ph. D. al teléfono anteriormente mencionado”

Firma

4.3.2. Cantidad de usuarios requeridos para la prueba

Nielsen [64] desarrolló un modelo matemático donde mostraba el porcentaje de problemas encontrados en una evaluación de usabilidad con n usuarios:

$$N(1 - (1 - L)^n)$$

Ecuación 1: Cantidad de problemas encontrados en una evaluación de usabilidad con n usuarios

Donde N es el número total de problemas de usabilidad en el diseño. Mientras que L es la proporción de problemas encontrados durante la evaluación por un usuario. Generalmente se establece L al 31%.

El grafico resultante con L al 31% es:



Grafico 1: Ecuación 1 graficada.

Nielsen dice “*add more and more users, you learn less and less*” debido a que se encontraran los mismos problemas una y otra vez, y no es obligatorio mantener la observación de las mismas cosas. Asegura también que después del **quinto usuario** no se encontrara nada nuevo [65].

Sin embargo, un estudio encontró que en ocasiones cinco usuarios no son suficientes; para sustentar esto, se realizó un experimento con 60 usuarios los cuales se distribuyeron al azar en grupos de 5 personas. Algunos de estos grupos encontraron el 99% de los problemas, mientras otros encontraron 55%. Aumentando el número de 5 a 10, el porcentaje más bajo de problemas encontrados fue de 80% y duplicando el número se obtiene un 95% de problemas encontrado [66].

El problema con la aseveración de Nielsen es que define los cinco usuarios si son homogéneos. En su estudio Nielsen dice que si se da el caso de ser heterogéneos se tendrá que poner más usuarios a realizar pruebas.

Sin embargo, todas estas discusiones quedan abducidas debido a la imposibilidad de conseguir usuarios que presenten enfermedades que alteren su motricidad y posean el conocimiento básico para realizar la encuesta, y las tareas propuestas en el computador es realmente reducido. Por medio del Doctor Carlos Jaime Yepes Álvarez, neurocirujano de una clínica en Medellín, se logró reunir un grupo de diez personas todas con estudios de media técnica o académica.

Tabla 3: Usuarios con enfermedades neurológicas que alteran la motricidad.

| Usuario | Enfermedad |
|------------|--------------------------------|
| Usuario 1 | Distrofia muscular progresiva |
| Usuario 2 | Distrofia muscular progresiva |
| Usuario 3 | Distrofia muscular progresiva |
| Usuario 4 | Distrofia muscular progresiva |
| Usuario 5 | Esclerosis lateral amiotrofica |
| Usuario 6 | Esclerosis lateral amiotrofica |
| Usuario 7 | Esclerosis lateral primaria |
| Usuario 8 | Esclerosis lateral primaria |
| Usuario 9 | Esclerosis lateral primaria |
| Usuario 10 | Parálisis bulbar progresiva |

Es importante anotar que todas las enfermedades son del sistema nervioso central, cada usuario posee una habilidad motriz. Incluso los dos usuarios que poseen atrofia muscular progresiva tienen diferente grado de pérdida de motricidad, esto es debido a que es una enfermedad progresiva; es decir, de acuerdo que pasa el tiempo los síntomas se acentúan de forma más notoria.

4.3.3. Nivel educativo de los usuarios

El nivel de educación se refiere al último grado que obtuvo el usuario. Los 10 usuarios poseen educación media académica o técnica. Que se define como:

Según la Ley 115 de 1994 [67], la educación media compone la culminación, afianzamiento y progreso en el logro de los grados anteriores y comprende dos

grados, el décimo y el undécimo 11. El propósito es la adquisición de las ideas y los valores universales y la preparación para el ingreso de los estudiantes a la Educación Superior y al trabajo.

La educación media tendrá el carácter de académica o técnica. Al finalizar se obtiene el título de bachiller que faculta al estudiante para ingresar a la educación superior en cualquiera de sus niveles y carreras.

4.3.3.1. Educación Media Académica:

La educación media académica permite al estudiante, según sus intereses y capacidades, profundizar en un campo específico de las ciencias, las artes o las humanidades y acceder a la educación superior.

Para el logro de los objetivos de la educación media académica son obligatorias y fundamentales las mismas áreas de la educación básica en un nivel más avanzado, además de las ciencias económicas, políticas y la filosofía.

Aunque todas las áreas de la educación media académica son obligatorias y fundamentales, las instituciones educativas organizarán la programación de tal manera que los estudiantes puedan intensificar, entre otros, en ciencias naturales, ciencias sociales, humanidades, arte o lenguas extranjeras, de acuerdo con su vocación e intereses, como orientación a la carrera que vayan a escoger en la educación superior [67].

4.3.3.2. Educación Media Técnica:

La educación media Técnica prepara a los estudiantes para el desempeño laboral en uno de los sectores de la producción y de los servicios, y para la continuación en la educación superior.

Estará dirigida a la formación calificada en especialidades tales como: agropecuaria, comercio, finanzas, administración, ecología, medio ambiente, industria, informática, minería, salud, recreación, turismo, deporte y las demás que requiera el sector productivo y de servicios. Debe incorporar, en su formación teórica y práctica, lo más avanzado de la ciencia y de la técnica, para que el estudiante esté en capacidad de adaptarse a las nuevas tecnologías y al avance de la ciencia.

Las especialidades que ofrezcan los distintos establecimientos educativos, deben corresponder a las necesidades regionales. Los objetivos específicos de la educación media técnica son la capacitación básica inicial para el trabajo; la preparación para vincularse al sector productivo y a las posibilidades de formación que éste ofrece, y la formación adecuada a los objetivos de educación media académica, que permita al educando el ingreso a la educación superior [67].

No obstante, es de gran dificultad hallar personas que presenten enfermedades neurodegenerativas que tengan un nivel educativo alto. Si bien la Constitución Política Colombiana garantiza el acceso al sistema de educación de las personas que tienen “*una condición de vida más difícil*” [68], los datos estadísticos proporcionados por el DANE no reflejan un acceso adecuado a la educación superior de estas personas [69].

A continuación, se muestran los pesos promedios dados por los usuarios:

Tabla 4: Pesos dados por los usuarios.

| | Media | Moda | Min | Max |
|--|-------|------|-----|-----|
| Principio de acción mínima | 9,8 | 10 | 9 | 10 |
| Eficiencia al llevar a cabo una tarea específica | 6,2 | 7 | 4 | 8 |
| Prevención de errores cuando se ejecuta una tarea | 9,6 | 10 | 8 | 10 |
| Tiempo de respuesta entre acción y respuesta corto | 5,8 | 5 | 5 | 8 |
| Tiempo de entrenamiento breve | 5,6 | 6 | 4 | 7 |
| Esfuerzo mínimo para realizar una tarea | 9 | 9 | 8 | 10 |
| Ayudas mientras el usuario se entrena | 3,6 | 4 | 3 | 4 |
| Adaptabilidad a los usuarios | 8,4 | 8 | 7 | 10 |
| Guardar información de preferencias de los usuarios | 7,2 | 6 | 6 | 9 |
| Affordance | 5,2 | 4 | 4 | 7 |
| Capacidad del software que posibilita al usuario predecir el resultado de sus acciones de forma intuitiva. | 6 | 6 | 4 | 8 |
| Proporcionar buena información de orientación (para que el usuario sepa qué pasa está dentro de un proceso). | 5,6 | 6 | 4 | 7 |
| Autonomía en las tareas hechas a cabo | 9,4 | 10 | 8 | 10 |
| No requerir de ningún hardware adicional para ejecutar el software | 8,8 | 9 | 8 | 10 |

4.4. Evaluación por expertos del software

Previamente se dijo, que los expertos también asignaron pesos a las heurísticas propuestas por sus compañeros, esto tiene el objetivo contrastar los pesos con la de los usuarios:

Tabla 5: Pesos dados por los expertos.

| | Media |
|---|-------|
| Principio de acción mínima | 8,7 |
| Eficiencia al llevar a cabo una tarea específica | 7,7 |
| Prevención de errores cuando se ejecuta una tarea | 9,0 |
| Tiempo de respuesta entre acción y respuesta corto | 6,0 |
| Tiempo de entrenamiento breve | 5,7 |
| Esfuerzo mínimo para realizar una tarea | 8,3 |
| Ayudas mientras el usuario se entrena | 6,0 |
| Adaptabilidad a los usuarios | 7,7 |
| Guardar información de preferencias de los usuarios | 7,3 |
| Affordance | 7,3 |
| Capacidad del software que posibilita al usuario predecir el resultado de sus acciones de forma intuitiva. | 6,5 |
| Proporcionar buena información de orientación (para que el usuario sepa en qué paso está dentro de un proceso). | 7,0 |
| Autonomía en las tareas hechas a cabo | 7,5 |
| No requerir de ningún hardware adicional para ejecutar el software | 8 |

Por medio de un cálculo de la varianza entre los pesos heurísticos asignados por usuarios y expertos, se tienen los siguientes resultados:

Tabla 6: Varianza en pesos heurísticos

| | Varianza |
|---|----------|
| Principio de acción mínima | 0,321111 |
| Eficiencia al llevar a cabo una tarea específica | 0,537778 |
| Prevención de errores cuando se ejecuta una tarea | 0,09 |
| Tiempo de respuesta entre acción y respuesta corto | 0,01 |
| Tiempo de entrenamiento breve | 0,001111 |
| Esfuerzo mínimo para realizar una tarea | 0,111111 |
| Ayudas mientras el usuario se entrena | 1,44 |
| Adaptabilidad a los usuarios | 0,134444 |
| Guardar información de preferencias de los usuarios | 0,004444 |
| Affordance | 1,137778 |
| Capacidad del software que posibilita al usuario predecir el resultado de sus acciones de forma intuitiva. | 0,0625 |
| Proporcionar buena información de orientación (para que el usuario sepa en qué paso está dentro de un proceso). | 0,49 |
| Autonomía en las tareas hechas a cabo | 0,9025 |
| No requerir de ningún hardware adicional para ejecutar el software | 0,16 |

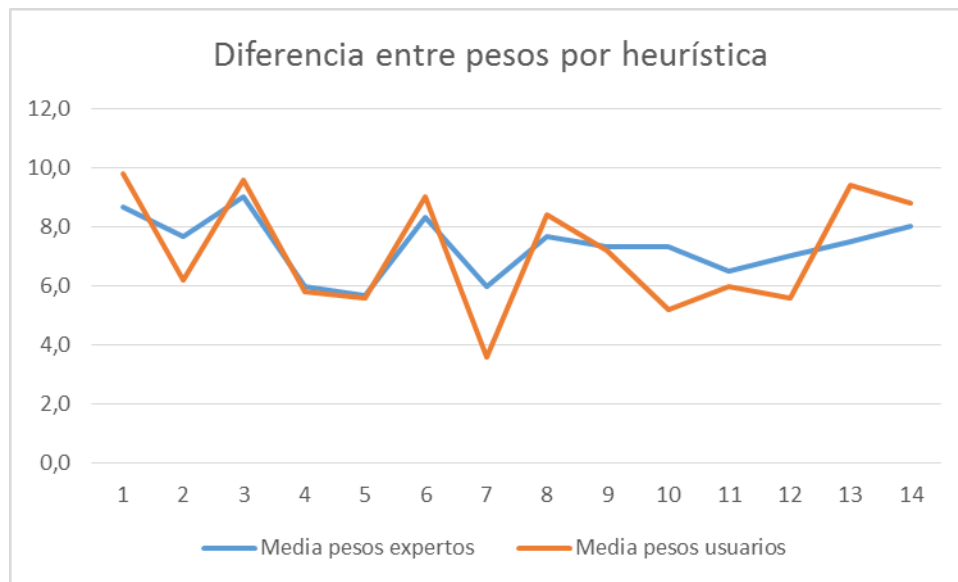


Figura 6: Diferencia entre pesos por heurística

De acuerdo a los datos obtenidos en la Figura 6 y Tabla 6 la varianza más grande observada hace referencia a la heurística “Ayudas mientras el usuario se entrena”, siendo la heurística peor calificada por los usuarios finales. De igual manera, “Affordance” fue la segunda con más baja media. Lo curioso en este análisis viene

dato con la tercera heurística que posee la mayor varianza “*Autonomía en las tareas hechas a cabo*” la cual es una de las heurísticas con el peso más alto por parte de los usuarios y uno de los más bajos para los expertos. Es precisamente en este punto, donde la investigación realizada como parte de la tesis que cobra importancia. Los usuarios al ser los beneficiarios del software, poseen características que los expertos no comprenden [48].

Luego de la asignación de los pesos, cada uno de los diez software seleccionados fueron evaluados por dos expertos. Las heurísticas resultantes generan unos porcentajes de usabilidad en las cuatro clases (Eficiencia, eficacia, aprendibilidad y satisfacción), así como una puntuación total. El cálculo de cada puntuación siguió el siguiente método: tras inspeccionar el software, el evaluador decide si el software cumplió la heurística, si la posee en algunos lugares pero no en otros, o no siguió la pauta en absoluto. Esto se conoce como una "*puntuación de presencia*" y fue marcado 2, 1, o 0, respectivamente. La puntuación final se calcula multiplicando el peso y la puntuación de presencia de cada heurística y sumando el producto. No obstante, Lynch [10] usa el promedio aritmético aplicado para obtener la usabilidad total de las cinco aplicaciones en un método inadecuado en este caso. Debido a que asigna igual importancia a una característica de usabilidad que posee un ítem (eficacia) a otra que posee cuatro ítems (eficiencia). Para evitar esto se usó un promedio con pesos (ver ecuación 1)

$$\sum_{i=0}^m a_i * n_i = \frac{a_1 * n_1 + a_2 * n_2 \dots + a_m * n_m}{\sum n}$$

Ecuación 2: Promedio con pesos (siendo n la cantidad de heurística en cada característica).

La evaluación de la usabilidad heurística ponderada fue realizada por dos evaluadores entrenados, y el puntaje promedio fue tomado para crear un índice de capacidad de uso final. De los 10 software evaluados, las puntuaciones de usabilidad oscilaron entre 37% y 80%, con la puntuación media de 59%. Estos datos se muestran en la Tabla 7.

Tabla 7: Puntuación de usabilidad de los diez software

| | Eficiencia | Eficacia | Aprendibilidad | Satisfacción | Total |
|-------------|------------|----------|----------------|--------------|-------|
| Software 1 | 44% | 50% | 19% | 54% | 43% |
| Software 2 | 75% | 0% | 62% | 77% | 67% |
| Software 3 | 60% | 100% | 81% | 82% | 76% |
| Software 4 | 69% | 100% | 87% | 81% | 80% |
| Software 5 | 75% | 0% | 31% | 38% | 45% |
| Software 6 | 40% | 0% | 24% | 49% | 37% |
| Software 7 | 84% | 50% | 62% | 70% | 71% |
| Software 8 | 40% | 100% | 50% | 77% | 61% |
| Software 9 | 65% | 50% | 62% | 80% | 69% |
| Software 10 | 35% | 0% | 50% | 48% | 41% |

Se considera una usabilidad alta si el índice total se encuentra entre 80% a 100%. Una usabilidad media entre 70% a 80% y una usabilidad baja entre 0% y 70%. El total de usabilidad reportado en esta investigación es bastante notorio; siete de los diez software presentan un bajo índice de usabilidad.

5. Validación

En este capítulo se muestra la validación de la colección de nuevas heurísticas. Para lo anterior, se siguió el método propuesto en “*A methodology to establish usability heuristics*” [70] en donde se presentan las etapas para crear un nuevo conjunto de heurísticas. Es del interés de esta investigación en particular la etapa cinco, a saber:

Para probar el nuevo conjunto heurístico se debe realizar una comparación entre una heurística tradicional y la nueva, mediante evaluaciones de casos seleccionados y añadiendo nuevos casos de prueba con usuarios.

Para realizar el experimento, una aplicación fue evaluada por dos grupos de expertos, con experiencia similar e igualdad de condiciones. El grupo uno de expertos usó el conjunto de reglas heurísticas creadas. Mientras que el grupo dos utilizó las heurísticas de Nielsen (ver apéndice A). Los problemas de usabilidad encontrados en ambos grupos se comparan, por medio de tres categorías:

1. Problemas identificados por ambos grupos de evaluadores.
2. Problemas identificados sólo por el grupo que uso el nuevo conjunto de heurísticas.
3. Problemas identificados sólo por el grupo que utilizó heurísticas de Nielsen.

En nuestro caso particular, la característica dos identificó mayormente los problemas de usabilidad presentes en el software que facilita el acceso de personas con problemas de motricidad respecto a las características uno y tres. Esto radica en el hecho que las heurísticas de Nielsen fueron creadas como propósito general de software.

5.1. Usuarios

Diez usuarios (dos mujeres y ocho hombres) con edades entre los 18 y 30 años con enfermedades neurodegenerativas que presentan problemas notorios de movilidad, aceptaron acceder al estudio con previo consentimiento informado aceptado y se les proporcionó transporte hasta el lugar del experimento.

5.2. Localización

Todas las pruebas fueron realizadas en la Clínica Las Américas en Medellín usando un computador portátil Dell. Las métricas de usabilidad fueron tomadas por medio de grabación en video y posteriormente se contabilizó cada una.

5.3. Procedimiento

Para realizar un experimento empírico de usabilidad se eligieron dos software (los software identificados como *software 2* y *software 4* con usabilidad baja y alta respectivamente) de los diez software evaluados por el método creado en esta investigación.

La tarea que permitió capturar las métricas fue escribir un párrafo de 30 palabras usando cada software. Las métricas que se tuvieron en cuenta fueron tiempo total que tomo hacer la tarea y cantidad de errores cometidos. Los errores fueron definidos como el fallo a la hora de seleccionar la letra requerida para completar la tarea.

Después de finalizar la tarea, los usuarios completaron una encuesta de escala de usabilidad del sistema (SUS por sus siglas en inglés):

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:

1. Creo que me gustaría utilizar este sistema con frecuencia

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

2. He encontrado el sistema innecesariamente complejo

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

3. Pensé que el sistema era fácil de usar

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

4. Creo que necesito el apoyo de una persona para poder utilizar este sistema

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

5. Me pareció que las diversas funciones en este sistema fueron bien integradas

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

6. Pensé que era demasiado inconsistente este sistema

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

7. Imagino que la mayoría de la gente aprender a utilizar este sistema muy rápidamente

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

8. Me pareció que el sistema sea muy engorroso de usar

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

9. Me sentí muy seguro utilizando el sistema

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

10. Necesitaba aprender un montón de cosas antes de que pudiera usar este sistema

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

El cálculo de la escala de usabilidad se realiza siguiendo el método propuesto por Brooke [33]:

Las afirmaciones con número impar impacta de manera positiva, el procedimiento para calcular el aporte de cada una se hace de la siguiente manera:

Calificación obtenida – 1 (menos uno)

Las afirmaciones con número par impacta de manera negativa, el procedimiento para calcular el aporte de cada una se hace de la siguiente manera:

5 - Calificación obtenida

Posterior a obtener la calificación parcial de cada afirmación se suman y multiplica por 2,5. La literatura registra que la nota media de los sistemas de propósito general es de 68.

Con el fin de expresar de forma adecuada los datos, estos fueron promediados y se muestran en la Figura 7.

El software con alta usabilidad será a partir de ahora el 1 y el de baja usabilidad 2.

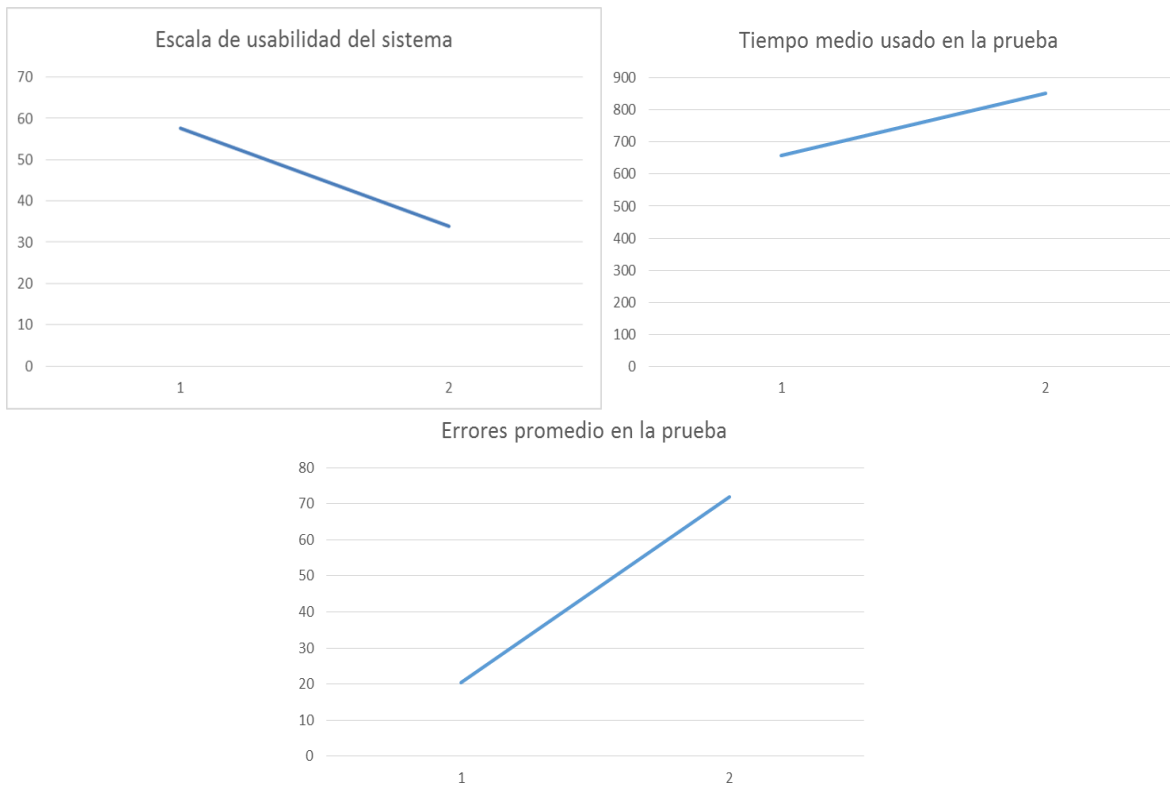


Figura 7: Grafico de dispersión de escala de usabilidad del sistema, tiempo medio usado en la prueba y errores promedio en la prueba contra los software.

Se puede observar en la Figura 7, que el tiempo y los errores son más bajos en el software con una alta usabilidad respecto al que presenta un bajo índice de usabilidad. Así mismo, la encuesta de la escala de usabilidad indico que el software 1 presenta una mayor escala de usabilidad respecto al 2.

Lo anterior indica (a pesar de los pocos datos obtenidos) que el índice de usabilidad puede predecir el tiempo medio usado en la prueba y los errores promedio en la prueba.

Las Figuras 8 y 9 a simple vista muestra una clara relación entre las variables de los datos tomados en las pruebas; siendo tiempo contra errores la que muestra una relación lineal.

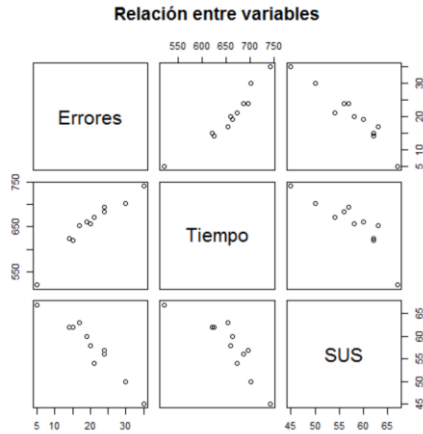


Figura 8: Grafico de Relación entre variables del software 1

Parece existir una relación positiva entre las variables Tiempo y Errores, conforme aumenta Tiempo, lo hace también Errores. Y una relación negativa entre las dos mencionadas variables con SUS, a medida que aumenta Tiempo y Errores, SUS disminuye.

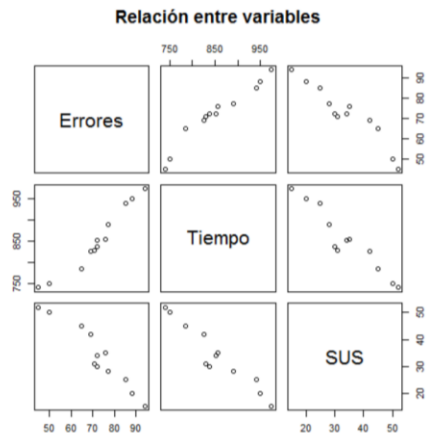


Figura 9: Grafico de Relación entre variables del software 2

De acuerdo a los resultados obtenidos en la regresión lineal que se muestra en la Tabla 8, en el modelo 2 del software 1 muestra que Errores promedio en la prueba es un predictor significativo de Tiempo para Software 1 ($t=0,000788$) y que Escala de usabilidad del sistema no fue significativa el tiempo más allá de los Errores ($t=0,086159$). Ambos modelos resultaron ser significativos. Al incluir Escala de usabilidad del sistema el R^2 aumento, no obstante al no aportar de manera significativa al modelo se eliminó. Los coeficientes de esta regresión muestran la

magnitud y la influencia en la variable Tiempo (variable dependiente). En esta regresión, se prueba el supuesto expresado en la Figura 8, Errores tienen una correlación positiva con el Tiempo, Escala de usabilidad del sistema tiene un impacto negativo sobre Tiempo.

Tabla 8: Modelo de regresión

| Modelo | R ² | R ² Ajustado | Significancia variable 1 | Significancia variable 2 | Estimado intercepto | Estimado variable 1 | Estimado variable 2 | Valor p |
|------------|----------------|----------------------------|-----------------------------|-----------------------------|------------------------|------------------------|------------------------|----------|
| Software 1 | | | | | | | | |
| 1 | 0,919 | 0,91 | 0,00000328 | -- | 519,034 | 6,8 | -- | 3,28E-06 |
| 2 | 0,9452 | 0,9315 | 0,000788 | 0,086159 | 148,454 | 10,595 | -5,091 | 9,01E-06 |
| Software 2 | | | | | | | | |
| 1 | 0,9407 | 0,9347 | 0,000000186 | -- | 481,5893 | 5,14 | -- | 1,06E-06 |
| 2 | 0,953 | 0,9425 | 0,030571 | 0,1591 | 698,394 | 3,276 | -2,417 | 1,86E-07 |

Nota: 1= Predictores: Errores promedio en la prueba. 2 = Predictores: Errores promedio en la prueba, escala de usabilidad del sistema.

El mismo proceso se realizó con el Software 2, los resultados de la regresión son mostrados en la Tabla 8, en el modelo 2, se identifica que al agregar Escala de usabilidad del sistema el R² crece respecto al modelo 1. No obstante, Escala de usabilidad no es significativa en el modelo (t=0,1591). Similares conclusiones del modelo del Software 1 se pueden extraer en este caso; Errores promedio en la prueba tienen una correlación positiva con el Tiempo, Escala de usabilidad del sistema tiene un impacto negativo sobre Tiempo. Ambos modelos son significativos.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

Aun en el siglo XXI existen numerosas enfermedades neurológicas que alteran la motricidad las cuales son provocadas por interrupciones en las señales enviadas por el cerebro. Inicialmente puede presentarse debilidad muscular e ir progresando hasta que al final se pierde la capacidad de controlar el movimiento voluntario. Este tipo de enfermedades son hereditarias o de aparición espontánea, por ello ninguna persona se encuentra libre de riesgo. A pesar de esto, existe muy poco estudio relacionado con el cómo acceden a la tecnología.

Esta investigación dentro de los límites relacionados con el poco (nulo) apoyo institucional obtenido en términos de recursos económicos consiguió establecer un método cuantitativo que mide la usabilidad de productos de software que brindan acceso al computador de personas con enfermedades que alteran la motricidad. La posibilidad de que el usuario final asigne pesos a las heurísticas, indicando cuáles son las más importantes para ellos y no que un experto fije pesos sin conocer las necesidades de estas personas.

6.2. Trabajo futuro

- Incrementar el número de heurísticas para obtener una mejor puntuación.
- Incrementar la cantidad de usuarios para mejorar el ajuste tanto en los pesos dados en las heurísticas como en el modelo de regresión.

Apéndice A: Heurísticas de Nielsen

“Diálogo simple y natural: Diálogos no deben contener información que sea irrelevante o raramente necesario. Cada unidad adicional de información en un diálogo compite con las unidades pertinentes de información y disminuye su visibilidad relativa. Toda la información debe aparecer en un orden natural y lógico.

Hablar el lenguaje del usuario: El diálogo debe ser claramente expresado en palabras, frases y conceptos familiares para el usuario, en lugar de en términos orientados al sistema.

Reducir al mínimo la carga de memoria de los usuarios: El usuario no debería tener que recordar la información de una parte del diálogo a otro. Instrucciones de uso del sistema deben ser visibles o fácilmente recuperable, cuando cada vez apropiado.

Consistencia: Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo.

Comentarios: El sistema siempre debe mantener a los usuarios informados sobre lo que está pasando, a través de la retroalimentación adecuada en un tiempo razonable.

Salidas claramente marcadas: los usuarios suelen elegir las funciones del sistema por error y necesitarán un marcado claramente "Salida de emergencia" para salir del estado no deseado sin tener que pasar a través de un diálogo ampliado.

Accesos directos: Aceleradores-invisibles por el novato usuario puede a menudo acelerar la interacción para el usuario experto de tal manera que el sistema puede servir tanto a los usuarios sin experiencia y con experiencia.

Los buenos mensajes de error: Deben ser expresadas en un lenguaje sencillo (sin códigos), indican precisamente el problema

Evitar errores: Incluso mejor que buenos mensajes de error es un cuidado diseño que impide que un problema se produzca en el primer lugar.

Ayuda y documentación: A pesar de que es mejor si el sistema puede ser utilizado sin la documentación, puede ser necesario proporcionar ayuda y documentación. Dicha información debe ser fácil de buscar, estar centrado en la tarea del usuario, lista de medidas concretas para llevar a cabo, y no ser demasiado grande. “

Texto original en Ingles [30].

Apéndice B: Validación de los modelos

Modelo 1 Software 1

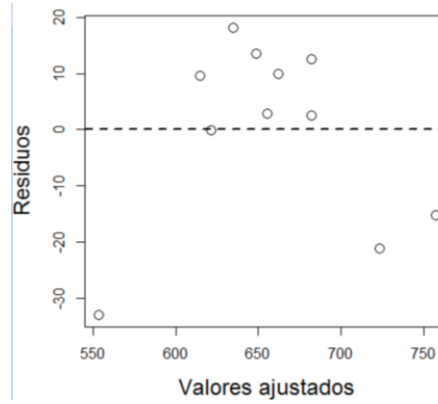


Figura 10: Validación de supuestos (residuales)

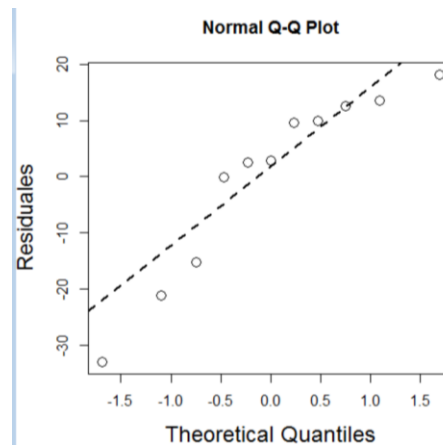


Figura 11: Normalidad

Shapiro-Wilk normality test

```
data: residuals(modelo)  
W = 0.88464, p-value = 0.1191
```

Modelo 2 Software 1

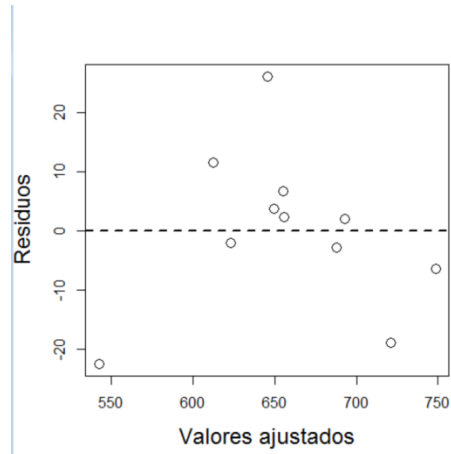


Figura 12: Validación de supuestos (residuales)

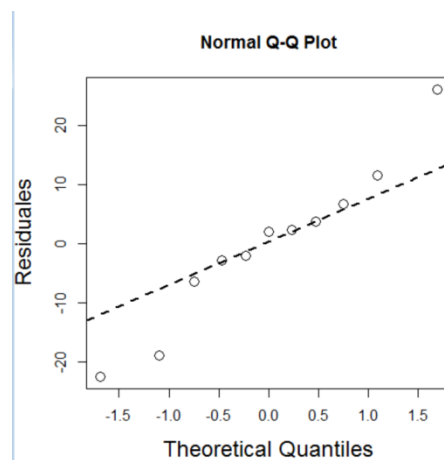


Figura 13: Normalidad

Shapiro-Wilk normality test

```
data: residuals(modelo)
W = 0.95806, p-value = 0.7473
```

Los modelos 1 y 2 del software 1 cuentan con evidencia muestral para sugerir que los errores distribuyen normal.

Modelo 1 Software 2

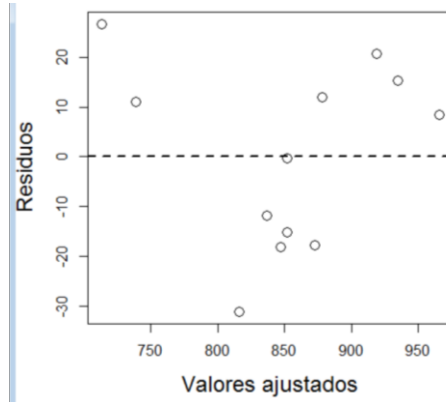


Figura 14: Validación de supuestos (residuales)

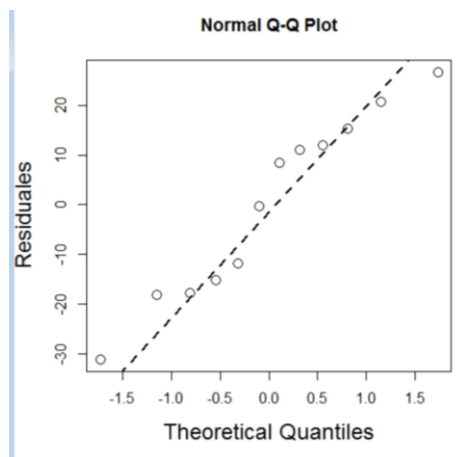


Figura 15: Normalidad

Shapiro-Wilk normality test

```
data: residuals(modelo)  
W = 0.94237, p-value = 0.5293
```

Modelo 2 Software 2

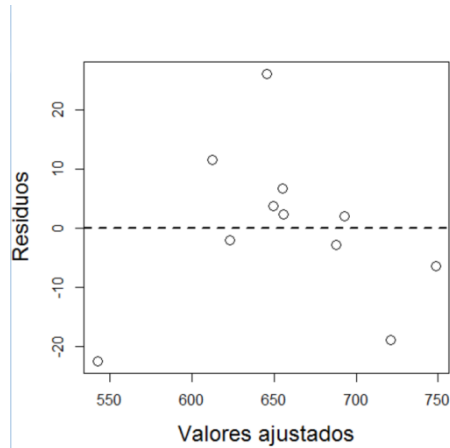


Figura 16: Validación de supuestos (residuales)

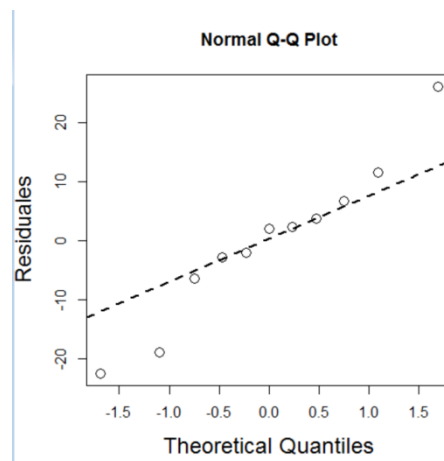


Figura 17: Normalidad

Shapiro-Wilk normality test

```
data: residuals(modelo)
W = 0.95806, p-value = 0.7473
```

Al igual que los anteriores modelos, el modelo 1 y 2 del software 2 cuentan con evidencia muestral para sugerir que los errores distribuyen normal.

7. Referencias

- [1] T. Jokela, N. Iivari, J. Matero, and M. Karukka, “The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11,” in *Proceedings of the Latin American conference on Human-computer interaction*, 2003.
- [2] X. Ferré Grau, “Marco de integración de la usabilidad en el proceso de desarrollo software.” Facultad de Informática (UPM), 12-Jul-2005.
- [3] J. Mankoff, H. Fait, and R. Juang, “Evaluating accessibility by simulating the experiences of users with vision or motor impairments,” *IBM Systems. Journal.*, vol. 44, no. 3, pp. 505–517, 2005.
- [4] J. Nielsen, “Usability inspection methods,” in *Conference companion on Human factors in computing systems.* , 1994.
- [5] A. Holzinger, “Usability engineering methods for software developers,” *Communications of the ACM*, 2005.
- [6] J. Bradford, “Evaluating high-level design: Synergistic use of inspection and usability methods for evaluating early software designs,” *Usability Inspections. methods*, 1994.
- [7] M. Macleod and R. Rengger, “The development of DRUM: A software tool for video-assisted usability evaluation,” *People Computing.*, 1993.
- [8] J. Hom, “The usability methods toolbox handbook,” 1998.
- [9] M. Camargo, L. Wendling, and E. Bonjour, “A fuzzy integral based methodology to elicit semantic spaces in usability tests,” *International Journal of Industrial Ergonomics.*, vol. 44, no. 1, pp. 11–17, Jan. 2014.

- [10] K. R. Lynch, D. J. Schwerha, and G. a. Johanson, "Development of a Weighted Heuristic for Website Evaluation for Older Adults," *International. Journal. Human. Computer. Interaction.*, vol. 29, no. 6, pp. 404–418, Jun. 2013.
- [11] A. Seffah, M. Donyaee, R. Kline, and H. Padda, "Usability measurement and metrics: A consolidated model," *Software. Quality. Journal.*, 2006.
- [12] G. Stucki, A. Cieza, and J. Melvin, "The International Classification of Functioning, Disability and Health (ICF): a unifying model for the conceptual description of the rehabilitation strategy.," *Journal. Rehabilitation. Medical.*, vol. 39, no. 4, pp. 279–85, 2007.
- [13] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, p. 1257., 2008,
- [14] H. Koester, "Abandonment of speech recognition by new users," *Proceedings. RESNA*, 2003.
- [15] B. Phillips and H. Zhao, "Predictors of assistive technology abandonment.," *Assistive. Technology.*, vol. 5, no. 1, pp. 36–45, Jan. 1993.
- [16] M. J. Scherer, "The change in emphasis from people to person: introduction to the special issue on Assistive Technology," *Disability and. Rehabilitation.*, vol. 24, no. 1–3, pp. 1–4, Jan. 2002.
- [17] S. Keates, J. Clarkson, and P. Street, "Investigating the Applicability of User Models for Motion-Impaired Users," pp. 129–136, 2000.
- [18] S. Keates, P. Langdon, P. J. Clarkson, and P. Robinson, "User Models and User Physical Capability," *User Model. User-adapter. Interaction.*, vol. 12, no. 2–3, pp. 139–169, Jun. 2002.

- [19] C. Law, A. Sears, and K. Price, "Issues in the categorization of disabilities for user testing," *Proceedings of HCII*, 2005.
- [20] J. Jacquet and A. Abran, "From software metrics to software measurement methods: a process model," *Software. Engineer. Standart*, 1997.
- [21] I. ISO, "IEC 25010: 2011: Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality," *International. Organization. Standart.*, 2011.
- [22] W. Humphrey, "Acquiring quality software," , 2005.
- [23] J. Nielsen, "Usability inspection methods," in *Conference companion on Human factors in computing systems - CHI '94*, 1994, pp. 413–414.
- [24] J. Nielsen, "Usability engineering," 1994, Elsevier.
- [25] X. Ferré Grau, "Principios Básicos de Usabilidad para Ingenieros Software," *Nueva Publicación de Prueba*, vol. 1, no. 1. p. 8, 05-Aug-2010.
- [26] A. Aipo, "Promoción de la Internación Personal-Ordenador, la Usabilidad y el Diseño centrado en el usuario. Libro IPO, El Diseño Ubicuo. [http: aipo. es/libro,](http://aipo.es/libro/)" 2002.
- [27] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, pp. 249–256.1990.
- [28] J. Nielsen, L. A. Blatt, J. Bradford, and P. Brooks, "Usability Inspection," in *Conference companion on Human factors in computing systems*. pp. 413–414, 1994.
- [29] J. Nielsen and R. Molich, "Teaching user interface design based on usability engineering," *ACM SIGCHI Bulletin* pp 45-48., 1989.

- [30] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* pp. 152–158., 1994,
- [31] M. Obeso, "Metodología de medición y evaluación de la usabilidad en sitios web educativos," 2005.
- [32] N. Jacobsen, M. Hertzum, and B. John, "The evaluator effect in usability tests," in *CHI 98 Conference Summer*, 1998.
- [33] J. Brooke, "SUS-A quick and dirty usability scale," *Usability Evaluation in Industry.*, 1996.
- [34] J. Preece, Y. Rogers, H. Sharp, and D. Benyon, "*Human-computer interaction*," 1994.
- [35] C. E. Jackson and J. Rosenfeld, "Motor neuron disease.," *Physics. Medical. Rehabilitation. Clinic. Neuron. Am.*, vol. 12, no. 2, pp. 335–52, ix–x, May 2001.
- [36] K. Talbot and R. Marsden, "Motor neuron disease," 2008.
- [37] L. Lacomblez, G. Bensimon, P. N. Leigh, P. Guillet, and V. Meininger, "Dose-ranging study of riluzole in amyotrophic lateral sclerosis. Amyotrophic Lateral Sclerosis/Riluzole Study Group II.," *Lancet*, vol. 347, no. 9013, pp. 1425–31, May 1996.
- [38] R. Worton, "ENHANCED PERSPECTIVE: Muscular Dystrophies--Diseases of the Dystrophin-Glycoprotein Complex," *Science (80-.)*, vol. 270, no. 5237, pp. 755–755, Nov. 1995.
- [39] S. Lefebvre, L. Bürglen, and S. Reboullet, "Identification and characterization of a spinal muscular atrophy-determining gene," *Cell*, 1995.

- [40] C. Okoli and S. D. Pawlowski, "The Delphi method as a research tool: an example, design considerations and applications," *Information & management*, 42(1), 15-29.., vol. 42, no. 1, pp. 15–29, Dec. 2004.
- [41] H. Linstone and M. Turoff, "The Delphi method: Techniques and applications," 1975.
- [42] E. Astigarraga, "El método delphi," *San Sebastián Univ. Deusto*, 2003.
- [43] G. Vanderheiden and J. Tobias, "Barriers, incentives and facilitators for adoption of universal design practices by consumer product manufacturers," *Proceedings. Human Factors*, 1998.
- [44] H. Petrie, G. Weber, and W. Fisher, "Personalization, interaction, and navigation in rich multimedia documents for print-disabled users," *IBM System. Journal.*, 2005.
- [45] M. Story, "Principles of universal design," *Universal. Design. Handbook.*, 2001.
- [46] J. Treviranus and J. Richards, "Authoring Tool Accessibility Guidelines," 1999, Available: www.w3.org/TR/WD-WAI-AUTOOLS.
- [47] T. Sullivan and R. Matson, "Barriers to use: usability and content accessibility on the Web's most popular sites," in *Proceedings on the 2000 conference on Universal Usability pp. 139-144*, 2000.
- [48] H. Petrie, F. Hamilton, N. King, and P. Pavan, "Remote usability evaluations With disabled people," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, p. 1133-1141, 2006.
- [49] D. Dykstra, "A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist," *A & M University*, 1993.
- [50] D. Pinelle and N. Wong, "Heuristic evaluation for games," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, 2008, p. 1453.

- [51] J. Spool and W. Schroeder, "Testing web sites," *CHI '01 extended abstracts on Human factors in computing systems - CHI '01*, 2001, p. 285.
- [52] R. Atterer, M. Wnuk, and A. Schmidt, "Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction," in *Proceedings of the 15th international conference on World Wide Web*, 2006.
- [53] S. Ssemugabi and R. De Villiers, "A comparative study of two usability evaluation methods using a web-based e-learning application," in *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pp. 132-142, 2007.
- [54] A. Nivala, S. Brewster, and T. Sarjakoski, "Usability evaluation of web mapping sites," *The Cartographic. Journal*, 2008.
- [55] M. Rowan, P. Gregor, D. Sloan, and P. Booth, "Evaluating web resources for disability access," in *Proceedings of the fourth international ACM conference on Assistive technologies*, pp. 80–84, 2000.
- [56] K. An, D. Kim, and J. Kim, "Development of a Virtual Keyboard System Using a Bio-signal Interface and Preliminary Usability Test," *Human-Computer Interaction Toward. Intelligent*, pp. 3–9, 2013.
- [57] R. Molich and D.- Ballerup, "HEURISTIC EVALUATION," no. April, pp. 249–256, 1990.
- [58] K. Karvonen, "The beauty of simplicity," in *Proceedings on the 2000 conference on Universal Usability*, 2000.
- [59] J. Jeng, "Usability assessment of academic digital libraries: Effectiveness, efficiency, satisfaction, and learnability," *Libri*, 2005.
- [60] M. Van Welie, "Breaking down usability," in *Proceedings of INTERACT*, 1999.

- [61] H. Banati, P. Bedi, and P. Grover, “Evaluating web usability from the user’s perspective,” *Journal. Computer. Science.*, 2006.
- [62] M. F. Story, “Maximizing usability: the principles of universal design.,” *Assistive. Technology.*, vol. 10, no. 1, pp. 4–12, 1998.
- [63] D. Norman, “The design of every things,” in *New York Currency.*, 1988.
- [64] J. Nielsen and T. Landauer, “A mathematical model of the finding of usability problems,” in *Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*, 1993.
- [65] J. Nielsen, “How Many Test Users in a Usability Study?,” 2012. [Online]. Available: <http://www.nngroup.com/articles/how-many-test-users/>.
- [66] L. Faulkner, “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing,” *Behavavior. Resolutions. Methods, Instruments, & Computers* , 2003.
- [67] J. Yarce, C. Lopera, and I. Pacheco, “La educación superior en Colombia,” 2002.
- [68] Congreso. de Colombia, “Constitución política de Colombia,” *Bogotá, Colomb. Leyer*, 1991.
- [69] Departamento Administrativo Nacional de Estadística (DANE), “Discapacidad Total Nacional,” 2007. [Online]. Available: http://www.dane.gov.co/files/investigaciones/discapacidad/Total_nacional.xls.
- [70] L. Masip, T. Granollers, and M. Oliva, “A Heuristic Evaluation Experiment to Validate the New Set of Usability Heuristics,” in *2011 Eighth International Conference on Information Technology: New Generations*, 2011, pp. 429–434.