ESCUELA POLITÉCNICA NACIONAL

INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN MÓVIL EDUCATIVA PARA EL ENTRENAMIENTO DE HABILIDADES COGNITIVAS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

ANDRÉS DAVID PAREDES RUANO
paredes.r.andres@gmail.com

CARLOS EDUARDO SAMPEDRO MORA
carlos.e.sampedro.91@gmail.com

DIRECTOR: PhD. MARCO SANTÓRUM
marco.santorum@epn.edu.ec

CO-DIRECTOR: MSc. MAYRA CARRIÓN
mayra.carrion@epn.edu.ec

Quito, enero 2019

DECLARACIÓN

Nosotros, Andrés David Paredes Ruano y Carlos Eduardo Sampedro Mora declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Andrés David Paredes Ruano

Carlos Eduardo Sampedro Mora

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Andrés David Paredes Ruano y Carlos Eduardo Sampedro Mora, bajo nuestra supervisión.

PhD. Marco Santórum	MSc. Mayra Carrión
Director	Co-Director

CONTENIDO

RESUMEN ..... **X**

ABSTRACT ..... **XI**

CAPÍTULO 1 ..... **1**

INTRODUCCIÓN **1**

1.1 Situación actual del problema 1

1.2 Objetivos 2

1.2.1 Objetivo General 2

1.2.2 Objetivos Específicos 2

1.3 Marco conceptual 2

1.3.1 TIC en la educación 2

1.3.2 Discapacidad intelectual y las TIC 4

1.3.3 Definición de discapacidad 5

CAPÍTULO 2 ..... **7**

METODOLOGÍA **7**

2.1 Scrum y Extreme Programming (XP) 7

2.2 Propuesta, herramientas y recursos utilizados 10

2.2.1 Modelo Vista Controlador 10

2.2.2 Aplicaciones Nativas 11

2.2.3 Herramientas y Recursos 12

2.2.4 Arquitectura de la aplicación 13

2.3 Desarrollo del sistema 14

2.3.1 Caso de estudio 14

2.3.2 Requerimientos 15

2.3.3 Diseño de la aplicación 17

2.3.4 Evaluación heurística 17

2.3.5 Definición de módulos de la aplicación 19

2.3.6 Diagrama de Navegación de la aplicación 21

2.3.7 Historias de usuario épicas 22

2.3.8 Product Backlog 23
2.3.9 Release Planning 25
2.3.10 Roles 26
2.4 Ejecución de Sprints 26
2.4.1 Sprint 1 26
2.4.2 Sprint 2 32
2.4.3 Sprint 3 44
2.4.4 Sprint 4 46
2.4.5 Sprint 5 49
CAPÍTULO 3 52
EVALUACIÓN Y RESULTADOS 52
3.1 Pruebas de funcionalidad 52
3.1.1 Módulo de navegación 52
3.1.2 Módulo de registro 53
3.1.3 Módulo personalización de talleres 56
3.1.4 Módulo de entrenamiento 59
3.1.5 Módulo de seguimiento y evaluación 63
3.2 Pruebas Usabilidad 64
3.2.1 Ejecución de pruebas 66
CAPÍTULO 4 71
CONCLUSIONES 71
PERSPECTIVAS Y RECOMEDACIONES 72
REFERENCIAS BIBLIOGRÁFICAS 74
ANEXOS 79
ANEXO A (JUSTIFICACIÓN Y PROTOCOLO DE LA METODOLOGÍA iPLUS) 79
ANEXO B (DESARROLLO) 79
ANEXO C (INSTRUMENTOS EVALUACIÓN) 79
ANEXO D (RESULTADOS) 80

FIGURAS

Figura 1: Modelo MVC (adaptado de [36]).....10

Figura 2: Modelo Arquitectura de una aplicación nativa (adaptado de [37]).....11

Figura 3. Arquitectura de la aplicación (elaborado por los autores) 14

Figura 4: Metodología iPlus junto con Scrum y XP (elaborado por los autores).....15

Figura 5: Pantalla inicial mockup 17

Figura 6: Resultado de la evaluación heurística 19

Figura 7: Interfaces que representan el módulo de registro.....20

Figura 8: Interfaces que representan el módulo de personalización de talleres.....20

Figura 9: Interfaces que representan el módulo de entrenamiento 21

Figura 10: Interfaces que representan el módulo de seguimiento y evaluación 21

Figura 11: Diagrama de navegación (elaborado por los autores) 22

Figura 12: Pantalla inicial, mockup vs interfaz desarrollada.....30

Figura 13: Menu de la aplicación, mockup vs interfaz desarrollada 31

Figura 14: Burndown Chart – Primer Sprint 31

Figura 15: Pantalla login, mockup vs interfaz desarrollada 43

Figura 16: Registro tutor, mockup vs interfaz desarrollada 43

Figura 17: Registro estudiante, mockup vs interfaz desarrollada.....43

Figura 18: Burndown Chart – Segundo Sprint 44

Figura 19: Lista de talleres registrados, mockup vs interfaz desarrollada 45

Figura 20: Agregar/editar palabra, mockup vs interfaz desarrollada 45

Figura 21: Creación de secuencia, mockup vs interfaz desarrollada 45

Figura 22: Burndown Chart – Tercer Sprint 46

Figura 23: Pantalla entrenamiento, mockup vs interfaz desarrollada.....47

Figura 24: Selección historia, mockup vs interfaz desarrollada.....47

Figura 25: Pantalla ordenar secuencia, mockup vs interfaz desarrollada 48

Figura 26: Resultados, mockup vs interfaz desarrollada.....48

Figura 27: Burndown Chart – Cuarto sprint 49

Figura 28: Gráfica estudiante, mockup vs interfaz desarrollada.....50

Figura 29: Tabla estudiante, mockup vs interfaz desarrollada 50

Figura 30: Burndown Chart – Quinto sprint.....51

Figura 31: Nube de palabras 68

Figura 32: Puntaje promedio SUS (adaptado de [60]) 69

TABLAS

Tabla 1: Herramientas y Recursos Utilizados. 13

Tabla 2: Historia de usuario épica HUE01 22

Tabla 3: Historia de usuario épica HUE02. 22

Tabla 4: Historia de usuario épica HUE03. 23

Tabla 5: Historia de usuario épica HUE04. 23

Tabla 6: Historia de usuario épica HUE05. 23

Tabla 7: Product Backlog. 25

Tabla 8: Release Planning 25

Tabla 9: Roles Scrum 26

Tabla 10: Historia de usuario 01 - Sprint 1....27

Tabla 11: Historia de usuario 02 - Sprint 1....27

Tabla 12: Sprint Backlog - Sprint 1. 29

Tabla 13: Pruebas de aceptación - Historia de usuario 01....30

Tabla 14: Pruebas de aceptación - Historia de usuario 02....30

Tabla 15: Historia de usuario 03 - Sprint 2....32

Tabla 16: Historia de usuario 04 - Sprint 2....33

Tabla 17: Historia de usuario 05 - Sprint 2....34

Tabla 18: Historia de usuario 06 - Sprint 2....35

Tabla 19: Historia de usuario 04 - Sprint 2....35

Tabla 20: Sprint Backlog - Sprint 2. 39

Tabla 21: Pruebas de aceptación - Historia de usuario 03....40

Tabla 22: Prueba de aceptación - Historia de usuario 04. 41

Tabla 23: Pruebas de aceptación - Historia de usuario 05....41

Tabla 24: Pruebas de aceptación - Historia de usuario 06....42

Tabla 25: Pruebas de aceptación - Historia de usuario 07....42

Tabla 26: Pruebas de funcionalidad - Sprint 1. 53

Tabla 27: Pruebas de funcionalidad - Sprint 2. 55

Tabla 28: Pruebas de funcionalidad - Sprint 3. 58

Tabla 29: Pruebas de funcionalidad - Sprint 4. 62

Tabla 30: Pruebas de funcionalidad - Sprint 5. 64

Tabla 31: Lista de palabras adaptadas a lectura fácil. 65

Tabla 32: Palabras escogidas por los estudiantes.....67

Tabla 33: Resultados encuesta SUS 70

RESUMEN

El presente trabajo tiene como objetivo principal entrenar habilidades cognitivas mediante la resolución de secuencias temporales y la revisión de vocabulario, por tanto, se describe el desarrollo de una aplicación móvil educativa para el entrenamiento de habilidades cognitivas.

En ese sentido, se procedió a realizar en principio un análisis de la situación actual del problema de manera resumida con el fin de obtener un diagnóstico, y definir tanto los objetivos del trabajo como la base conceptual que permitió sustentar el proyecto.

Posteriormente, se definió la metodología más apropiada y se procedió con el desarrollo de la aplicación, lo cual conllevó realizar una descripción del caso de estudio, la toma de requerimientos, el diseño de la aplicación y su implementación final.

Luego se procedió con el análisis de los resultados obtenidos tras la elaboración del aplicativo móvil, para lo cual se realizaron pruebas tanto de usabilidad como de funcionalidad, con el propósito de comprobar el cumplimiento de aplicativo respecto a los requerimientos y objetivos planteados al inicio del proyecto.

Finalmente, se presentan las conclusiones y recomendaciones obtenidas durante el desarrollo del proyecto, cada uno de los objetivos que fueron propuestos tienen al menos una conclusión que lo justifique.

Palabras Clave: Aplicación móvil, software educativo, discapacidad, TIC, enfoque ágil.

ABSTRACT

The main objective of this work is to train cognitive skills by solving temporal sequences and reviewing vocabulary, therefore, the development of an educational mobile application for the training of cognitive skills is described.

In this sense, an analysis of the current situation of the problem was carried out in a summarized manner in order to obtain a diagnosis, and define both the objectives of the work and the conceptual basis that supported the project.

Subsequently, the most appropriate methodology was defined and the application was developed, which entailed making a description of the case study, taking of requirements, the application design and its final implementation.

Then after the development of the mobile application, the obtained results were analysed, for which usability and functionality test were carried out in order to check compliance with the the requirements and objectives raised at the beginning of the project.

Finally, the obtained conclusions and recommendations during the development of the project are presented, each of the proposed objectives has at least one conclusion that justifies it.

Keywords: Mobile application, educational software, Disability, ICT, agile approach

CAPÍTULO 1

INTRODUCCIÓN

1.1 Situación actual del problema

En la sociedad contemporánea, las personas con discapacidad intelectual constituyen uno de los grupos más propensos a tener dificultades dentro de la integración social; sobre todo en el aspecto laboral, debido a la limitación que presentan para adaptarse y desarrollar actividades en ambientes desconocidos [1] .

Según el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), el 23 % de la población con discapacidad en el Ecuador sufre de discapacidad mental, entre los cuales el 66.08%, se encuentra en edades de 18 a 65 años, es decir, alrededor de 66 mil personas se ubican en la edad para realizar actividades laborales; sin embargo, no pueden desempeñarlas debido a su condición [2].

Por lo tanto, existe una creciente preocupación por eliminar hasta donde sea posible dichas desventajas por medio de acciones específicas. De acuerdo con [3], se debe recuperar la o las funciones faltantes y, cuando no sea posible la completa recuperación, compensarla con la rehabilitación, la cual consiste tanto en desarrollar las habilidades y destrezas necesarias como en dotar a las personas de elementos compensatorios.

Hoy las nuevas tecnologías de información y comunicación (TIC) han dado un notable cambio a la sociedad, incluso a las relaciones sociales y productivas, pues estas herramientas son consideradas como indispensables para la producción y transmisión de conocimientos, la generación de riqueza y en general, el desarrollo progresivo de la humanidad. Tal condición resulta aplicable a todos los grupos de población, particularmente, para las personas que ven disminuidas sus posibilidades por alguna situación de discapacidad física, cognitiva o sensorial [4].

Las aplicaciones móviles como un segmento emergente de las TIC han llegado a ser un elemento fundamental en la construcción de conocimiento, por cuanto a través de su utilización se incrementan las posibilidades de interactuar con los miembros del grupo de estudio y mejora la comunicación; por lo tanto, se difumina la barrera que separa a docentes y alumnos. La tendencia actual hacia el uso de dispositivos móviles en educación se encuentra enfocada en que cada vez se utilicen con mayor frecuencia estos aparatos en las aulas y en los centros educativos y culturales en el futuro [5].

La Fundación para la Integración del Niño Especial (FINE) perteneciente a Quito en Ecuador, es una organización no gubernamental sin fines de lucro, que tiene como objetivo principal impulsar el desarrollo y bienestar del ser humano con discapacidad, a partir de la implementación de centros que faciliten una mayor integración a la sociedad [6]. Es por ello, que la referida fundación necesita una herramienta que permita el entrenamiento de habilidades cognitivas de personas con discapacidad intelectual y facilite la enseñanza de diversas actividades dentro del campo laboral, adaptándose a sus necesidades específicas, razón por la cual se plantea el desarrollo de una aplicación móvil que logre estos fines a través de ejercicios de ordenamiento de secuencias temporales y que representen escenarios laborales específicos, en los cuales previamente se mostrará el vocabulario de los diferentes espacios para una mejor concepción del escenario en cuestión.

1.2 Objetivos

1.2.1 Objetivo General

- Desarrollar mediante un enfoque ágil una aplicación móvil educativa que permita el entrenamiento de habilidades cognitivas, aplicado al aprendizaje de vocabulario y secuencias temporales de tareas laborales.

1.2.2 Objetivos Específicos

- Desarrollar como caso de estudio, un módulo de entrenamiento para vocabulario y secuencias temporales de tareas laborales y adicionalmente que esta permita la personalización de escenarios.
- Implementar un módulo de seguimiento de progreso del desarrollo de la habilidad cognitiva de las personas con discapacidad.
- Evaluar el caso de estudio mediante pruebas de funcionalidad y usabilidad.

1.3 Marco conceptual

1.3.1 TIC en la educación

Las Tecnologías de la Información y la Comunicación (TIC) en la actualidad son prácticamente omnipresentes, lo cual representa una oportunidad y un desafío para la sociedad, debido a que impone la tarea urgente de encontrar un gran sentido y uso en pro de lograr el desarrollo de sociedades más democráticas e inclusivas, que fortalezcan la colaboración, la creatividad y la distribución más justa del conocimiento científico, contribuyendo de esta manera a una educación más equitativa y de calidad para todos [7].

La incorporación de las TIC a la educación se ha convertido en un proceso que implica algo más allá de las herramientas tecnológicas que conforman el ambiente educativo, específicamente en lo que se refiere a una construcción didáctica y la manera de construir y consolidar un aprendizaje significativo con base en la tecnología, resaltando que las TIC han logrado convertirse en instrumentos educativos, capaces de mejorar la calidad educativa del estudiante, y revolucionando la forma en que se obtiene, se maneja y se interpreta la información [8].

Dentro de la gran cantidad de TIC dispuestas actualmente, conviene enfocarse particularmente en la denominada información digital, que se puede definir como programas de aplicación y programas que muestran o administran la información de una aplicación didáctica como páginas WEB, bases de datos, programas de aplicación de procesamiento de palabras, entre otros [9]. En este tipo de información se encuentran contenidas las aplicaciones o software educativo diseñadas para que los estudiantes pueden aprender, entrenarse, tomar decisiones dentro y fuera de un aula de clase [10].

Software educativo

Los términos de software educativo, programas educativos y programas didácticos son utilizados comúnmente como sinónimos para designar genéricamente a los programas para computador, creados con la finalidad específica de ser usados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje [11].

Todo software educativo comparte cinco características esenciales [11]:

- Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.
- Utilizan el ordenador como soporte donde los alumnos realizan las actividades que proponen.
- Son interactivos, contestan inmediatamente a las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- Individualizan el trabajo de los estudiantes, debido a que se adaptan al ritmo de trabajo de cada uno y pueden acoplar sus actividades según las actuaciones de los alumnos.
- Son fáciles de usar, por cuanto los conocimientos informáticos necesarios para utilizar la mayoría de estos programas resultan similares a los conocimientos de electrónica

requeridos para poner en marcha un video, es decir, son mínimos, aunque cada programa amerita conocer unas reglas de funcionamiento.

El software educativo constituye un valioso medio de enseñanza para la educación y formación general del estudiante, en virtud de sus características y tipo, que permiten su inserción dentro del proceso, en apoyo directo del proceso de enseñanza-aprendizaje, lo cual representa un efectivo instrumento para el desarrollo educacional del ser humano.

Tanto docentes como estudiantes se interesan cada vez más por la utilización de las TIC en el proceso educativo, partiendo de la premisa que dice “entre más visual se haga el aprendizaje, mayor será el volumen de contenidos que se logrará procesar e incorporar en forma de conocimientos”. De esta forma se alcanza también una mejor retención de conocimientos [10].

Como una característica principal de las TIC destaca su continua evolución, toda vez que aparecen nuevas tecnologías al mismo tiempo que otras se vuelven obsoletas. Desde la aparición de las tecnologías móviles, las mismas han demostrado ser prácticas y útiles para todo el mundo, y en la medida que estos dispositivos ganan protagonismo a nivel mundial, va surgiendo un gran interés en torno a cómo usar tales tecnologías en el ámbito educativo [12]. Es importante resaltar, que una gran parte de la población estudiantil cuenta con alguno de estos dispositivos o puede acceder a uno, debido al fácil acceso económico que representa su adquisición, en comparación con un computador convencional.

Cabe mencionar, que los dispositivos móviles por sí mismos no son útiles como herramientas educativas, por cuanto en un principio se comercializaron principalmente con la intención de ser utilizados como dispositivos de comunicación y entretenimiento, razón por la que han llegado a desempeñar un papel importante en la economía de cada país, y en el conjunto de la sociedad. Por tal motivo, se hace necesario el estudio de formas pedagógicas que permitan hacer uso de la interacción y colaboración entre usuarios, orientadas al aprendizaje [13]. El software educativo puede aprovechar las facilidades tanto visuales, ergonómicas y de movilidad ofrecidas por los dispositivos móviles, diseñando aplicaciones que estén orientadas al aprendizaje en diferentes lugares o escenarios específicos [14].

1.3.2 Discapacidad intelectual y las TIC

El uso de nuevas tecnologías ha introducido profundos cambios en el entorno de cualquier persona y el acceso a las mismas puede encontrarse condicionado por diferentes barreras para algunos individuos o grupos sociales. Frecuentemente, la aplicación de las TIC requiere de ciertos conocimientos o competencias que deben ser adquiridos, lo cual puede resultar

complicado para ciertos sectores sociales, prevaleciendo incluso el grupo de las personas con discapacidad intelectual [15].

Existen varias investigaciones que ha demostrado la manera en como las TIC ayudan a mejorar las competencias de los grupos de atención prioritaria como [16] [17]. En [18] se reflexiona de cómo las TIC pueden fortalecer las competencias de las personas con discapacidad intelectual, y la forma en que el software y los dispositivos ayudan a estos sujetos por medio de una metodología adecuada. Además, estudios sobre inteligencia, plasticidad cerebral y aprendizaje confirman el apoyo de las TIC en la modificación y mejora del potencial intelectual de las personas, a través de la estimulación multisensorial, las relaciones interpersonales y el aprendizaje activo.

Existe un sinnúmero de aplicaciones educativas que contribuyen al fortalecimiento de las competencias de estas personas, algunas de ellas son: La Mesita [19], evo [20], Abrapalabra [21], Stimulus Home [22], Soy Visual [23], entre otros.

1.3.3 Definición de discapacidad

La palabra discapacidad está formada por la preposición “dis” que indica negación, separación, anulación de una cosa, cualidad o función. La palabra capacidad hace referencia a un talento, suficiencia o inteligencia; entonces, discapacidad es la restricción o ausencia de la capacidad de realizar una actividad en la forma que se considera normal para una persona, como por ejemplo comunicarse, alimentarse, vestirse, correr, caminar, entre otros [24].

Discapacidad intelectual

También conocida como discapacidad cognitiva, se refiere a una seria de limitaciones en las habilidades diarias que una persona aprende. Según la Asociación Americana de Discapacidades Intelectuales y del Desarrollo (AAIDD): “La discapacidad intelectual es un estado individual que se caracteriza por presentar limitaciones significativas tanto en el funcionamiento intelectual como en la conducta adaptativa, tal y como se manifiesta en las habilidades adaptativas conceptuales, sociales y prácticas, y por ser originada antes de los 18 años” [25].

Tipos de discapacidad intelectual

La Discapacidad Intelectual se caracteriza por un retraso en el desarrollo del funcionamiento intelectual y por dificultades en el funcionamiento social adaptativo. A continuación, se describen cuatro niveles de gravedad [26]:

- Profunda: Estos individuos no pueden cuidar de sí mismos y no tienen lenguaje. Su capacidad para expresar emociones es limitada y difícil de comprender; con frecuencia presentan convulsiones, discapacidades físicas y tienen una expectativa de vida reducida.
- Grave: Todos los aspectos del desarrollo están retrasados, tienen dificultad para pronunciar palabras y un vocabulario muy limitado. Con práctica y tiempo considerable, pueden adquirir habilidades básicas de cuidar de sí mismos, pero todavía necesitan apoyo en la escuela, en casa y en la comunidad.
- Moderada: Demoran en alcanzar los hitos del desarrollo intelectual; su capacidad para aprender y pensar lógicamente está disminuida, pero son capaces de comunicarse y cuidar de sí mismos con algún apoyo. Con supervisión, pueden realizar trabajos no calificados o semi-especializados.
- Leve: Son capaces de comunicarse y aprender habilidades básicas. Su capacidad de usar conceptos abstractos, analizar y sintetizar se encuentra afectada, pero pueden llegar a leer y calcular a un nivel de tercer a sexto grado. Tienen la capacidad para hacer tareas domésticas, cuidarse a sí mismos y realizar trabajos no calificados o semi-especializados y requieren algún apoyo.

La aplicación que se pretende desarrollar en el presente proyecto se encuentra dirigida especialmente para personas con discapacidad intelectual leve y moderada, no obstante, puede ser usada por cualquier tipo de persona.

En definitiva, se ha realizado una revisión de la situación actual sobre la problemática de las TIC en el marco de la educación para personas con discapacidad intelectual, lo cual ha brindado aportes para la comprensión de la forma en que una aplicación educativa tiende a mejorar ciertas competencias de estas personas en diferentes ámbitos. A partir de esta revisión teórica, se han definido los objetivos generales y específicos del presente proyecto, y en el capítulo 2 se determina la metodología a utilizar para el desarrollo del aplicativo.

CAPÍTULO 2

METODOLOGÍA

2.1 Scrum y Extreme Programming (XP)

Bajo la consideración de los autores de este proyecto, se estimó que, para la realización del mismo conviene la definición de una metodología ágil capaz de adaptarse a nuevos requisitos emergentes y que no requiera documentación extensa, siendo la opción más adecuada para llevar a cabo el proyecto.

Es por ello, que se utilizó metodologías de desarrollo como Scrum y XP, debido a que ambas procuran un mayor énfasis en la adaptabilidad a los cambios, y por los escenarios en los que tienen mayor protagonismo estas metodologías, complementándose entre sí (anexo A.1).

Scrum

Define un marco para la gestión de proyectos y se encuentra especialmente indicada para proyectos con un rápido cambio de requisitos, teniendo como principales características: el desarrollo de software que se realiza mediante iteraciones denominadas sprints, con una duración de 30 días, destacando que el resultado de cada sprint es un incremento ejecutable mostrado al cliente; y las reuniones a lo largo proyecto, las cuales son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [27].

El Equipo Scrum

Los Equipos Scrum son auto-organizados y multifuncionales, siendo los primeros aquellos que eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo, mientras que los equipos multifuncionales cuentan con todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo, el cual consta de los siguientes roles [28]:

- Product Owner: El dueño de producto es el responsable de maximizar el valor del producto resultante del trabajo del Equipo de Desarrollo. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog).
- Scrum Master: es un líder al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser útiles y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

- Scrum Team: El Equipo de Desarrollo consiste en los profesionales que realizan el trabajo de entregar un incremento de producto “terminado” que potencialmente se pueda poner en producción al final de cada Sprint. Un Incremento “Terminado” es obligatorio en la Revisión del Sprint, resaltando que solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que resultan de gran utilidad para proporcionar transparencia y oportunidades, tanto para la inspección como la adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave necesaria para asegurar que todos tengan el mismo entendimiento del artefacto [29].

- Lista de Producto (Product Backlog): La Lista de Producto es una lista ordenada de todo lo que se conoce como necesario en el producto; siendo la única fuente de requisitos para cualquier cambio a realizarse en el mismo. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.
- Lista de Pendientes del Sprint (Sprint Backlog): La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el equipo de desarrollo acerca de qué funcionalidad formará parte del próximo incremento, y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.
- Burndown chart o gráfico de trabajo pendiente: permite conocer si el proyecto será finiquitado en la fecha indicada, por lo que muestra la velocidad con la que se van cumpliendo los objetivos. El trabajo pendiente se representa en el eje vertical y el tiempo en el eje horizontal (arquitectura de la aplicación).

Extreme Programing (XP)

Consiste en una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones

implementadas y coraje para enfrentar los cambios. XP se define como una herramienta especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico [30].

Historias de Usuario

Las historias de usuario se refieren a una técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, destacando que en cualquier momento éstas pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [31].

Por su parte, Scrum no recomienda ninguna buena práctica para el desarrollo del producto, mientras que XP recomienda algunas buenas prácticas entre las cuales se destacan las siguientes.

- Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores. Los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, donde varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo [32].
- Estándares de programación: XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios [33].

Scrum y la programación extrema

Estos proporcionan prácticas y reglas complementarias, fomentan valores similares, y minimizan las desconexiones problemáticas entre la administración y los desarrolladores. Tales metodologías combinadas, proporcionan una estructura dentro de la cual un cliente puede evolucionar un producto de software que mejor se adapte a sus necesidades e implementar funcionalidades de calidad en forma incremental, para aprovechar las oportunidades de negocios [34].

2.2 Propuesta, herramientas y recursos utilizados

2.2.1 Modelo Vista Controlador

Para el desarrollo del aplicativo se ha elegido como patrón de diseño el Modelo-Vista-Controlador, pues permite dividir la lógica de negocio del diseño, haciendo que la aplicación sea más escalable en el futuro.

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos [35].

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

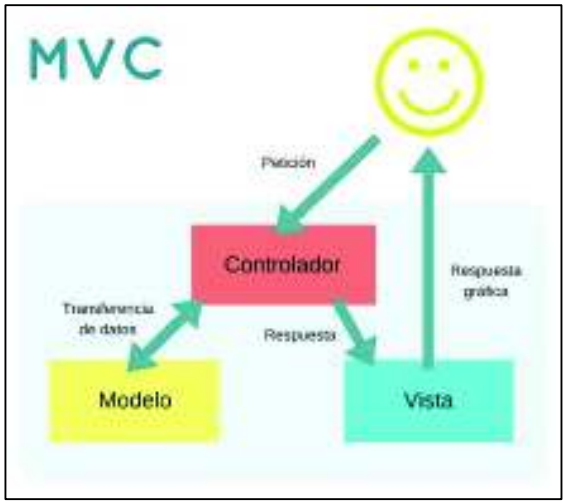


Figura 1: Modelo MVC (adaptado de [36])

2.2.2 Aplicaciones Nativas

Una App Nativa es una aplicación desarrollada con herramientas específicas para que éstas se ejecuten en el sistema operativo nativo de cada dispositivo. Estas aplicaciones pueden acceder a los sistemas operativos del equipo móvil, con la intención de facilitar el uso de dispositivos como brújula, cámara, correo, GPS, entre otros. Las mismas se instalan en el aparato y, normalmente, tienen un uso óptimo de la funcionalidad del móvil. En la figura 2 se muestran los componentes de la arquitectura de una aplicación nativa.

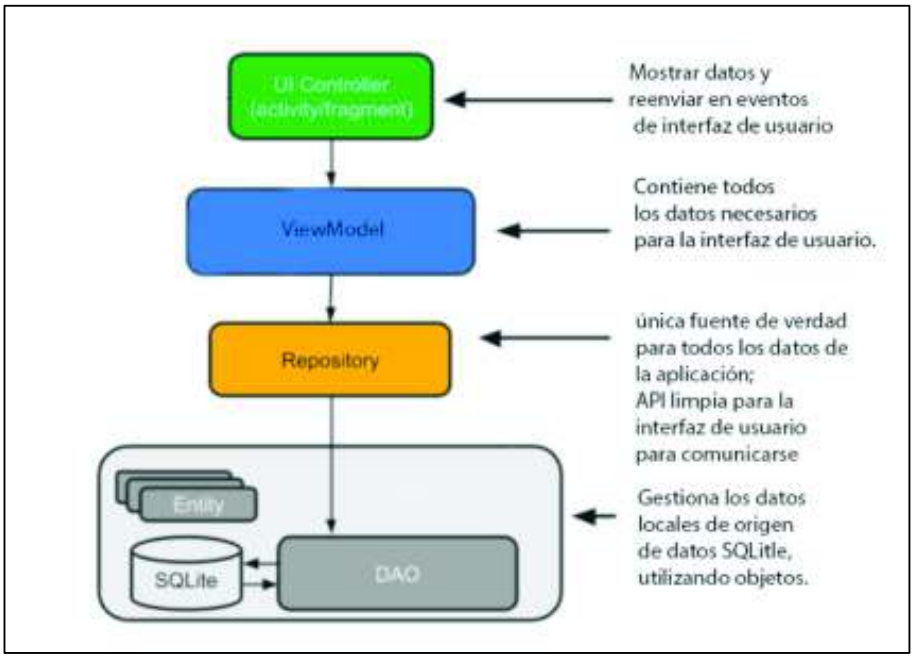


Figura 2: Modelo Arquitectura de una aplicación nativa (adaptado de [37])

- Data Access Objects (DAO): que significa objeto de acceso a datos, se refiere a un mapeo de consultas SQL a funciones. Se acostumbraba a definir estas consultas en una clase auxiliar, como SQLiteOpenHelper. Cuando utiliza un DAO, llama a los métodos y los componentes se encargan del resto.
- Repositorio: una clase que crea para administrar múltiples fuentes de datos.
- ViewModel: un objeto ViewModel proporciona los datos para un componente específico de la interfaz de usuario (IU), como un fragmento o actividad, y contiene lógica empresarial de manejo de datos para comunicarse con el modelo [37].

Cada componente depende solo del que se encuentra en un nivel inferior. Por ejemplo, las actividades y los fragmentos dependen solo de un modelo de vista, y el repositorio es la única

clase que depende de muchas otras clases; en este ejemplo, el repositorio depende de un modelo de datos persistente [38].

2.2.3 Herramientas y Recursos

La tabla 1 se muestra las herramientas y recursos tecnológicos utilizados para el desarrollo del presente proyecto.

Logo	Nombre	Descripción	Tipo
	Android Studio	Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android [39].	Programación
	SQLite	SQLite es un sistema de gestión de bases de datos relacional contenida en una relativamente pequeña biblioteca escrita en C [40].	Programación
	SQLite Manager	Es un potente sistema de gestión de bases de datos, combina una interfaz fácil de usar [41].	Programación
	Balsamiq	Herramienta que permite realizar prototipos de proyectos [42].	Diseño
	Adobe Photoshop	Es un editor de gráficos rasterizados que es usado principalmente para el retoque de fotografías y gráficos [43].	Diseño

	Adobe Ilustrador	Adobe Illustrator es un editor de gráficos vectoriales en forma de taller de arte que trabaja sobre un tablero de dibujo [44].	Diseño
	Flaticon	Es un repositorio online de iconos para aplicaciones web y móviles [45].	Diseño
	Icon8	Es un repositorio online de iconos para aplicaciones web y móviles [46].	Diseño
	Iloveimg	Es una app online para editar y gestionar imágenes completamente gratis y fácil de usar [47].	Diseño
	ZapsPlat	Es una biblioteca de efectos de sonido online gratuita [48].	Diseño

Tabla 1: Herramientas y Recursos Utilizados.

2.2.4 Arquitectura de la aplicación

La arquitectura propuesta contempla el desarrollo de una aplicación móvil para dispositivos Android, la cual está compuesta por tres capas: capa de presentación, capa de aplicación y capa de persistencia [49], como se muestra en la figura 3.

- Capa de presentación: se encarga de albergar toda la interfaz de usuario tablas, botones, menús, formularios, etc. Dicho nivel se encuentra compuesta por diversos archivos XML que se presentan al usuario.

- Capa de aplicación: se encuentra las clases como Activities, Fragments, etc., que obtienen información a través de la capa de presentación, y se encarga de gestionar los datos y enviarlos la capa de persistencia.
- Capa de persistencia: cuenta con las clases DAO que controlan el acceso a la base de datos SQLite ya sea agregando, modificando o eliminando datos.

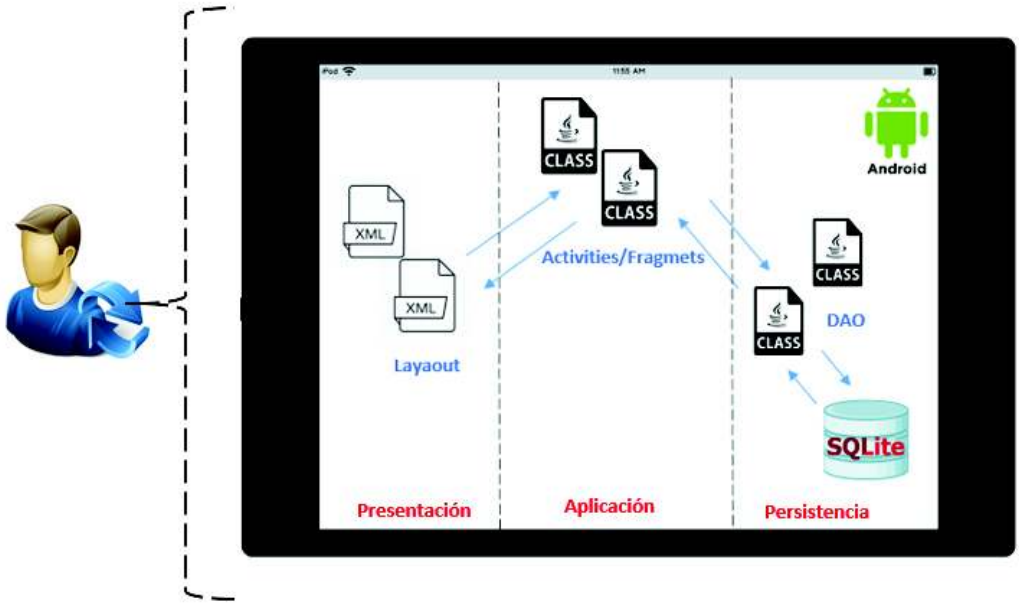


Figura 3. Arquitectura de la aplicación (elaborado por los autores)

2.3 Desarrollo del sistema

2.3.1 Caso de estudio

El presente proyecto consiste en el desarrollo de una aplicación móvil educativa para el entrenamiento de habilidades cognitivas, desarrollada con la participación de expertos de diferentes temáticas y autoridades de la fundación para la cual va a ser desarrollado el producto, cuyo propósito está dirigido al hecho que las personas con discapacidad la utilicen y se obtenga como resultado que mejoren sus habilidades cognitivas, aprendiendo a relacionarse mejor en ambientes desconocidos. La misma enseñará un vocabulario específico de escenarios laborales y secuencias temporales planteados por los terapeutas y especialistas de la fundación.

La aplicación móvil educativa se encuentra dirigida a personas con discapacidad intelectual dentro del rango de leve a moderado, y las mismas no deben presentar dificultades físicas que impidan el uso normal de un dispositivo móvil. La aplicación presenta un conjunto de actividades para ayudar a entrenar y/o mejorar el pensamiento lógico, la memoria y

comprensión de la persona con discapacidad, donde se abarca el entrenamiento de habilidades cognitivas aplicado al aprendizaje de vocabulario y secuencias temporales de tareas laborales, además, cuenta con una sección que permite realizar un seguimiento al estudiante.

La aplicación es solo para uso en modo *Landscape* (pantalla horizontal) y se recomienda que se use en dispositivos de más de 7 pulgadas para una mejor experiencia por parte del usuario.

2.3.2 Requerimientos

Los requerimientos para realizar la aplicación fueron generados por el *development team* en conjunto con el *Product Owner*, los cuales son representados mediante historias de usuario épicas obtenidas a partir de entrevistas con un experto en la temática y con ayuda de la metodología iPlus, que actualmente se encuentra en fase experimental propuesta por MSc. Mayra Carrión [50]. Tales historias épicas posteriormente se desglosaron en historias de usuario para especificar los requerimientos del producto con más detalle.

iPlus

El aplicativo desarrollado es un software educativo y como tal requiere de un objetivo pedagógico. Scrum no especifica una manera de cómo definir dicho objetivo o como levantar información relevante para desarrollar una aplicación de esta índole, razón por la cual se utilizó la metodología iPlus, por cuanto la misma se enfoca en el levantamiento de requerimientos para desarrollar juegos serios. Por lo tanto, iPlus se usó fundamentalmente para obtener información relevante y definir el objetivo pedagógico de la aplicación.

iPlus fue usada junto con las metodologías Scrum y XP. En la figura 3 se muestra como iPlus se integró con estos enfoques ágiles.

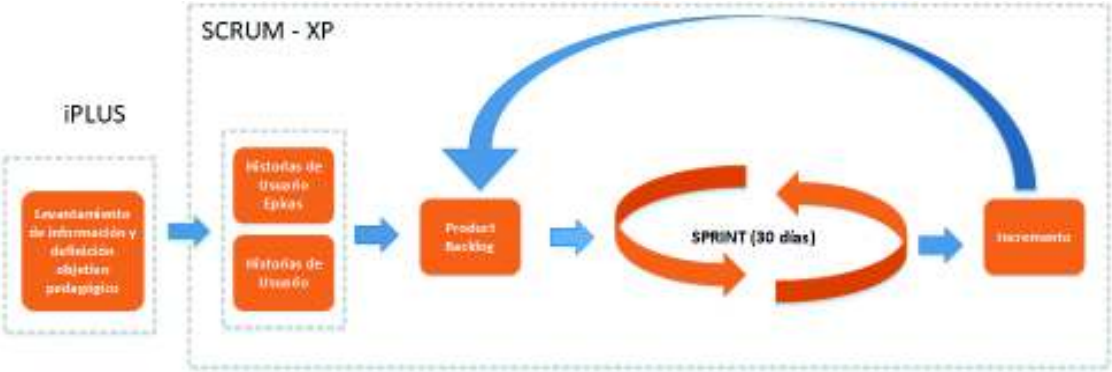


Figura 4: Metodología iPlus junto con Scrum y XP (elaborado por los autores)

iPlus se utilizó fundamentalmente para obtener información relevante y definir el objetivo pedagógico de la aplicación, pues al tratarse de un software educativo se necesitó una manera de definir dicho objetivo y obtener información más específica en este ámbito, siendo que la metodología Scrum no especifica una manera en concreto para levantar requerimientos de este tipo [50].

El protocolo experimental consta de las siguientes actividades:

Actividad 1: Establecer el contexto, donde los participantes son introducidos al protocolo experimental.

Actividad 2: Realizar la entrevista al experto en la temática.

Actividad 3: Definir los objetivos pedagógicos.

Actividad 4: Establecer género del juego serio.

Actividad 5: Diseñar bosquejos del juego en base a los géneros escogidos y priorizar las ideas.

Actividad 6: Analizar las ideas priorizadas el experto en la temática.

Actividad 7: Establecer mecánicas lúdicas y funcionalidades a ser implementadas en el juego serio.

Actividad 8: Generar el nombre de la aplicación.

El protocolo experimental que se siguió y los resultados obtenidos se encuentran especificados en el anexo A.2 y anexo A.3, respectivamente.

Al tratarse de un protocolo experimental se obtuvieron datos que fueron filtrados para obtener los más relevantes y útiles en el desarrollo de la aplicación, por tanto, el resto fueron descartados por no aportar información útil. En este sentido, de la información filtrada y analizada se pudo extraer:

- El objetivo pedagógico: apoyar el proceso de aprendizaje y desarrollo de funciones cognitivas de personas dentro de una diversidad funcional, mediante la TIC que permita lograr una sistematización de la evolución e interiorización de los mismos. Ver anexo A.4.
- La aplicación constará de dos roles principales que son:
    - **Tutor**: Se define como la persona que está a cargo o supervisa uno o más estudiantes; esta persona tiene la responsabilidad de realizar las siguientes tareas en el aplicativo:
        - Crear los talleres

- Crear el vocabulario específico de cada taller.
- Crear las secuencias de cada taller.
- Seleccionar al estudiante a su cargo para realizar el entrenamiento.
- Revisar el progreso de cada estudiante.

- **Estudiante:** Se define como la persona que realizará el entrenamiento en la aplicación, bien sea una persona con discapacidad intelectual o no. Esta persona es responsable de realizar las siguientes tareas en el aplicativo:
  - Revisar el vocabulario de cada taller.
  - Resolver las secuencias temporales de cada taller.

2.3.3 Diseño de la aplicación

Previo al desarrollo de la aplicación, se diseñó una maqueta inicial (mockup) con el fin de tener una concepción inicial de la misma, donde se muestran los requerimientos previamente obtenidos.

Se realizó el mockup con las necesidades planteadas, y siguiendo los criterios de usabilidad de las 10 heurísticas de Nielsen, para identificar posibles problemas de usabilidad. Las 10 heurísticas o reglas son claves para poder facilitar el uso de cualquier persona, no solo de sitios web, sino en cualquier sistema de información [51]. En la figura 4 se muestra la pantalla inicial del mockup en el anexo B.1 se encuentra el mockup inicial completo.

Figura 5: Pantalla inicial mockup

2.3.4 Evaluación heurística

A esta versión inicial del mockup se le realizo una evaluación heurística, la cual se define como un método de inspección de la usabilidad sin usuarios, que consiste en examinar la

calidad de una interfaz por parte de varios evaluadores expertos en el tema, siendo el principal objetivo de este tipo de evaluación medir la calidad de la interfaz del aplicativo en relación a su facilidad para ser aprendido y usado por primera vez [52].

Considerando que la propuesta de Nielsen para usabilidad fue realizada hace mucho tiempo y a lo largo de los últimos años han aparecido varias modificaciones por parte de varios autores, se tomó como referencia las 14 heurísticas propuestas por Costin Pribeanu como criterio de usabilidad, debido a que Pribeanu agrupa las heurísticas de varios autores y las sintetiza, extrayendo de manera general las más relevantes [53]. Por lo que se consideró el trabajo de Pribeanu para realizar la evaluación heurística.

Las actividades que se realizaron en la evaluación fueron las siguientes:

- Realizar cuatro documentos:
    - Protocolo de evaluación (anexo B.1)
    - Instrucciones de usuario (anexo B.2)
    - Mockup (anexo B.3)
    - Formulario de evaluación (anexo B.4)
- Enviar los documentos a los evaluadores.
- Analizar los resultados y realizar los respectivos cambios que sean sugeridos.

Para el análisis de los resultados se dividió el flujo de trabajo de la aplicación en tareas (cada una contiene un conjunto de interfaces), distinguiendo tareas del tutor y del estudiante. Las tareas del tutor consisten en: registro en la aplicación, creación de talleres y registro de estudiantes, selección del estudiante para realizar el entrenamiento y acceso a la evaluación; y, las tareas del estudiante son: revisión del vocabulario y realizar el ejercicio.

En la figura 5, se muestra un resumen de los resultados obtenidos por parte de los tres evaluadores en una tarea, donde las heurísticas se encuentran separadas en 4 grupos: Guía al usuario, esfuerzo de usuario, control y libertad del usuario y soporte, como señala Pribeanu.

En el grupo de heurísticas “guía al usuario” el total de problemas es 13 y el total de severidades 60. Para la severidad se ha seleccionada la escala 0-4 y N/A donde 0 implica que no existe problema y 4 que es un problema grave. En la severidad 0 el total es de 42, en la severidad 1 el total es 9, en la severidad 2 el total es 3, en la severidad 3 el total es 1 y en la severidad 4 es 0; esto según lo que han seleccionado los expertos para el grupo de interfaces de la tarea que realiza el tutor. Lo que implica que el mayor número de problemas se encuentran en la escala 1 y 2, pues la escala 0 y N/A no son consideradas. Este proceso se utilizó para todos los grupos de heurísticas.

En síntesis, la evaluación de esta tarea dio como resultado que el mayor número de problemas tienen severidad 1 y 2, lo cual según la escala que se consideró son problemas estéticos y menores, es decir, son problemas de baja prioridad. En el anexo B.5 se encuentran los resultados de todas las tareas del tutor y del estudiante.

Heuristics			Severity						Total Problems	All severities
Group	H0	Description	0	1	2	3	4	N/A		
User guidance	H1	Prompting	11	2	1	0	0	0	3	15
	H2	Feedback	9	5	1	0	0	0	6	15
	H3	Information architecture	12	2	1	0	0	0	3	15
	H4	Grouping/distinction	9	0	0	1	0	5	1	15
Total - User guidance			42	9	3	1	0	5	13	60
User effort	H5	Consistency	10	1	0	0	0	0	1	15
	H6	Cognitive workload	10	4	0	1	0	0	5	15
	H7	Minimal actions	12	3	0	0	0	0	3	15
Total - User effort			36	8	0	1	0	0	9	45
User control and freedom	H8	Explicit user actions	13	1	0	1	0	0	2	15
	H9	User control	9	3	0	1	0	2	4	15
	H10	Flexibility	8	3	3	0	0	1	4	15
Total - User control and freedom			30	7	3	3	0	3	10	45
User support	H11	Compatibility with the user	10	2	1	2	1	1	6	15
	H12	Task guidance and support	8	5	0	1	0	0	7	15
	H13	Error management	8	4	2	1	0	1	8	15
	H14	Help and documentation	9	2	1	1	0	1	5	15
Total - User support			35	13	4	6	1	3	26	60
Total TTI			143	38	9	11	1	13	58	208

Figura 6: Resultado de la evaluación heurística

Una vez finalizada la evaluación se obtuvo la versión final del mockup (anexo B.6).

Es importante mencionar que los problemas de usabilidad encontrados fueron solucionados en la etapa de desarrollo de la aplicación. Las interfaces se mostrarán en el desarrollo de cada sprint y tal evaluación ayudó a realizar de una manera más ágil el desarrollo de la aplicación.

2.3.5 Definición de módulos de la aplicación

Los módulos con los cuales se definió la aplicación fueron:

- Módulo de Registro: En este módulo el tutor se registra ingresando su información y se conecta al sistema. La sesión del tutor permanece activa hasta que decide desconectarse. El tutor tiene la tarea de crear a los estudiantes que utilizarán la aplicación, siendo también posible crear, editar o eliminar alumnos de la aplicación. Los estudiantes serán visibles solo para el tutor que los crea.

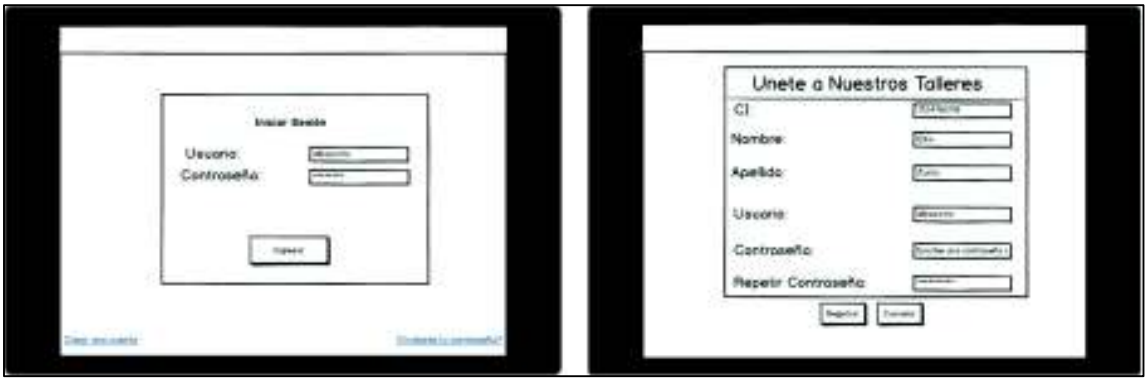


Figura 7: Interfaces que representan el módulo de registro

- Personalización de talleres: En este módulo, el tutor puede crear, editar y ver talleres con el contenido que se presentará en el módulo de entrenamiento. Cada taller registrado tiene su propio vocabulario y secuencias, por lo que el tutor crea y edita el contenido. Para crear secuencias, el tutor debe primero crear una historia, es decir, una descripción de la actividad que se realiza en el taller. Una vez realizado este paso, el tutor está habilitado para crear la secuencia del taller.

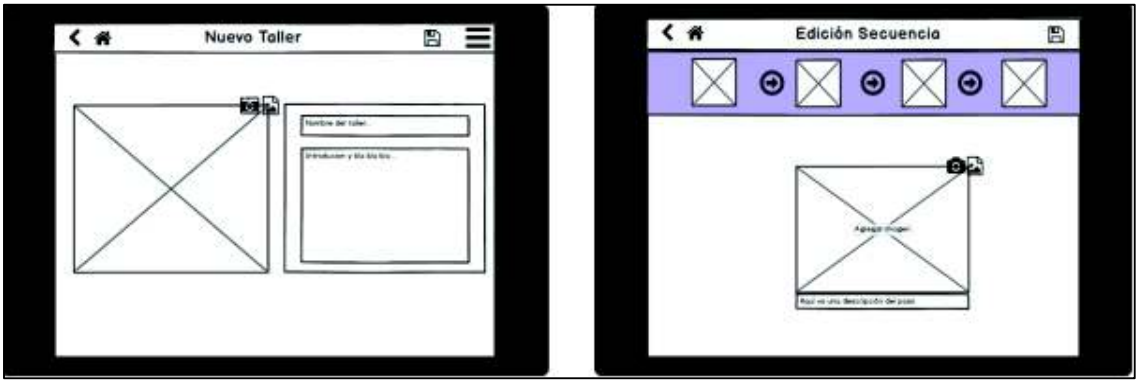


Figura 8: Interfaces que representan el módulo de personalización de talleres

- Módulo de Entrenamiento: En este módulo, el estudiante entrena sus habilidades cognitivas ordenando secuencias temporales, se presenta el vocabulario y las secuencias asociadas con cada taller creadas en el módulo de personalización. El tutor en este módulo ayuda al alumno, selecciona el taller y la dificultad del mismo. Además, él se encuentra a cargo de hacer observaciones en cada ejercicio que el estudiante realiza, de ser necesario.

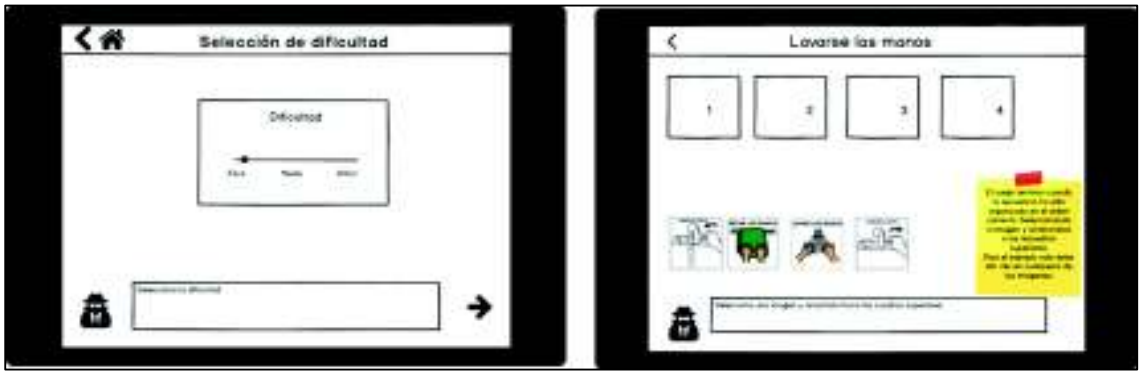


Figura 9: Interfaces que representan el módulo de entrenamiento

- Módulo de Seguimiento y Evaluación: En este módulo, el tutor puede revisar los resultados obtenidos por cada estudiante en el entrenamiento a través de un gráfico, en el que se visualiza el tiempo, los éxitos y fracasos de los ejercicios realizados. Cada gráfico se puede exportar a Excel y los datos también se pueden mostrar como una tabla.

Figura 10: Interfaces que representan el módulo de seguimiento y evaluación

2.3.6 Diagrama de Navegación de la aplicación

La figura 10 representa el diagrama de navegación de la aplicación, en donde se muestra una aproximación de las secciones y actividades que tiene la misma, el diagrama separa secciones con circunferencias y actividades por recuadros, asimismo, diferencia las funciones de los tipos de usuarios por colores.

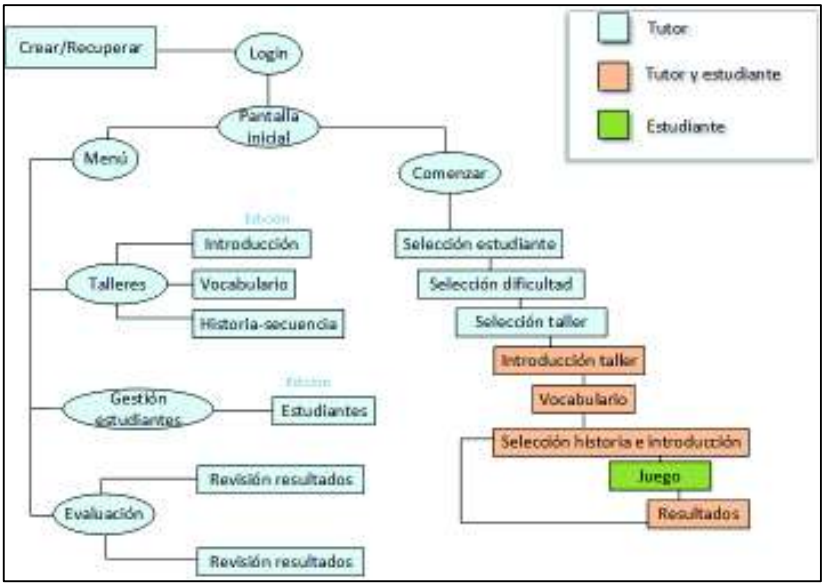


Figura 11: Diagrama de navegación (elaborado por los autores)

2.3.7 Historias de usuario épicas

Las tablas 2, 3, 4, 5 y 6 muestran las historias de usuario épicas que se definieron a partir de los requerimientos planteados.

Historia de usuario épica	**HUE01**
Título: Desarrollar Módulo de Navegación	
Descripción: Como tutor requiero que la aplicación contenga una interfaz que facilite la navegación entre los distintos módulos de la aplicación.	

Tabla 2: Historia de usuario épica HUE01

Historia de usuario épica	**HUE02**
Título: Desarrollar módulo de creación y personalización de talleres	
Descripción: Como tutor requiero que mi aplicación me permita la creación y personalización de escenarios de tareas laborales y vocabulario.	

Tabla 3: Historia de usuario épica HUE02.

Historia de usuario épica	HUE03
Título: Desarrollar Módulo de Seguimiento y Evaluación	
Descripción: Como tutor requiero que la aplicación me permita realizar el seguimiento de los estudiantes en función de los datos históricos.	

Tabla 4: Historia de usuario épica HUE03.

Historia de usuario épica	HUE04
Título: Desarrollar Módulo de Entrenamiento	
Descripción: Como tutor requiero que la aplicación tenga un módulo de entrenamiento el cual va estar compuesto de dos partes, vocabulario y secuencias temporales de tareas laborales.	

Tabla 5: Historia de usuario épica HUE04.

Historia de usuario épica	HUE05
Título: Desarrollar Módulo de Registro	
Descripción: Como tutor requiero que la aplicación tenga un módulo de registro de estudiantes para cada tutor por consiguiente la aplicación debe permitir registrar nuevos tutores.	

Tabla 6: Historia de usuario épica HUE05.

2.3.8 Product Backlog

Una vez concluida la fase de diseño del aplicativo, las historias de usuario épicas se subdividieron en historias de usuarios más específicas generando el siguiente *product backlog.*

Product Backlog				
HU Épica	**Código**	**Historia de usuario**	**Prioridad**	**Duración (días)**
HUE01	**HU01**	Desarrollar interfaz bienvenida de la aplicación.	Alta	4
	HU02	Desarrollar el menú de la interfaz principal de la aplicación.	Alta	2
HUE02	**HU03**	Desarrollar interfaz para ingresar a la aplicación.	Alta	2
	HU04	Desarrollar interfaz para registro de tutores.	Alta	3
	HU05	Desarrollar interfaz para cambiar contraseña.	Alta	2
	HU06	Desarrollar interfaz de estudiantes.	Alta	2
	HU07	Desarrollar interfaz para añadir o editar estudiante.	Alta	2
HUE03	**HU08**	Desarrollar interfaz de talleres.	Alta	1
	HU09	Desarrollar interfaz para añadir o editar taller.	Alta	2
	HU10	Desarrollar interfaz de Vocabulario.	Alta	1
	HU11	Desarrollar interfaz para añadir o editar vocabulario.	Alta	2
	HU12	Desarrollar interfaz de Historias.	Alta	1
	HU13	Desarrollar interfaz para añadir o editar historia.	Alta	2
	HU14	Desarrollar interfaz de Secuencias.	Alta	1
	HU15	Desarrollar interfaz editar secuencia.	Alta	3
HUE04	**HU16**	Desarrollar interfaz de entrenamiento.	Alta	1
	HU17	Desarrollar una pantalla de confirmación.	Alta	1
	HU18	Desarrollar interfaz de selección de dificultad.	Alta	1
	HU19	Desarrollar interfaz de selección de taller.	Alta	2

	HU20	Desarrollar interfaz de bienvenida al taller.	Alta	1
	HU21	Desarrollar interfaz de Vocabulario.	Alta	1
	HU22	Desarrollar interfaz de Selección de Historia.	Alta	1
	HU23	Desarrollar pantalla para seleccionar el tiempo.	Alta	1
	HU24	Desarrollar interfaz de secuencia (ejercicio).	Alta	5
	H25	Desarrollar interfaz de resultado del ejercicio	Alta	2
HUE05	**HU26**	Desarrollar interfaz de evaluación.	Alta	2
	HU27	Desarrollar interfaz de gráfica estudiante.	Alta	5
	HU28	Desarrollar interfaz de tabla estudiante.	Alta	3

Tabla 7: Product Backlog.

2.3.9 Release Planning

Una vez definido el *product backlog* se procede a definir el *release planning*, el cual define el número de sprints que tendrá el desarrollo de la aplicación y cuales historias de usuario serán abarcadas en cada sprint, teniendo una duración aproximada de dos semanas cada uno.

Sprint 1	**Sprint 2**	**Sprint 3**	**Sprint 4**	**Sprint 5**
HU01	HU03	HU08	HU16	HU26
HU02	HU04	HU09	HU17	HU27
	HU05	HU10	HU18	HU28
	HU06	HU11	HU19	
	HU07	HU12	HU20	
		HU13	HU21	
		HU14	HU22	
		HU15	HU23	
			HU24	
			HU25	

Tabla 8: Release Planning

2.3.10 Roles

Para el desarrollo del presente proyecto se identificaron los siguientes roles que se muestran en la tabla 9.

Nombre	Rol
PhD. Marco Santórum G.	Scrum Master
Fundación FINE - Yolanda Ortiz Carranco MSc.	Product Owner
Carlos Eduardo Sampedro Mora Andrés David Paredes Ruano	Scrum Team

Tabla 9: Roles Scrum

2.4 Ejecución de Sprints

2.4.1 Sprint 1

Sprint Planning

Sprint Goal

Implementar la pantalla de bienvenida y el menú de navegación de la aplicación.

Historias de Usuario del Sprint

Las tablas 10 y 11 muestran las historias de usuario del primer Sprint.

Historia de usuario	HU01
Título: Desarrollar interfaz bienvenida de la aplicación.	
Descripción: Como tutor, requiero que mi aplicación contenga una interfaz de bienvenida y que contenga un menú para navegar a los distintos módulos.	
Prioridad: Alta	**Esfuerzo:** 4 días
Criterios de aceptación: La interfaz debe contener: • Un botón que permita comenzar con el entrenamiento. • Un menú visible para la navegación.	

• Un texto que dé la bienvenida al usuario

Tabla 10: Historia de usuario 01 - Sprint 1.

Historia de usuario	HU02
Título: Desarrollar el menú de la interfaz principal de la aplicación.	
Descripción: Como tutor, requiero que mi aplicación contenga un menú para navegar a los distintos módulos.	
Prioridad: Alta	**Esfuerzo:** 2 días
Criterios de aceptación: El menú debe contener: • Un botón que permita dirigirse a la configuración de talleres. • Un botón que permita dirigirse al registro de estudiantes. • Un botón que permita dirigirse a la evaluación de estudiantes. • Un botón que permita salir de la sesión.	

Tabla 11: Historia de usuario 02 - Sprint 1.

Sprint Backlog

La tabla 12 muestra el *Sprint Backlog* con las tareas que se ejecutaron en este Sprint.

ID	Historia de Usuario	Criterios de Aceptación	Actividades
HU01	Desarrollar interfaz bienvenida de la aplicación.	La interfaz debe contener: ● Un campo de texto para ingresar usuario. ● Un campo de texto para ingresar contraseña. ● Un botón para iniciar la sesión.	Diseñar la interfaz en base al mockup con los siguientes campos: ● Un campo de texto para ingresar usuario. ● Un campo de texto para ingresar contraseña. ● Un botón para iniciar la sesión. Implementar la interfaz con las especificaciones anteriores.
		Un menú visible para la navegación.	Diseñar el menú en base al mockup y debe contener las opciones: ● Mis talleres ● Estudiantes ● Evaluación ● Salida Implementar un método que permita la navegación en el menú.
		Un texto que dé la bienvenida al usuario	Añadir un campo de texto que dé la bienvenida al usuario.

HU02	Desarrollar el menú de la interfaz principal de la aplicación.	Un botón que permita dirigirse a talleres.	Implementar un método que permita dirigirse a la interfaz de talleres.
		Un botón que permita dirigirse al registro de estudiantes.	Implementar un método que permita dirigirse a la interfaz de estudiantes.
		Un botón que permita dirigirse a la evaluación de estudiantes.	Implementar un método que permita dirigirse a la interfaz de evaluación.
		Un botón que permita salir de la sesión.	Implementar un método que permita salir de la sesión.

Tabla 12: Sprint Backlog - Sprint 1.

Sprint Review

Al finalizar el sprint se obtuvo un incremento que cumple con los criterios de aceptación mostrados en las historias de usuario, el cual consistió en implementar la pantalla de bienvenida a la aplicación y el menú de navegación de ésta.

Es relevante mencionar que se cumplieron todas las tareas del sprint y no se realizaron cambios en el Sprint Backlog. En consecuencia, las historias de usuario que se ejecutaron en el siguiente sprint son: HU03 hasta HU07

Pruebas de aceptación

Las tablas 13 y 14 muestran las pruebas de aceptación de cada historia de usuario de este Sprint.

Código	Criterios de aceptación	Completo
HU01	Un botón que permita comenzar con el entrenamiento.	SI
	Un menú visible para la navegación.	SI
	Un texto que dé la bienvenida al usuario.	SI

Tabla 13: Pruebas de aceptación - Historia de usuario 01.

Código	Criterios de aceptación	Completo
HU02	Un botón que permita dirigirse a la configuración de talleres.	SI
	Un botón que permita dirigirse al registro de estudiantes.	SI
	Un botón que permita dirigirse a la evaluación de estudiantes.	SI
	Un botón que permita salir de la sesión.	SI

Tabla 14: Pruebas de aceptación - Historia de usuario 02.

A continuación, se muestran algunas de las interfaces que fueron implementadas en el sprint, tanto el mockup como la implementación final.

Figura 12: Pantalla inicial, mockup vs interfaz desarrollada

Figura 13: Menu de la aplicación, mockup vs interfaz desarrollada

Spring Retrospective

Al ejecutar la aplicación en un dispositivo móvil se evidenció que la pantalla principal de la aplicación funciona correctamente al igual que el menú de navegación de ésta. El sprint se realizó sin contratiempos y se terminó en el tiempo esperado. En la figura 13 denominada *Burndown chart* se muestra cómo se desarrolló el desarrollo a lo largo del tiempo, destacando que el trabajo remanente se muestra en el eje vertical y el tiempo en el eje horizontal, siendo una serie temporal del trabajo pendiente [54].

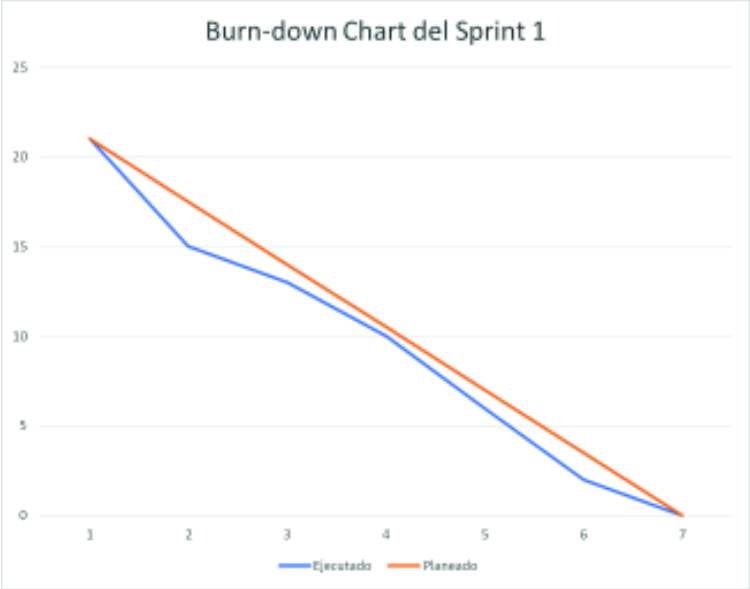


Figura 14: Burndown Chart – Primer Sprint

2.4.2 Sprint 2

Sprint Planning

Sprint Goal

Se refiere a la implementación de las interfaces para el ingreso a la aplicación, registro de estudiantes y tutores.

Historias de Usuario del Sprint

Las tablas 15, 16, 17, 18 y 19 muestran las historias de usuario a desarrollar en este Sprint.

Historia de usuario	HU03
Título: Desarrollar interfaz para ingresar a la aplicación.	
Descripción: Como tutor, requiero que mi aplicación contenga una interfaz que me permite iniciar sesión en la aplicación mediante un usuario y contraseña.	
Prioridad: Alta	**Esfuerzo:** 2 días
Criterios de aceptación: La interfaz debe contener: • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un botón para iniciar la sesión. • Si los datos ingresados son correctos permitirá el acceso a la aplicación. • Una opción para registrar un nuevo tutor. • Una opción para cambiar la contraseña.	

Tabla 15: Historia de usuario 03 - Sprint 2.

Historia de usuario	HU04
Título: Desarrollar interfaz para registro de tutores.	
Descripción: Como tutor, requiero que mi aplicación contenga una interfaz que me permita registrar en la aplicación.	
Prioridad: Alta	**Esfuerzo:** 3 días
Criterios de aceptación: La interfaz debe contener: • Un campo de texto para ingresar el documento de identidad. • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un campo de texto para repetir contraseña. • Un botón que permita registrar al tutor. • Una validación para evitar tener el mismo tutor registrado. • Un botón para cancelar el proceso.	

Tabla 16: Historia de usuario 04 - Sprint 2.

Historia de usuario	HU05
Título: Desarrollar interfaz para cambiar contraseña.	
Descripción: Como tutor, requiero que mi aplicación contenga una interfaz que me permita modificar mi usuario y contraseña.	

Prioridad: Alta	**Esfuerzo:** 2 días
Criterios de aceptación: La interfaz debe contener: • Un campo de texto para ingresar el documento de identidad. • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un campo de texto para repetir contraseña. • Un botón que permita cambiar los datos. • Una validación para evitar tener el mismo tutor registrado. Un botón para cancelar el proceso.	

Tabla 17: Historia de usuario 05 - Sprint 2.

Historia de usuario	**HU06**
Título: Desarrollar interfaz de estudiantes.	
Descripción: Como tutor, requiero que la aplicación contenga una interfaz que visualice a los estudiantes registrados, además que permita añadir, editar o eliminar.	
Prioridad: Alta	**Esfuerzo:** 2 días
Criterios de aceptación: La interfaz debe contener: • Una lista de los estudiantes registrados. • Un botón que permita registrar nuevos estudiantes. • Un botón que permita la opción de eliminar uno o varios estudiantes. • Un botón que permita volver a la pantalla principal de la aplicación.	

- La opción de seleccionar un estudiante de la lista para su edición.
- La opción de seleccionar un estudiante de la lista y redirigir a la interfaz de añadir o editar estudiante.

Tabla 18: Historia de usuario 06 - Sprint 2.

Historia de usuario	**HU07**
Título: Desarrollar interfaz para añadir o editar estudiante.	
Descripción: Como tutor, requiero que para añadir o editar un estudiante cuente con una pantalla para ingresar los datos del mismo.	
Prioridad: Alta	**Esfuerzo:** 2 días
Criterios de aceptación: La interfaz debe contener: • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar edad. • Una sección para seleccionar el género. • Un campo de texto para ingresar el perfil. • La opción para tomar o elegir una foto de la galería. • Un botón para guardar los cambios realizados. • La opción regresar a la pantalla anterior. • La opción para regresar a la pantalla principal.	

Tabla 19: Historia de usuario 04 - Sprint 2.

Sprint Backlog

La tabla 20 contiene el Sprint Backlog con las tareas realizadas en este Sprint.

ID	Historia de Usuario	Criterios de Aceptación	Actividades
HU03	Desarrollar interfaz para ingresar a la aplicación	La interfaz debe contener: • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un botón para iniciar la sesión. Si los datos ingresados son correctos permitirá el acceso a la aplicación. Una opción para registrar un nuevo tutor. Una opción cambiar la contraseña.	Implementar la interfaz siguiendo el diseño del mockup. Implementar la validación de usuarios registrados en los botones. Implementar un método que permita ir a la interfaz de registro tutor. Implementar un método que permita ir a la interfaz de cambio de contraseña.

HU04	Desarrollar interfaz para registro de tutores	La interfaz debe contener: • Un campo de texto para ingresar el documento de identidad. • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un campo de texto para repetir contraseña. • Un botón que permita registrar al tutor. • Un botón para cancelar el proceso. Una validación para evitar tener el mismo tutor registrado.	Implementar la interfaz siguiendo el diseño del mockup. Implementar un método para validar el registro de tutores. Implementar un método para cancelar la operación.

HU05	Desarrollar interfaz para cambiar contraseña	La interfaz debe contener: • Un campo de texto para ingresar el documento de identidad. • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar usuario. • Un campo de texto para ingresar contraseña. • Un campo de texto para repetir contraseña. • Un botón que permita registrar al tutor. • Un botón para cancelar el proceso. Una validación para evitar tener el mismo tutor registrado.	Implementar interfaz siguiendo el diseño del mockup. Implementar un método que permitir validar la actualización de los datos del tutor. Implementar un método para cancelar la operación.
HU06	Desarrollar interfaz de estudiantes	La interfaz debe contener: • Una lista de los estudiantes registrados. • Un botón que permita registrar nuevos estudiantes • Un botón que permita la opción de eliminar uno o varios estudiantes • Un botón que permita volver a la pantalla principal de la aplicación.	Implementar interfaz siguiendo el diseño del mockup. Implementar un método que permita redirigir a la interfaz de añadir o editar estudiante. Implementar un método para borrar los estudiantes

		La opción de seleccionar un estudiante de la lista y redirigir a la interfaz de añadir o editar estudiante. La opción de buscar un estudiante dentro de la lista	Implementar un método que permita la búsqueda de un estudiante basándose en el nombre.
HU07	Desarrollar interfaz para añadir o editar estudiante	La interfaz debe contener: • Un campo de texto para ingresar nombre. • Un campo de texto para ingresar apellido. • Un campo de texto para ingresar edad. • Una sección para seleccionar el género. • Un campo de texto para ingresar el perfil. • La opción para tomar o elegir una foto de la galería. • Un botón para guardar los cambios realizados. • La opción regresar a la pantalla anterior. • La opción para regresar a la pantalla principal.	Implementar interfaz siguiendo el diseño del mockup. Implementar un método para validar el registro de estudiantes. Implementar método para cancelar tarea Implementar un método para acceder a la cámara o galería del dispositivo móvil.

Tabla 20: Sprint Backlog - Sprint 2.

Sprint Review

Al finalizar el sprint se obtuvo un incremento que cumple con los criterios de aceptación mostrados en las historias de usuario, que consistió en implementar las pantallas de bienvenida a la aplicación y el menú de navegación de ésta ingresó a la aplicación, al registro de tutores y al cambio de contraseña. También, se cumplieron todas las tareas del sprint y se efectuaron cambios en el Sprint Backlog, señalando que el único cambio que se realizó en el sprint fue añadir la funcionalidad de buscar estudiantes en la lista, lo cual se realizó sin contratiempos. En consecuencia, las historias de usuario ejecutadas en el siguiente sprint fueron: HU08 hasta HU1.

Pruebas de aceptación

Las tablas 21, 22, 23, 24 y 25 contienen las pruebas de aceptación respectivas para cada historia de usuario realizada en este Sprint.

Código	Criterios de aceptación	Completo
HU03	La interfaz debe contener: ● Un campo de texto para ingresar usuario. ● Un campo de texto para ingresar contraseña. ● Un botón para iniciar la sesión.	SI
	Una opción cambiar la contraseña.	SI
	Si los datos ingresados son correctos permitirá el acceso a la aplicación.	SI
	Una opción para registrar un nuevo tutor.	SI

Tabla 21: Pruebas de aceptación - Historia de usuario 03.

Código	Criterios de aceptación	Completo
HU04	La interfaz debe contener: ● Un campo de texto para ingresar el documento de identidad. ● Un campo de texto para ingresar nombre. ● Un campo de texto para ingresar apellido. ● Un campo de texto para ingresar usuario. ● Un campo de texto para ingresar contraseña. ● Un campo de texto para repetir contraseña. ● Un botón que permita registrar al tutor. ● Un botón para cancelar el proceso.	SI
	Una validación para evitar tener el mismo tutor registrado.	SI

Tabla 22: Prueba de aceptación - Historia de usuario 04.

Código	Criterios de aceptación	Completo
HU05	La interfaz debe contener: ● Un campo de texto para ingresar el documento de identidad. ● Un campo de texto para ingresar nombre. ● Un campo de texto para ingresar apellido. ● Un campo de texto para ingresar usuario. ● Un campo de texto para ingresar contraseña. ● Un campo de texto para repetir contraseña. ● Un botón que permita registrar al tutor. ● Un botón para cancelar el proceso.	SI
	Una validación para evitar tener el mismo tutor registrado.	SI

Tabla 23: Pruebas de aceptación - Historia de usuario 05.

Código	Criterios de aceptación	Completo
HU06	La interfaz debe contener: ● Una lista de los estudiantes registrados. ● Un botón que permita registrar nuevos estudiantes. ● Un botón que permita la opción de eliminar uno o varios estudiantes. ● Un botón que permita volver a la pantalla principal de la aplicación.	SI
	La opción de seleccionar un estudiante de la lista y redirigir a la interfaz de añadir o editar estudiante.	SI
	La opción de buscar un estudiante dentro de la lista.	SI

Tabla 24: Pruebas de aceptación - Historia de usuario 06.

Código	Criterios de aceptación	Completo
HU07	La interfaz debe contener: ● Un campo de texto para ingresar nombre. ● Un campo de texto para ingresar apellido. ● Un campo de texto para ingresar edad. ● Una sección para seleccionar el género. ● Un campo de texto para ingresar el perfil. ● La opción para tomar o elegir una foto de la galería. ● Un botón para guardar los cambios realizados. ● La opción para regresar a la pantalla anterior. ● La opción para regresar a la pantalla principal.	SI

Tabla 25: Pruebas de aceptación - Historia de usuario 07.

A continuación, se muestran algunas de las interfaces que fueron implementadas en el sprint, tanto en el mockup como en la implementación final.

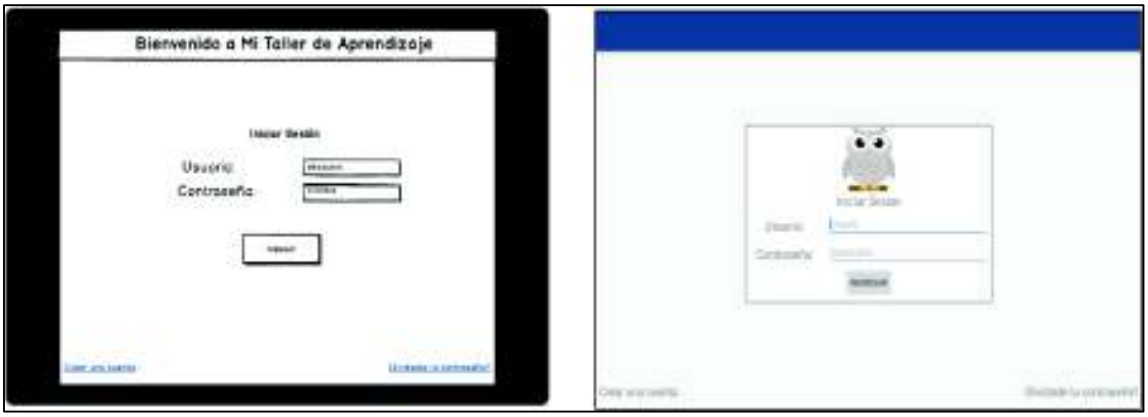


Figura 15: Pantalla login, mockup vs interfaz desarrollada

Figura 16: Registro tutor, mockup vs interfaz desarrollada

Figura 17: Registro estudiante, mockup vs interfaz desarrollada

Spring Retrospective

Tras usar la aplicación en un dispositivo móvil se evidenció que al existir un número elevado de estudiantes puede llegar a ser confusa su selección para edición, por lo que se decidió

añadir una funcionalidad que permite la búsqueda por nombre. El sprint se realizó sin contratiempos y se culminó en el tiempo esperado, tal como se aprecia en la figura 17.

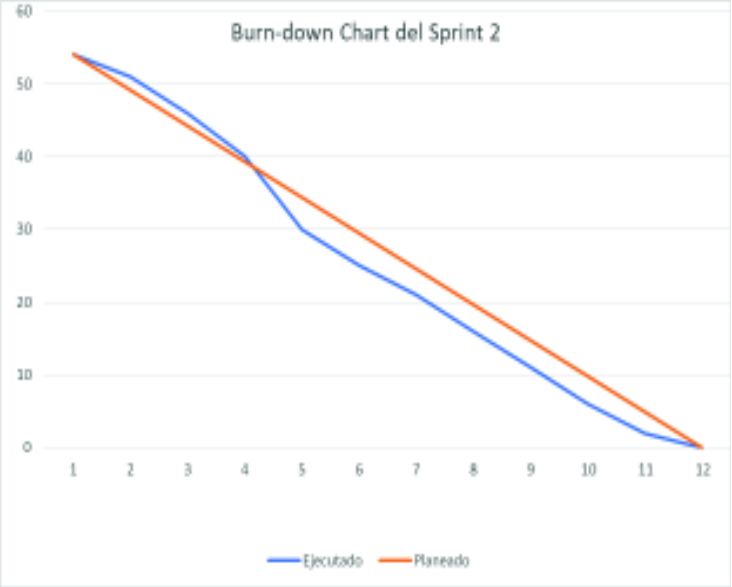


Figura 18: Burndown Chart – Segundo Sprint

Nota: A partir del Sprint 3 solo se presenta un resumen del mismo y los sprints completos se muestran en el anexo B.7.

2.4.3 Sprint 3

Sprint Goal

Implementar las interfaces que permita la creación y edición de talleres como lo especifica la aplicación.

Sprint Review

Al finalizar el sprint se obtuvo un incremento que cumple con los criterios de aceptación mostrados en las historias de usuario, que consistió en implementar las pantallas de creación y edición de talleres.

Por su parte, se cumplieron todas las tareas del sprint y no se realizaron cambios en el Sprint Backlog. En consecuencia, las historias de usuario que se ejecutaron en el siguiente sprint fueron: HU16 hasta HU25.

A continuación, se muestran algunas de las interfaces implementadas en el sprint, tanto el mockup como la implementación final.

Figura 19: Lista de talleres registrados, mockup vs interfaz desarrollada

Figura 20: Agregar/editar palabra, mockup vs interfaz desarrollada

Figura 21: Creación de secuencia, mockup vs interfaz desarrollada

Spring Retrospective

Durante el desarrollo del módulo de edición y creación de talleres se encontró que la función de borrador podría llegar a ser confusa para el usuario, provocando incluso errores durante el uso de la aplicación, por lo cual se decidió modificar el aspecto de dicha funcionalidad para evitar algún posible error por parte del usuario. El sprint se realizó sin contratiempos y se terminó en el tiempo esperado, tal como se aprecia en la figura 21.

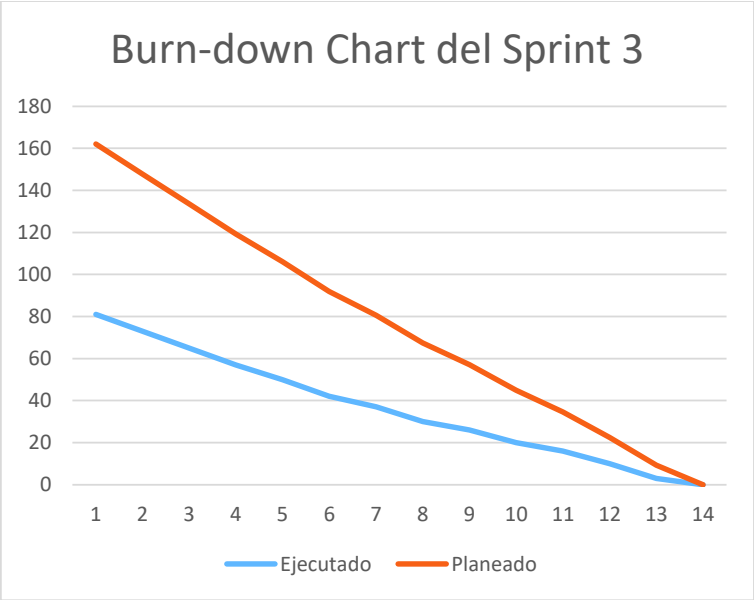


Figura 22: Burndown Chart – Tercer Sprint

2.4.4 Sprint 4

Sprint Goal

Comprende la implementación de las interfaces que permitan al tutor y al estudiante realizar el entrenamiento en las secuencias temporales y el vocabulario de los talleres que han sido previamente creados.

Sprint Review

Al finalizar el sprint se obtuvo el incremento que cumple con los criterios de aceptación mostrados en las historias de usuario, que consistió en las interfaces pertinentes para realizar el entrenamiento de habilidades cognitivas del estudiante.

Durante el desarrollo del sprint se aumentó una interfaz que permite al estudiante observar y aprender las secuencias, la cual surgió debido a la necesidad de un aprendizaje previo de las secuencias por parte del estudiante, para poder resolver posteriormente los ejercicios.

También, se cumplieron todas las tareas del sprint y no se realizaron más cambios en el Sprint Backlog. En consecuencia, las historias de usuario que se ejecutaron en el siguiente sprint fueron: HU26 hasta HU28.

A continuación, se muestran algunas de las interfaces implementadas en el sprint, tanto el mockup como la implementación final.

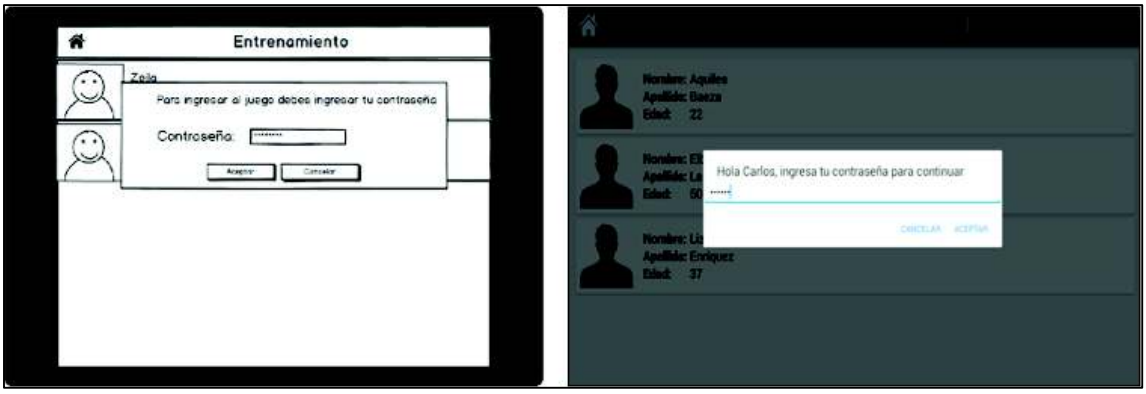


Figura 23: Pantalla entrenamiento, mockup vs interfaz desarrollada

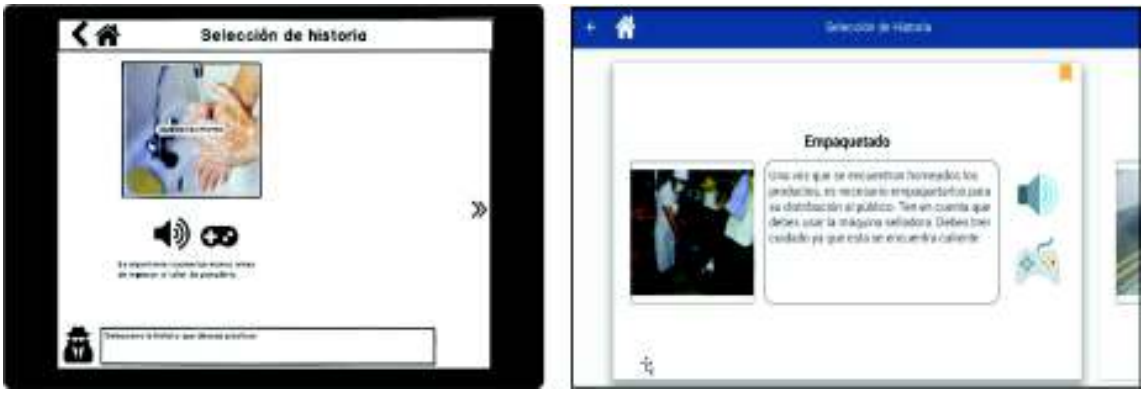


Figura 24: Selección historia, mockup vs interfaz desarrollada

Figura 25: Pantalla ordenar secuencia, mockup vs interfaz desarrollada

Figura 26: Resultados, mockup vs interfaz desarrollada

Spring Retrospective

Durante el desarrollo del módulo de entrenamiento se añadió una pantalla *popup* (ventana emergente) lo que ocasiono un leve retraso en la ejecución del sprint. Una vez realizada la pantalla *popup,* el sprint se ejecutó sin contratiempos y se terminó en el tiempo esperado, tal como se aprecia en la figura 26.

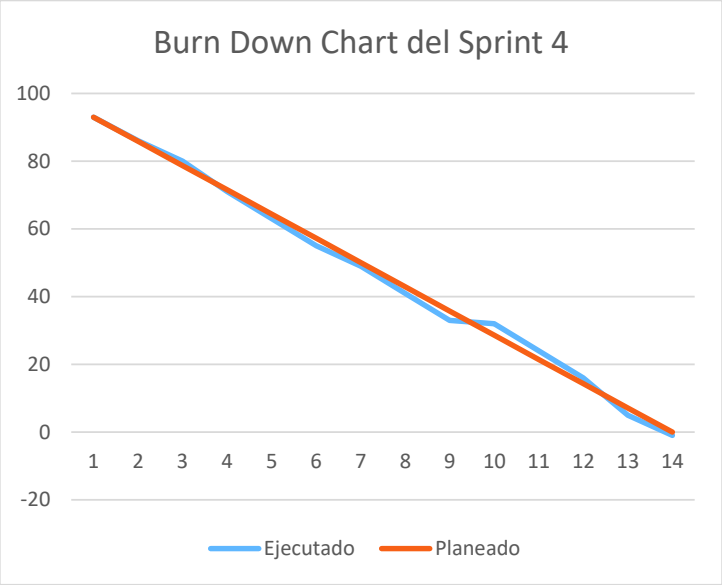


Figura 27: Burndown Chart – Cuarto sprint

2.4.5 Sprint 5

Sprint Goal

Contempla el hecho de implementar las interfaces que permita al tutor observar el desempeño, en función de los datos históricos recolectados durante los entrenamientos.

Sprint Review

Al finalizar el sprint se obtuvo el incremento que cumple con los criterios de aceptación mostrados en las historias de usuario, que consistió en el desarrollo de las interfaces para mostrar el progreso del estudiante durante los ejercicios.

En este apartado, se cumplieron todas las tareas del sprint y no se realizaron más cambios en el Sprint Backlog. En consecuencia, las historias de usuario se llevaron a cabo en su totalidad, sin dejar ninguna tarea del product backlog por realizar.

A continuación, se muestran algunas de las interfaces implementadas en el sprint, tanto el mockup como la implementación final.

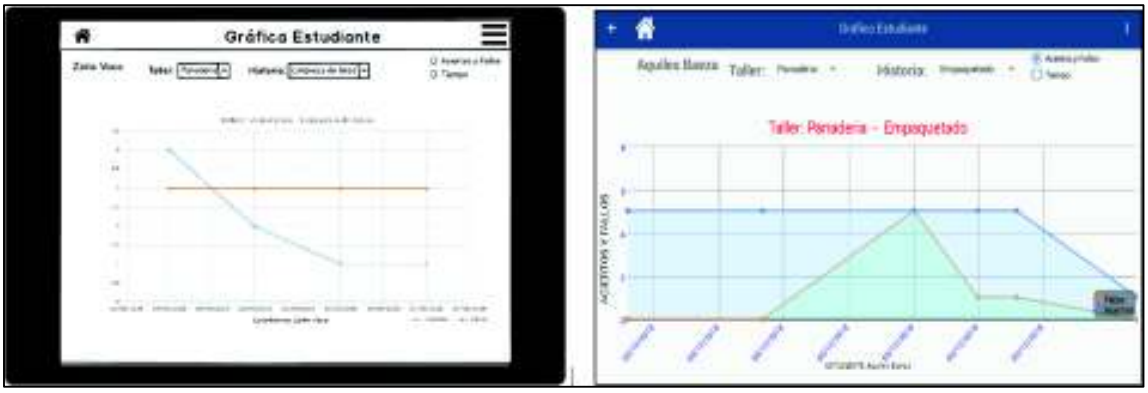


Figura 28: Gráfica estudiante, mockup vs interfaz desarrollada

Figura 29: Tabla estudiante, mockup vs interfaz desarrollada

Spring Retrospective

La implementación de este módulo se realizó sin problemas. Se puede observar en la figura 29 un leve retraso los primeros días; sin embargo, el sprint se ejecutó sin contratiempos y se terminó en el tiempo esperado.

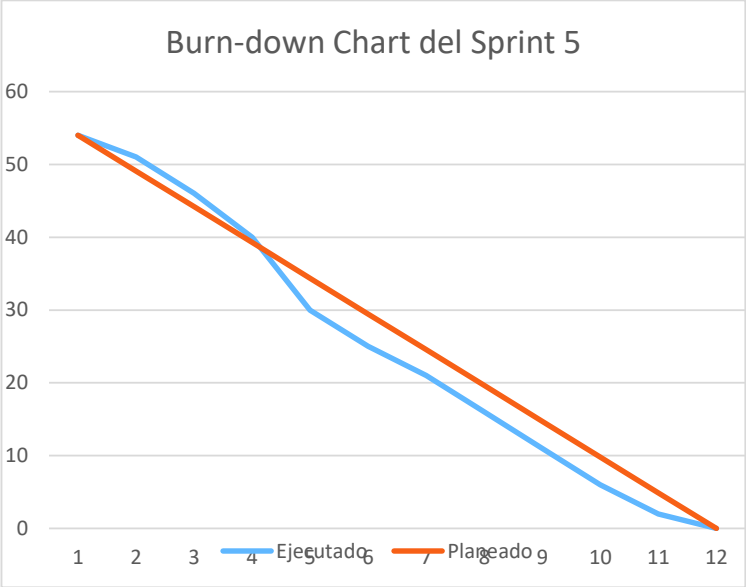


Figura 30: Burndown Chart – Quinto sprint

En este capítulo se han elegido en definitiva las metodologías agiles XP y Scrum para el desarrollo, estableciendo la arquitectura de la aplicación, la definición de las herramientas y recursos necesarios para el desarrollo, así como la descripción del caso de estudio a implementarse. También, se realizó el levantamiento de requerimientos con ayuda de la metodología iPlus. En el caso de estudio, se especificaron los módulos que contiene la aplicación y se elaboraron los sprints, según fueron planificados. En el siguiente capítulo se evalúa el caso de estudio mediante las pruebas de funcionalidad y usabilidad, así como el análisis de los resultados obtenidos.

CAPÍTULO 3

EVALUACIÓN Y RESULTADOS

3.1 Pruebas de funcionalidad

Con el propósito de conocer si el software está listo a ser liberado resultó necesario definir un criterio de aprobación, por cuanto resulta difícil obtener un código libre de errores. Una vez terminada la fase de desarrollo, se procede a realizar un conjunto de pruebas de aceptación o de funcionalidad, con el fin de verificar el correcto funcionamiento de la aplicación y señalar cuando las funcionalidades de una iteración han sido completadas exitosamente; éstas se desarrollan en un ambiente de producción que simula las características que tendrá el software cuando sea usado. Las mismas se realizan tomando en cuenta los requisitos funcionales previamente especificados en las historias de usuario. Por su parte, la metodología XP propone realizar dicha evaluación mediante el uso casos de prueba [55].

A continuación, se especifican los casos de prueba que se usaron para cada módulo. Es importante mencionar que en la columna código se tendrá ciertas letras que nombran al módulo en el cuál se encuentran por ejemplo para la descripción del primer módulo se tiene CPN que se refiere al código de pantalla de navegación.

3.1.1 Módulo de navegación

Para el módulo de navegación se han definido cuatro casos de prueba los resultados obtenidos, los cuales se muestran en la siguiente tabla donde se especifican los pasos a seguir. Los casos de prueba que fueron exitosos se encuentran representados con el ícono de visto en la columna de código y los que no con el icono de advertencia en la misma columna.

Sprint 1				
Código	**Descripción del caso**	**Prerrequisitos**	**Resultados**	
			Esperado	**Obtenido**
CPN01	Verificar la interfaz de bienvenida a la aplicación	Aplicación iniciada y login de un usuario	Se muestra la interfaz de bienvenida	Se muestra la interfaz de bienvenida
CPN02	Verificar que el botón “comenzar” inicie el entrenamiento	Aplicación iniciada y login de un usuario	Se muestra el botón comenzar	Se muestra el botón comenzar
CPN03	Verificar el acceso al menú principal de la aplicación	Aplicación iniciada y login de un usuario	Se muestra el menú principal de la aplicación	Se muestra el menú principal de la aplicación
CPN04	Verificar el funcionamiento del menú principal de la aplicación	Interfaz de menú iniciada	Se verifica el funcionamiento del menú principal de la aplicación que redirija a los diferentes módulos	Se verifica el funcionamiento del menú principal de la aplicación que redirija a los diferentes módulos

Tabla 26: Pruebas de funcionalidad - Sprint 1.

3.1.2 Módulo de registro

En cuanto al módulo de registro se han definido 8 casos de prueba, cuyos resultados obtenidos se muestran en la tabla 27.

Sprint 2				
Código	**Descripción del caso**	**Prerrequisitos**	**Resultados**	
			Esperado	**Obtenido**
CPR01	Verificar que se presente la interfaz para ingresar a la aplicación	Iniciar aplicación	Se muestra una pantalla de inicio	Se muestra una pantalla de inicio
CPR02	Verificar exista la opción para registrarse	Aplicación iniciada	Se muestra la opción para crear una cuenta	Se muestra la opción para crear una cuenta
CPR03	Verificar que se puede añadir un tutor	Aplicación iniciada	Al ingresar los campos solicitados se puede registrar un tutor	Al ingresar los campos solicitados se puede registrar un tutor
CPR04	Verificar funcionamiento para recuperar contraseña	Aplicación iniciada y un tutor registrado	Mostrar que para recuperar la contraseña se debe ingresar la cédula del usuario y se permita la modificación de los datos.	Se muestra que para recuperar la contraseña se debe ingresar la cédula del usuario y se permita la modificación de los datos.

CPR05	Verificar el funcionamiento del Login	Tutor registrado	Mostrar que al ingresar los datos sean correcto o incorrectos y muestren el resultado correspondiente	Mostrar que al ingresar los datos sean correctos o incorrectos y muestren el resultado correspondiente
CPR06	Comprobar la visualización de los estudiantes	Aplicación iniciada, ingreso con tutor, opción de menú seleccionada: estudiantes	Se muestra la pantalla con la lista de estudiantes registrados	Se muestra la pantalla con la lista de estudiantes registrados
CPR07	Verificar que se puede registrar un nuevo estudiante o editar uno.	Aplicación iniciada, ingreso con tutor, opción de menú seleccionada: estudiantes	Verificar que al seleccionar un estudiante registrado o al seleccionar la opción para añadir uno nuevo se permita la creación o modificación respectivamente	Se presentó un error en la captura de las imágenes utilizando la cámara del dispositivo, lo que no permitía registrar o modificar un estudiante.
CPR08	Verificar que efectivamente se crea o se edita un estudiante	Aplicación iniciada, ingreso con tutor, opción de menú seleccionada: estudiantes	Comprobar que se refleja en la lista los cambios realizados	Comprobar que se refleja en la lista los cambios realizados

Tabla 27: Pruebas de funcionalidad - Sprint 2.

Nota: El caso de prueba CPR07 no cumplió con el resultado esperado, este caso fue corregido de inmediato, dando los respectivos permisos de lectura, escritura y uso de la cámara del dispositivo.

3.1.3 Módulo personalización de talleres

Para el módulo de personalización de talleres se han definido 11 casos de prueba, cuyos resultados obtenidos se muestran en la tabla 28.

Sprint 3				
Código	**Descripción del caso**	**Prerrequisitos**	**Resultados**	
			Esperado	**Obtenido**
CPPT01 ✓	Verificar la visualización de los talleres que han sido registrados	Aplicación iniciada y login de un usuario	Se muestra los talleres registrados	Se muestra los talleres registrados
CPPT02 ✓	Verificar el funcionamiento del botón añadir	Aplicación iniciada y login de un usuario	Se muestra la pantalla para registrar un taller	Se muestra la pantalla para registrar un taller
CPPT03 ✓	Verificar funcionamiento de la edición de un taller	Aplicación iniciada, login y un taller registrado	Al seleccionar un taller registrado se puede editar el mismo	Al seleccionar un taller registrado se puede editar el mismo
CPPT04 ✓	Verificar el funcionamiento del botón guardar al crear o editar un taller	Taller seleccionado o botón añadir presionado	Se registra o se edita un taller satisfactoriamente	Se registra o se edita un taller satisfactoriamente

CPPT05	Verificar el funcionamiento del submenú de talleres, redirecciona a vocabulario	Taller registrado y seleccionado	Se muestra la pantalla de vocabulario	Se muestra la pantalla de vocabulario
CPPT06	Verificar el funcionamiento del submenú de talleres, redirecciona a historia	Taller registrado y seleccionado	Se muestra la pantalla de historia	Se muestra la pantalla de historia
CPPT07	Verificar el funcionamiento del submenú de talleres, redirecciona a secuencia	Taller registrado y seleccionado	Se muestra la pantalla de secuencia	Se muestra la pantalla de secuencia
CPPT08	Verificar creación y edición de vocabulario	Taller registrado, seleccionado y opción vocabulario seleccionada	Se muestra la pantalla para registrar o editar una palabra	El audio del vocabulario no se registraba correctamente
CPPT09	Verificar el funcionamiento del botón guardar al crear o editar una palabra	Taller seleccionado o botón añadir presionado	Se registra o se edita una palabra satisfactoriamente	Se registra o se edita una palabra satisfactoriamente

CPPT08	Verificar creación y edición de una historia	Taller registrado, seleccionado y opción historia seleccionada	Se muestra la pantalla para registrar o editar una historia	Se muestra la pantalla para registrar o editar una historia
CPPT09	Verificar el funcionamiento del botón guardar al crear o editar una historia	Taller seleccionado o botón añadir presionado	Se registra o se edita una historia satisfactoriamente	Se registra o se edita una historia satisfactoriamente
CPPT10	Verificar que la secuencia de cada historia se crea con el número de imágenes correspondiente a su dificultad	Taller seleccionado y opción secuencia seleccionada	Se muestra la pantalla de edición de secuencia con el número correspondiente de imágenes a su dificultad definida en la historia	Se muestra la pantalla de edición de secuencia con el número correspondiente de imágenes a su dificultad definida en la historia
CPPT11	Verificar que las imágenes se arrastren en los campos correspondientes de la secuencia	Imagen cargada ya sea de galería o desde cámara	Poder arrastrar una imagen seleccionada a una posición de la secuencia	Poder arrastrar una imagen seleccionada a una posición de la secuencia

Tabla 28: Pruebas de funcionalidad - Sprint 3.

Nota: El caso de prueba CPPT08 no cumplió con el resultado esperado, por tanto, se procedió a dar una solución a partir del otorgamiento de los respectivos permisos de uso de micrófono del dispositivo.

3.1.4 Módulo de entrenamiento

Para el módulo de personalización de talleres se han definido 15 casos de prueba, cuyos resultados obtenidos se muestran en la tabla 29.

Sprint 4				
Código	**Descripción del caso**	**Prerrequisitos**	**Resultados**	
			Esperado	**Obtenido**
CPE01	Verificar la lista de estudiantes registrados	Estudiantes registrados	Se muestra la lista de los estudiantes registrados	Se muestra la lista de los estudiantes registrados
CPE02	Verificar comprobación de contraseña del tutor	Estudiante seleccionado	Se muestra pantalla para ingresar contraseña	Se muestra pantalla para ingresar contraseña
CPE03	Verificar funcionamiento de la selección de dificultad del entrenamiento	Estudiante seleccionado, contraseña de tutor ingresada	Se muestra seek bar para selección de dificultad (Fácil, medio, difícil)	Se muestra seek bar para selección de dificultad (Fácil, medio, difícil)
CPE04	Verificar que se muestren todos los talleres registrados	Estudiante seleccionado, contraseña de tutor ingresada y dificultad seleccionada	Se muestra una lista de los talleres registrados	No se podía mostrar más de un taller a la vez

CPE05	Verificar presencia de la imagen e introducción del taller	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada y taller seleccionado	Mostrar la imagen introductoria del taller junto con su descripción	Se muestra la imagen introductoria del taller junto con su descripción
CPE06	Verificar lista de vocabulario del taller	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada y taller seleccionado	Verificar la presencia de todo el vocabulario existente en el taller	Se verifica la presencia de todo el vocabulario existente en el taller
CPE07	Verificar audio de cada palabra del vocabulario	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada y taller seleccionado	Verificar la existencia de audio relacionado con cada palabra del vocabulario	Podían reproducirse más de 1 audio a la vez
CPE08	Mostrar las historias asociadas al taller y a la dificultad	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada, taller seleccionado y repaso de vocabulario	Verificar lista de historias del taller que tengan la dificultad que ha sido seleccionada	Se verifica la lista de historias del taller que tengan la dificultad que ha sido seleccionada

CPE09	Verificar audio y descripción de las historias del taller	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada, taller seleccionado y repaso de vocabulario	Verificar que cada historia cuente con su respectiva introducción y que se reproduzca el audio relacionado	Se verifica que cada historia cuente con su respectiva introducción y que se reproduzca el audio relacionado
CPE10	Verificar que se muestre una pantalla que indique los pasos a seguir para ordenar la secuencia	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada, taller seleccionado, repaso de vocabulario e historia seleccionada	Verificar que cada historia primero muestra los pasos de la secuencia a ordenar	Los pasos se intercalaban entre las secuencias existentes
CPE11	Verificar que se permita elegir el tiempo de dura el juego	Estudiante seleccionado, contraseña de tutor ingresada, dificultad seleccionada, taller seleccionado, repaso de vocabulario e historia seleccionada	Verificar la existencia de un complemento que permite elegir el tiempo	El tiempo no dejaba de correr aun cuando la actividad se cancelaba
CPE12	Verificar pantalla del juego	Historia seleccionada y tiempo seleccionado	Comprobar que la pantalla de juego muestra las imágenes correspondientes a la historia	Se comprueba que la pantalla de juego muestra las imágenes correspondientes a la historia

CPE13	Verificar funcionalidad para arrastrar imágenes dentro del juego	Historia seleccionada y tiempo seleccionado	Comprobar que las imágenes en el juego se pueden arrastrar a los espacios requeridos	Se comprueba que las imágenes en el juego se pueden arrastrar a los espacios requeridos
CPE14	Comprobar resultados correctos e incorrectos	Historia seleccionada y tiempo seleccionado, juego finalizado correcto o incorrecto	Comprobar que el juego finaliza correctamente ya sea por culminación de ejercicio o por finalización del tiempo	Se comprueba que el juego finaliza correctamente ya sea por culminación de ejercicio o por finalización del tiempo
CPE15	Mostrar información de resultados	Juego finalizado	Comprobar que al finalizar el juego se muestre la información del mismo finalizado	Se comprueba que al finalizar el juego se muestre la información del mismo finalizado

Tabla 29: Pruebas de funcionalidad - Sprint 4.

Nota: El caso CPE04 se corrigió mediante la refactorización completa del código de esta sección. El caso CPE07 se resolvió implementando una validación en el reproductor de audio. El caso CPE10 se solucionó modificando la base de datos. El caso CPE11 se corrigió finalizando proceso en segundo plano.

3.1.5 Módulo de seguimiento y evaluación

Para el módulo de personalización de seguimiento y evaluación se ha definido 6 casos de prueba, cuyos resultados obtenidos se muestran en la tabla 30.

Sprint 5				
Código	**Descripción del caso**	**Prerrequisitos**	**Resultados**	
			Esperado	**Obtenido**
CPSE01	Verificar la lista de estudiantes registrados	Estudiantes registrados y ejercicios realizados por los mismos	Se muestra la lista de los estudiantes registrados	Se muestra la lista de los estudiantes registrados
CPSE02	Verificar gráfica de resultados con criterios de aciertos y fallos	Estudiante seleccionado, taller seleccionado, historia seleccionada y criterio seleccionado aciertos y fallos	Se muestra la gráfica de aciertos y fallos	Se muestra la gráfica de aciertos y fallos
CPSE03	Verificar gráfica de resultados con criterios de tiempo	Estudiante seleccionado, taller seleccionado, historia seleccionada y criterio seleccionado tiempo	Se muestra la gráfica de tiempo	La gráfica de tiempo se muestra al mismo tiempo que la gráfica de aciertos y fallos

CPSE04	Verificar funcionamiento de exportar gráfica	Estudiante seleccionado, taller seleccionado, historia seleccionada y criterio seleccionado	Se verifica que al escoger la opción exportar se crea una imagen de la gráfica	Se verifica que al escoger la opción exportar se crea una imagen de la gráfica
CPSE05	Verificar el funcionamiento de la opción ver tabla	Estudiante seleccionado y opción ver tabla seleccionada	Se muestra la tabla con los resultados de los ejercicios realizados por el estudiante	Se muestra la tabla con los resultados de los ejercicios realizados por el estudiante
CPSE06	Verificar funcionamiento de exportar tabla	Estudiante seleccionado, opción ver tabla seleccionada e icono exportar	Se verifica que al escoger la opción exportar se crea un archivo Excel	Se verifica que al escoger la opción exportar se crea un archivo Excel

Tabla 30: Pruebas de funcionalidad - Sprint 5.

Nota: El caso CPSE03 se solucionó mediante una validación de los criterios.

3.2 Pruebas Usabilidad

Una vez finalizado el desarrollo de la aplicación con sus respectivas pruebas, para corroborar el cumplimiento de los objetivos propuestos para el proyecto y la conformidad del stakeholder con el producto desarrollado, se procedió a la ejecución de pruebas de usabilidad.

El procedimiento para realizar las referidas pruebas consistió en 3 etapas:

- La primera etapa consiste en la elaboración de un protocolo en el cual se detallan los pasos a seguir (anexo C.1).
- En la segunda etapa se aplica el protocolo establecido.
- En la tercera etapa se analizan los resultados obtenidos en la etapa 2.

Para la ejecución del protocolo se utilizó la encuesta de usabilidad *System Usability Scale* (SUS) y las cartas de reacción de Microsoft adaptadas a lectura fácil.

Cabe destacar que el SUS se refiere a una herramienta metodológica que utilizada para medir la usabilidad de un objeto, dispositivo o aplicación. Por otro lado, la encuesta consiste en 10 preguntas, cada una de las cuales puede ser puntuada de 1 a 5, donde 1 significa Total desacuerdo y 5 significa Total acuerdo [56], a la cual se le añadieron 3 preguntas descriptivas, con el fin de obtener información relevante para mejorar la aplicación.

Conviene destacar, que las cartas de reacción de Microsoft capturan los sentimientos en base a una serie de palabras/frases, las cuales ayudan a evaluar las respuestas de los usuarios a las cualidades estéticas de una aplicación [57].

Por tal razón, se han escogido 9 palabras de una lista de 118, que fueron adaptadas a lectura fácil para una mejor comprensión por parte de los estudiantes de la fundación [58]. En la tabla 31 se muestra las palabras escogidas.

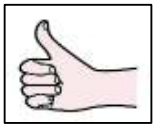
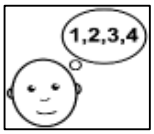
Creativo	Está bien 	Alegría
Confuso	Comprender	Feo
Enfadado	Amigable	Fácil 

Tabla 31: Lista de palabras adaptadas a lectura fácil.

Según [59], se aplica el test SUS a cinco usuarios, debido a que las pruebas con 5 usuarios permite encontrar casi la misma cantidad de problemas de usabilidad que se encontrarían al realizar la evaluación a más personas, mientras que con las cartas de Microsoft adaptadas a

lectura fácil se aplicara a los estudiantes que la fundación considere los más apropiados para experimentar la aplicación según su criterio.

Es así como se escogió aplicar la evaluación a partir del formulario de SUS a los tutores, debido a que estos pueden responder a una encuesta objetivamente. Las cartas de Microsoft adaptadas a lectura fácil se aplicaron a los estudiantes, por cuanto estos en su mayoría no saben leer o carecen de la capacidad de dar una respuesta objetiva a formularios planteados; además, no existe una forma de evaluar una aplicación dirigida para este tipo de usuarios que no conlleve un análisis a largo plazo y un conocimiento profundo de las discapacidades y las TIC. Por lo tanto, se propuso utilizar este test para obtener una reacción inicial a la aplicación por parte de los estudiantes.

3.2.1 Ejecución de pruebas

Fase 1 (Definición del protocolo)

Las actividades definidas en el protocolo son:

- Carta de consentimiento (anexo C.2).
- Encuesta demográfica para tutores (anexo C.3).
- Definición de la dinámica del experimento con los tutores y los estudiantes.
- Encuesta de usabilidad para el tutor y los estudiantes a su cargo (anexo C.4)

Fase 2 (Ejecución del protocolo)

La ejecución del protocolo se llevó a cabo de la siguiente manera:

- La carta de consentimiento fue realizada con el propósito de corroborar que los estudiantes participantes en la prueba contaban con la debida autorización para realizar las mismas, la cual fue recibida por la directora de la fundación, quien confirmó que los estudiantes pueden participar en el experimento.
- En la ejecución de la prueba primero se realizó una demostración de las funcionalidades del aplicativo al tutor asignado para las pruebas.
- Posteriormente, se realizó junto con el tutor un ejercicio para corroborar que el mismo entendió el funcionamiento del aplicativo.
- Luego, se procedió a poner en marcha los ejercicios con los estudiantes escogidos por el tutor, quien trabajó en conjunto con cada estudiante para realizar las pruebas. Las tareas del tutor fueron registrar a cada estudiante y ejecutar el entrenamiento con el mismo, mientras que las actividades del estudiante consistieron en realizar el entrenamiento guiado.

- Una vez finalizada la capacitación de un estudiante, se procedió a usar las tarjetas de reacción para determinar el sentimiento general que provocó el uso de la aplicación al estudiante lo cual se llevó a cabo con cada alumno, posterior al entrenamiento.
- Por último, una vez que se realizaron los ejercicios con todos los estudiantes escogidos, el tutor procedió a llenar la encuesta demográfica y el SUS.

Fase 3 (Análisis de resultados)

La última fase consistió en recolectar los resultados de las diferentes encuestas realizadas y tabular las mismas, a fin de determinar si la aplicación fue bien recibida por parte de los usuarios finales.

Evaluación estudiantes

Respecto a la evaluación practicada a los estudiantes se les solicitó escoger tarjetas, las cuales determinaron cuál fue su sentimiento al usar la aplicación, obteniendo como resultados los descritos en la tabla 32, los cuales evidencian que 5 de los 6 estudiantes marcan confuso como sentimiento hacia la aplicación; sin embargo, también se marcan las opciones fácil y comprensión, lo que implica que al iniciar la aplicación esta es confusa pero al finalizar el ejercicio la misma les parece fácil y con un contenido comprensible. En la tabla 32 se muestran cuáles palabras escogieron cada uno los estudiantes al finalizar el ejercicio.

Palabras	**Estudiante 1**	**Estudiante 2**	**Estudiante 3**	**Estudiante 4**	**Estudiante 5**	**Estudiante 6**	**Total**
Confuso	1	1	1	1	1		5
Feo							0
Está Bien (OK)			1	1			2
Comprender				1	1	1	3
Alegría					1	1	2
Amigable			1				1
Creativo							0
Enfadado							0
Fácil	1		1	1			3

Tabla 32: Palabras escogidas por los estudiantes

Tras la obtención de estos resultados se procedió a crear una nube de palabras mostrada en la figura 30, en la cual se presenta de manera más gráfica el efecto obtenido. Como se puede apreciar, aunque “confuso” sea el sentimiento predomínate este cambia a “fácil” y “compresible”, resaltando que la palabra confuso fue la más escogida, debido a que la aplicación mostrada fue totalmente desconocida para ellos, pero con el uso de la misma llega a ser fácil y comprensible.

Figura 31: Nube de palabras

La evaluación aplicada a los estudiantes se realizó en conjunto con el tutor, siendo éste la persona que ayudaba a elegir la palabra de manera más objetiva al estudiante, motivado a la condición del estudiante, quien no podía realizar una elección objetiva de las palabras. Por ejemplo, el estudiante seleccionaba en primera instancia la palabra alegría, pero al presentarle otra palabra como enfadado también era seleccionada, lo que presenta una contradicción. La presencia del tutor fue indispensable para realizar la evaluación y evitar este tipo de problemas.

Evaluación tutores

En esta sección de la evaluación conviene indicar que la fundación solo proporcionó a un tutor para las pruebas, debido a la disponibilidad de tiempo de los mismos, por lo cual a consideración de los autores se decidió realizar más evaluaciones a otras personas fuera de la fundación, con el fin de obtener una estadística más representativa de los resultados.

Previo a que los usuarios de prueba llenaran el formulario SUS, éstos completaron un formulario demográfico simple, con el fin de determinar las características de estas personas, lo cual arrojó como resultado que todos los usuarios de prueba tienen un nivel de uso de tecnologías por lo menos medio y su nivel escolar es superior al secundario; por tanto, no existe impedimento alguno para utilizar la aplicación de manera efectiva.

Para determinar el grado de usabilidad con el formulario SUS las respuestas siguieron el tratamiento descrito a continuación.

Existe una tentación de considerar a los resultados como porcentajes o incluso como percentiles, un error muy común y que puede derivar en resultados incorrectos. Por tal razón, y en pro de obtener los resultados, se suman aquellos datos promediados que se obtuvieron de los cuestionarios realizados a los usuarios, considerando lo siguiente: las preguntas impares (1, 3, 5,7 y 9) tomaron el valor asignado por el usuario, y se le restó 1, mientras que las preguntas pares (2, 4, 6, 8,10), fueron de 5 menos el valor asignado por los entrevistados. Una vez obtenido el número final, se multiplicó por 2,5 [60] .

La Figura 31, muestra cómo se comparan las calificaciones adjetivas (peor imaginable, pobre, ok, bien, excelente y mejor imaginable) con la escala de calificación escolar (F, D, C, B, A) y los rangos de aceptabilidad (no aceptable, marginal y aceptable) en relación al puntaje promedio del SUS.

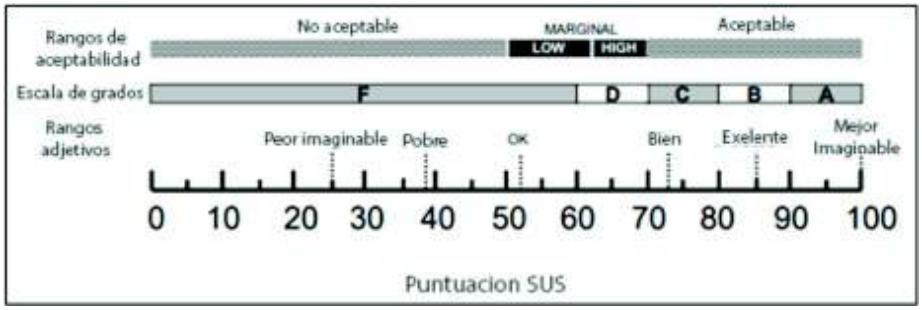


Figura 32: Puntaje promedio SUS (adaptado de [60])

En base a la escala mencionada previamente, se determina que la aplicación es aceptable para los usuarios objetivos. En la tabla 33 se observa el puntaje de cada usuario de prueba, un promedio ha obtenido por la aplicación de 87/100, lo cual se asemeja en la escala escolar a una B y el adjetivo es excelente.

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10	Totales
Encuesta 1	5	1	5	3	5	1	5	1	5	3	90
Encuesta 2	4	1	5	2	4	2	5	2	4	1	85
Encuesta 3	3	2	5	2	5	1	4	1	4	1	85
Encuesta 4	4	3	3	2	5	2	4	1	4	1	77,5
Encuesta 5	5	1	5	2	5	1	5	1	5	1	97,5
Promedio											87

Tabla 33: Resultados encuesta SUS

La evaluación aplicada al tutor de la fundación menciona que la aplicación es útil para estimular el lenguaje y el aprendizaje de los estudiantes. Por otro lado, los otros usuarios de prueba, tienen críticas positivas y solo sugieren cambios menores como colores y dibujos. Finalmente, todos los usuarios de prueba concuerdan en que la aplicación es fácil de usar y útil para la enseñanza.

En este capítulo se hicieron pruebas de usabilidad y funcionalidad, donde las primeras consistieron en seleccionar diferentes casos para cada módulo y probar si los mismos realmente cumplían los requisitos, y las pruebas de usabilidad se realizaron mediante el System Usability Scale (SUS), el cual generó una ponderación de si la aplicación es usable o no, siendo el resultado del presente caso de estudio positivo para estas pruebas, tanto de usabilidad como de funcionalidad. En el capítulo final se mencionan las conclusiones obtenidas de este trabajo.

CAPÍTULO 4

CONCLUSIONES

Desarrollo de una aplicación móvil para el entrenamiento de habilidades cognitivas mediante un enfoque ágil.

- Se ha desarrollado una aplicación móvil nativa para sistemas operativos Android mediante el uso de las metodologías ágiles Scrum, XP y la metodología iPlus.
- Scrum se utilizó para facilitar la gestión general del proyecto, debido a que usa iteraciones incrementales sprints y la forma en que permite manejar el grupo de trabajo facilita la interacción entre el mismo.

Implementar módulos de entrenamiento y vocabulario

- Se desarrolló el módulo de entrenamiento de vocabulario y secuencias temporales con el propósito de que las personas entrenen sus habilidades cognitivas. Por otra parte, se implementó el módulo de personalización de talleres, con el propósito de que los usuarios tutores puedan crear sus propios talleres con el contenido que consideren necesario.

Implementar módulo de seguimiento y evaluación

- Se ha implementado un módulo de seguimiento y evaluación que permite visualizar el progreso de la persona con discapacidad, mediante el despliegue de información referente a los ejercicios realizados por el estudiante, donde se puede notar el tiempo, el número de aciertos y fallas que ha tenido durante el transcurso de los ejercicios.

Evaluar el caso de estudio mediante pruebas de usabilidad y funcionalidad

- El caso de estudio se ha evaluado mediante pruebas de funcionalidad, las cuales se han desarrollado exitosamente utilizando casos de prueba para situaciones específicas de cada módulo, lo que demostró que la aplicación cumple con los requerimientos establecidos.
- La elaboración de prototipos con mockups de la aplicación y su respectiva evaluación por parte de expertos en usabilidad, aseguró encontrar de manera temprana posibles problemas en cuanto al diseño de la misma, lo que facilitó la etapa de desarrollo, puesto que los problemas encontrados por los evaluadores fueron menores y se

subsanaron en la fase inicial del proyecto sin mayores dificultades; a su vez, se logró prevenir que existan cambios drásticos en etapas avanzadas del desarrollo.

- Las pruebas de funcionalidad realizadas a los dos tipos de usuarios dieron resultados positivos, demostrando así que la aplicación puede ser usada sin contratiempos por los tutores y estudiantes. Cabe mencionar, que los estudiantes muestran un cierto grado de confusión al ser la primera vez que usan la aplicación, sin embargo, al finalizar el ejercicio se evidencia que los mismos la consideran fácil de utilizar, y comprendieron el ejercicio.

Propósito del login

- El login de la aplicación está diseñado con el fin de brindar al usuario un acceso controlado al aplicativo, de tal manera que los datos generados por cada usuario sean independientes entre sí, por cuanto la aplicación permite que cada tutor este a cargo de un grupo específico de estudiantes, a razón del funcionamiento interno de la fundación, que exige la presencia de un tutor a cargo de un taller en específico.

Alcance

- La aplicación puede ser usada en cualquier escenario laboral para la inserción y capacitación de nuevo personal, siempre y cuando los escenarios planteados estén bien desarrollados, lo cual permite la comprensión de los mismos.

PERSPECTIVAS Y RECOMEDACIONES

Con una infraestructura tecnológica mejor establecida por parte de la fundación, se recomienda refactorizar la aplicación de tal manera que se permita almacenar la información de los talleres, los datos de los usuarios, y las estadísticas de los estudiantes en un servidor web, para que puedan ser visualizadas o editadas en cualquier momento con mayor eficacia; esto considerando que la versión actual almacena y maneja toda la información de manera local.

Para la creación de nuevos talleres se recomienda utilizar imágenes ilustrativas que representen de forma clara y concisa los escenarios que se desean representar. Asimismo, se recomienda que antes de la creación de nuevos talleres se cuente con la definición del

proceso a representar, de tal forma que no existan errores en el número de imágenes o el orden de las secuencias creadas.

Se recomienda realizar un estudio para determinar una forma efectiva de medir la usabilidad de aplicaciones educativas para personas con discapacidad intelectual, debido a que actualmente, aunque existen diferentes formas para realizar esta evaluación no se toman en cuenta usuarios que presenten alguna discapacidad y las respuestas obtenidas de los diferentes métodos pueden ser ambiguas y generar disparidades en las estadísticas resultantes de evaluar la usabilidad de las aplicaciones.

REFERENCIAS BIBLIOGRÁFICAS

[1] C. Ordóñez, “Breve análisis de la inserción laboral de personas con discapacidad en el Ecuador,” *Alteridad, Revisra Educ.*, vol. 6, no. 2, pp. 145–147, 2011.

[2] CONADIS, “Estadísticas de Discapacidad,” 2018. [Online]. Available: https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/. [Accessed: 27-Dec-2018].

[3] R. A. Koon and M. E. de la Vega, “El impacto tecnológico en las personas con discapacidad.,” *II Congr. Iberoam. Informática Educ. Espec.*, p. 19, 2005.

[4] K. M. Fallas and F. C. Cordero, “TIC y Educación de Personas con Discapacidad en Costa Rica,” *Congr. Iberoam. Ciencia, Tecnol. Innovación y Educ.*, pp. 1–25, 2014.

[5] S. A. C. Gustavo and O. S. A. Vladimir, “Tendencias actuales en el uso de dispositivos móviles en educación,” *Educoas.Org*, vol. 147, p. 21, 2012.

[6] Fundacion Fine, “Fine.” [Online]. Available: http://www.fundacionfine.com/es/contacto.html. [Accessed: 22-Mar-2018].

[7] UNESCO, “Enfoque estratégico sobre tics en educación en américa latina y el caribe,” *Of. Reg. Educ. para América Lat. y el Caribe*, 2013.

[8] R. M. Hernandez, “Impacto de las TIC en la educación: Retos y Perspectivas,” *Woodhead Publ. Mater.*, vol. 5, pp. 325–347, 2017.

[9] S. Castro, B. Guzmán, and D. Casado, “Las Tic en los procesos de enseñanza y aprendizaje,” *Laurus*, vol. 13, no. 23, pp. 213–234, 2007.

[10] S. M. O. Montoya, “Software educativo para el buen uso de las Tics,” UNIVERSIDAD CATÓLICA DE PEREIRA, 2014.

[11] P. Marquès, “El software educativo.” [Online]. Available: http://www.lmi.ub.es/te/any96/marques_software/. [Accessed: 22-Mar-2018].

[12] A. V. Basantes, M. E. Naranjo, M. C. Gallegos, and N. M. Benítez, “Los dispositivos móviles en el proceso de aprendizaje de la facultad de educación ciencia y tecnología de la universidad técnica del norte de ecuador,” *Form. Univ.*, vol. 10, no. 2, pp. 79–88, 2017.

[13] Unesco, “Mobile Learning Week Report,” *Learning*, p. 29, 2011.

[14] P. Santana Mansilla, G. Lescano, and R. Costaguta, “Accesibilidad de aplicaciones móviles para discapacitados visuales: problemas y estrategias de solución,” no. September, 2015.

[15] P. G. Recacha and A. M. Cafranga, “Las personas con discapacidad intelectual ante las TIC,” *Comunicar*, vol. 18, no. 36, pp. 173–180, 2011.

[16] M. Zappalá, D. Köppel, A. Suchodolski, *Inclusión de TIC en escuelas para alumnos con Discapacidad Intelectual*. Argentina, 2011.

[17] M. L. de M. García, “Las TICs aplicadas a las Necesidades Educativas Especiales,” 2014.

[18] R. Sánchez, “TIC para estimular las Inteligencias,” *II Congr. Nac. sobre Discapac. Intelect.*, pp. 1–14, 2008.

[19] CEDETI, “La Mesita.” [Online]. Available: http://www.cedeti.cl/tecnologias-inclusivas/software-educativo/la-mesita/. [Accessed: 28-Dec-2018].

[20] D. visual D.O.C.E., “Evo: Programa de estimulación y entrenamiento visual.” [Online]. Available: https://asociaciondoce.com/2016/11/14/evo-programa-de-estimulacion-visual/. [Accessed: 28-Dec-2018].

[21] Educ@conTIC, “Abrapalabra.” [Online]. Available: http://www.educacontic.es/blog/abrapalabra. [Accessed: 28-Dec-2018].

[22] Stimulus, “APP profesional de estimulación cognitiva.” [Online]. Available: https://stimuluspro.com/familiar-stimulus-home/. [Accessed: 28-Dec-2018].

[23] Fundación Orange, “Soyvisual.” [Online]. Available: https://www.soyvisual.org/. [Accessed: 28-Dec-2018].

[24] A. A. García and R. Hidalgo, “Definiciones de discapacidad en España,” *Orthop. J. Sport. Med.*, pp. 1–17, 2011.

[25] J. Flórez, “Discapacidad intelectual: ¿Qué es? ¿Qué define? ¿Qué se pretende? Jesús,” *Motiv. Emot.*, vol. 30, no. 3, pp. 243–250, 2006.

[26] X. Ke and J. Liu, “Discapacidad intelectual,” in *Manual de Salud Mental Infantil y Adolescente de la IACAPAP OMS*, 2017, pp. 1–28.

[27] Scrum Guides, “The Scrum Guides.” [Online]. Available: https://www.scrumguides.org/scrum-guide.html#artifacts-sprintbacklog. [Accessed: 29-Dec-2018].

[28] K. Schwaber and J. Sutherland, “La Guía de Scrum,” *Scrumguides.Org*, vol. 1, p. 21, 2013.

[29] J. M. E. Lerena, “Aplicación para la gestión de proyectos ágiles con Scrum,” Universidad de La Rioja, 2013.

[30] B. K. Beck and C. Andres, *Extreme-Programming-Explained-Embrace-Change-2nd-Edition-*, Second Edi. Addison Wesley Professional, 2004.

[31] R. Jeffries, A. Anderson, and C. Hendrickson, *Extreme Programming Installed*. 2001.

[32] D. Karlström, “Introducing Extreme Programming – An Experience Report,” *Third Int. Conf. Extrem. Program. Agil. Process. Softw. Eng.*, pp. 24–29, 2002.

[33] P. Letelier and M. C. Penadés, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),” *Dep. Sist. Informáticos y Comput. (DSIC*, 2006.

[34] K. Mar and K. Schwaber, “Scrum with XP,” 2002.

[35] Universidad de Alicante, “Modelo vista controlador (MVC),” 2018. [Online]. Available: https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html. [Accessed: 29-Dec-2018].

[36] M. García, “MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?,” 2017. [Online]. Available: https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve. [Accessed: 02-Dec-2018].

[37] GitBook.com, “Architecture Components.” [Online]. Available: https://google-developer-training.github.io/android-developer-advanced-course-concepts/unit-6-working-with-architecture-components/lesson-14-architecture-components/14-1-c-architecture-components/14-1-c-architecture-components.html. [Accessed: 02-Dec-2018].

[38] A. Developers, “Guide to app architecture,” *Android Developers*, 2018. [Online]. Available: https://developer.android.com/jetpack/docs/guide. [Accessed: 02-Dec-2018].

[39] Android Developers, “Conoce Android Studio,” *Abril 25, 2018.*, 2018. [Online]. Available: https://developer.android.com/studio/intro/?hl=es-419. [Accessed: 02-Dec-2018].

[40] SQLite.org, “SQLite Home Page,” 2000. [Online]. Available: https://www.sqlite.org/index.html. [Accessed: 02-Dec-2018].

[41] Sql. Manager, “SQLite Manager 0.8.1,” 2013. [Online]. Available: https://add0n.com/sqlite-manager.html. [Accessed: 02-Dec-2018].

[42] Balsamiq, “Balsamiq. Rapid, effective and fun wireframing software. - Balsamiq,” 2015.

[Online]. Available: https://balsamiq.com/. [Accessed: 02-Dec-2018].

[43] Adobe, “Adobe Photoshop CC.” [Online]. Available: https://www.adobe.com/la/products/photoshop.html?promoid=PC1PQQ5T&mv=other. [Accessed: 29-Dec-2018].

[44] Adobe, “Adobe Illustrator CC.” [Online]. Available: https://www.adobe.com/la/products/illustrator.html?promoid=PGRQQLFS&mv=other. [Accessed: 29-Dec-2018].

[45] Freepik Company, “Free vector icons.” [Online]. Available: https://www.flaticon.com/. [Accessed: 29-Dec-2018].

[46] Icons8, “icons8.” [Online]. Available: https://iconos8.es/. [Accessed: 29-Dec-2018].

[47] ILoveIMG, “iLoveIMG Herramientas online para la edición de imágenes.” [Online]. Available: https://www.iloveimg.com/es. [Accessed: 29-Dec-2018].

[48] ZaoSolat.com, “ZapSplat.” [Online]. Available: https://www.zapsplat.com/. [Accessed: 29-Dec-2018].

[49] G. L. García, I. L. Martínez, G. S. P. Camarena, M. A. A. Figueroa, and B. A. O. Zepahua, “Arquitectura de software de una aplicación móvil para desarrollar un sistema de identificación por radiofrecuencia,” *ReCIBE. Rev. electrónica Comput. Informática, Biomédica y Electrónica*, p. 17, 2015.

[50] M. Carrion, M. Santorum, M. Perez, and J. Aguilar, “A participatory methodology for the design of serious games in the educational environment,” in *2017 Congreso Internacional de Innovacion y Tendencias en Ingenieria (CONIITI)*, 2017, pp. 1–6.

[51] J. Nielsen, “10 usability heuristics for user interface design,” *courses.ischool.utexas.edu*, 1995. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed: 30-Dec-2018].

[52] M. P. Gonzáles, J. Lorés, and A. Pascual, “Evaluacion Heuristica,” vol. 91, pp. 399–404, 2017.

[53] C. Pribueanu, “A Revised Set of Usability Heuristics for the Evaluation of Interactive Systems,” *Inform. Econ.*, vol. 21, no. 3/2017, pp. 31–38, 2017.

[54] G. Dinwiddie, “Feel the burn getting the most out of burn charts.” 2009.

[55] J. Gutiérrez, M. J. Escalona, M. Mejías, and J. Torres, “Pruebas del sistema en programacion extrema.”

[56] J. Sauro, “MeasuringU: Measuring Usability with the System Usability Scale (SUS),” 2011. [Online]. Available: https://measuringu.com/sus/. [Accessed: 30-Dec-2018].

[57] UX for the Masses, “Capturing user feedback with Microsoft’s product reaction cards,” 2002. [Online]. Available: http://www.uxforthemasses.com/product-reaction-cards/. [Accessed: 30-Dec-2018].

[58] Plena inclucion Madrid, “Diccionario Fácil.” [Online]. Available: http://diccionariofacil.org/proyecto/. [Accessed: 30-Dec-2018].

[59] J. Nielsen, “How Many Test Users in a Usability Study?,” 2012. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/. [Accessed: 10-Jan-2019].

[60] A. Bangor, T. Staff, P. Kortum, and J. Miller, “Determining What Individual SUS Scores Mean,” *J. usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.

ANEXOS

Los anexos que se presentan son en formato digital, debido a la extensión de los mismos (ver cd adjunto).

ANEXO A (JUSTIFICACIÓN Y PROTOCOLO DE LA METODOLOGÍA iPLUS)

A.1 Metodología de desarrollo

A.2 Protocolo experimental metodología iPLUS

A.3 Resultados protocolo iPLUS

A.4 Objetivo pedagógico

ANEXO B (DESARROLLO)

B.1 Protocolo de evaluación

B.2 Instrucciones de usuario

B.3 Mockup inicial

B.4 Formulario de evaluación

B.5 Resultados evaluación heurística

B.6 Mockup final

B.7 Sprints

ANEXO C (INSTRUMENTOS EVALUACIÓN)

C.1 Protocolo pruebas de usabilidad

C.2 Carta de consentimiento

C.3 Formato encuesta demográfica

C.4 Formato encuesta SUS

ANEXO D (RESULTADOS)

D.1 Evidencia multimedia del protocolo iPLUS

D.2 Resultado encuesta demográfica (escaneadas)

D.3 Resultado encuesta SUS (escaneadas)

D.4 Evidencia multimedia de pruebas de usabilidad