



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE SISTEMAS Y COMPUTACIÓN**

**Título “Desarrollo de un software educativo aplicando la metodología
Kanban, para el aprendizaje de vocabulario básico de inglés”**

**Trabajo de titulación para optar al título de Ingeniero en Sistemas
y Computación**

Autor:

Zhuño Quizhpe, Sahiry Xavier

Tutor:

Mgs. Lady Marieliza Espinoza Tinoco

Riobamba, Ecuador 2022

DERECHOS DE AUTORÍA

Yo, Sahiry Xavier Zhuño Quizhpe, con cédula de ciudadanía 1105353989, autor del trabajo de investigación titulado: **DESARROLLO DE UN SOFTWARE EDUCATIVO APLICANDO LA METODOLOGÍA KANBAN, PARA EL APRENDIZAJE DE VOCABULARIO BÁSICO DE INGLÉS**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 14 de abril del 2023.



Sahiry Xavier Zhuño Quizhpe

C.I: 1105353989



ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN CARRERAS NO VIGENTES

En la Ciudad de Riobamba, a los ocho días del mes de marzo de 2023, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **SAHIRY XAVIER ZHUÑO QUIZHPE**, con CC: **1105353989**, de la carrera **INGENIERIA EN SISTEMAS Y COMPUTACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado **"DESARROLLO DE UN SOFTWARE EDUCATIVO APLICANDO LA METODOLOGÍA KANBAN, PARA EL APRENDIZAJE DE VOCABULARIO BÁSICO DE INGLÉS"**, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Firmado electrónicamente por:
**LADY MARIELIZA
ESPINOZA TINOCO**

Mgs. Lady Espinoza
TUTOR

DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL;

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación DESARROLLO DE UN SOFTWARE EDUCATIVO APLICANDO LA METODOLOGÍA KANBAN, PARA EL APRENDIZAJE DE VOCABULARIO BÁSICO DE INGLÉS, presentado por Sahiry Xavier Zhuño Quizhpe con cédula de identidad número 1105353989, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 14 de abril del 2023

Milton Paul López Ramos, Mgs.
PRESIDENTE DEL TRIBUNAL DE GRADO

Miryan Estela Narváez Vilema, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

Ximena Alexandra Quintana López, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

Lady Marieliza Espinoza Tinoco, Mgs.
TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **DESARROLLO DE UN SOFTWARE EDUCATIVO APLICANDO LA METODOLOGÍA KANBAN, PARA EL APRENDIZAJE DE VOCABULARIO BÁSICO DE INGLÉS**, presentado por Sahiry Xavier Zhuño Quizhpe, con cédula de identidad número 1105353989, bajo la tutoría de Mg. Lady Marieliza Espinoza Tinoco; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 14 de abril del 2023.

Presidente del Tribunal de Grado
Mgs. Milton Paul López Ramos



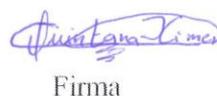
Firma

Miembro del Tribunal de Grado
PhD Miryan Estela Narváez Vilema



Firma

Miembro del Tribunal de Grado
PhD Ximena Alexandra Quintana López



Firma



CERTIFICACIÓN

Que, **SAHIRY XAVIER ZHUÑO QUIZHPE con CC: 1105353989**, estudiante de la Carrera **INGENIERIA EN SISTEMAS Y COMPUTACION NO VIGENTE**, Facultad de **INGENIERIA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "DESARROLLO DE UN SOFTWARE EDUCATIVO APLICANDO LA METODOLOGÍA KANBAN, PARA EL APRENDIZAJE DE VOCABULARIO BÁSICO DE INGLÉS", cumple con el **3 %**, de acuerdo al reporte del sistema Anti plagio **Urkund**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 8 de Marzo de 2023



El rúbrico electrónico pertenece a:
LADY MARIELIZA
ESPINOZA TINOCO

Mgs. Lady Espinoza
TUTOR TRABAJO DE INVESTIGACIÓN

DEDICATORIA

A Dios, por ser mi guía y fortaleza en momentos difíciles. A mis padres, por su amor incondicional y apoyo incansable en mi formación académica. A mi hermana, por su amistad y motivación constante. A la Universidad, por brindarme las herramientas y oportunidades necesarias para alcanzar mi meta. Este trabajo de titulación es un humilde homenaje a ustedes, que han sido parte fundamental de mi crecimiento y éxito.

Sahiry Xavier Zhuño Quizhpe

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a la Universidad Nacional de Chimborazo, especialmente a la carrera de Ingeniería en Sistemas y Computación, por brindarme la formación académica y los recursos necesarios para alcanzar mi meta. A los docentes, por su dedicación y paciencia en guiarme en mi camino de aprendizaje. A mis padres, por su incondicional amor y apoyo incansable en mi formación. A mi hermana, por su motivación y amistad constante. A Dios, por ser mi guía y fortaleza en los momentos más difíciles.

Sahiry Xavier Zhuño Quizhpe

ÍNDICE GENERAL

DERECHOS DE AUTORÍA.....	2
ACTA FAVORABLE	3
DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL.....	4
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL.....	5
CERTIFICADO ANTIPLAGIO.....	6
DEDICATORIA.....	7
AGRADECIMIENTO.....	8
RESUMEN.....	14
ABSTRACT	15
CAPÍTULO I. INTRODUCCION.....	16
1.1 PLANTEAMIENTO DEL PROBLEMA.....	17
1.1.1 Problema y Justificación.....	17
1.2 Objetivos	17
1.2.1 General.....	17
1.2.2 Específicos	17
CAPÍTULO II. MARCO TEÓRICO.....	18
2.1 Metodología de desarrollo de software	18
2.1.1 Tipos de metodologías de desarrollo de software.....	18
2.1.2 Metodología tradicional.....	19
2.1.3 Metodología Ágil.....	20
2.1.4 Valores del manifiesto ágil	21
2.1.5 Principios de la metodología ágil.....	22
2.1.6 Principales metodologías ágiles.....	22
2.1.7 Comparación entre metodología tradicional y ágil.....	23
2.2 Kanban en la industria de producción.....	24
2.2.1 Historia	24
2.2.2 Método Kanban de producción.....	25

2.2.3	Principios básicos	25
2.2.4	Tipos de Kanban	26
2.3	Metodología ágil Kanban en el desarrollo de software.....	27
2.3.1	Principios	28
2.3.2	Prácticas de Kanban.....	28
2.3.3	Tablero Kanban.....	30
2.3.4	Herramientas Kanban	32
2.4	Norma ISO/IEC 25000	33
2.4.1	ISO/IEC 2501n - Modelo de calidad	33
2.4.2	Característica - Usabilidad.....	34
2.4.3	Subcaracterística - Facilidad de aprendizaje.....	34
CAPÍTULO III. METODOLOGIA		35
3.1	Tipo y diseño de investigación.....	35
3.1.1	Según la fuente de investigación.....	35
3.1.2	Según el objeto de estudio	35
3.2	Identificación de Variables.....	35
3.2.1	Variable Independiente	35
3.2.2	Variable dependiente.....	35
3.2.3	Operacionalización de variables.....	35
3.3	Población y Muestra.....	38
3.3.1	Población.....	38
3.3.2	Muestra	38
3.4	Técnicas e instrumentos de recolección de datos	38
3.4.1	Encuesta.....	38
3.5	Técnicas de análisis e interpretación de la información	38
3.6	Procedimientos.....	39
3.6.1	Análisis de requerimientos	39

3.6.2	Diseño	39
3.6.3	Desarrollo del software educativo aplicando Kanban	44
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN		56
4.1	Resultados de la encuesta de Evaluación de Usabilidad/Criterio facilidad de aprendizaje	57
4.2	Discusión de los resultados Evaluación de Usabilidad – Criterio facilidad de aprendizaje	60
CAPÍTULO V. CONCLUSIONES y RECOMENDACIONES.....		62
5.1	Conclusiones	62
5.2	Recomendaciones.....	62
BIBLIOGRAFÍA		63
ANEXOS.....		67

ÍNDICE DE TABLAS

Tabla 1:	Comparativa entre las metodologías tradicionales y ágiles	23
Tabla 2:	Operacionalización de variables	36
Tabla 3:	Población de estudio.....	38
Tabla 4:	Requerimientos funcionales	39
Tabla 5:	Requerimientos no funcionales	39
Tabla 6:	Escala de Likert - Pregunta 1.....	56
Tabla 7:	Escala de Likert - Pregunta 2.....	56
Tabla 8:	Escala de Likert - Pregunta 3 - 4	56
Tabla 9:	Escala de Likert - Pregunta 5.....	56
Tabla 10:	Reporte-tiempo	59
Tabla 11:	Resultado de encuesta – Tiempo empleado.....	60
Tabla 12:	Resultados de la encuesta	60

ÍNDICE DE FIGURAS

Figura 1: Metodologías tradicionales.....	20
Figura 2: Metodologías ágiles.....	21
Figura 3: Taiichi Ōno	25
Figura 4: Kanban o tarjeta de retiro.....	26
Figura 5: Kanban o tarjeta de producción	27
Figura 6: Tablero Kanban.....	30
Figura 7: Componentes del tablero Kanban	31
Figura 8: División de la Familia ISO/IEC 25000.....	33
Figura 9: Características del modelo de calidad.....	34
Figura 10: Diseño - Splash screen	40
Figura 11: Diseño - ventana de inicio de sesión	40
Figura 12: Diseño - ventana menú principal.....	40
Figura 13: Diseño - ventana lección 1 – Escuchar	41
Figura 14: Diseño - ventana lección 2 – Escribir.....	41
Figura 15: Diseño - ventana lección 3 – Selección.....	42
Figura 16: Diagrama de caso de uso.....	42
Figura 17: Arquitectura N capas.....	43
Figura 18: Diagrama entidad-relación (Diagrama ER)	43
Figura 19: Diagrama de base de datos	44
Figura 20: Flujo de trabajo	45
Figura 21: Etapa 1 - tablero Kanban.....	45
Figura 22: Etapa 2 - tablero Kanban.....	46
Figura 23: Etapa 3 - Tablero Kanban	47
Figura 24: Fracción de código – ClsDataBase	48
Figura 25: Código de la clase – ClsPalabra	49
Figura 26: Código de la clase – ClsUsuario.....	50
Figura 27: Código de la clase – ClsPalabraLN.....	51
Figura 28: Fracción de código – ClsUsuarioLN.....	52
Figura 29: Splash screen.....	52
Figura 30: Interfaz y código - ventana de inicio de sesión y registro	53
Figura 31: Interfaz y código - ventana menú principal.....	53
Figura 32: Interfaz y código - ventana lección 1	54

Figura 33: Interfaz y código - ventana lección 2	54
Figura 34: Interfaz y código - ventana lección 3	55
Figura 35: Pregunta 1 – Evaluación de usabilidad	57
Figura 36: Pregunta 2 – Evaluación de usabilidad	57
Figura 37: Pregunta 3 – Evaluación de usabilidad	58
Figura 38: Pregunta 4 – Evaluación de usabilidad	58
Figura 39: Pregunta 5 – Evaluación de usabilidad	59

RESUMEN

El desarrollo de software es un proceso complejo que requiere gestión efectiva de los recursos y la organización correcta de las tareas para obtener un producto de calidad en el menor tiempo. Debido a su enfoque visual y eficiente gestión de proyectos, Kanban ha ganado popularidad como metodología para el desarrollo de software. Utilizando las etapas del ciclo de vida del desarrollo de software, Kanban representa las diferentes etapas del proceso. Visualizar el flujo de trabajo, limitar el trabajo en progreso y medir el tiempo empleado son las reglas de Kanban. El presente proyecto de investigación tuvo como objetivo desarrollo de software educativo aplicando la metodología Kanban, para el aprendizaje de vocabulario básico de inglés. Esta investigación se desarrolló bajo el enfoque cuantitativo, investigación bibliográfica e investigación aplicada. La técnica de recolección de datos utilizada fue la encuesta y se definió escalas de Likert para las diferentes preguntas, con el propósito de evaluar la usabilidad en base a la Norma ISO/IEC 25000 criterio facilidad de aprendizaje. En base a los resultados se evidenció que la interfaz del software desarrollado es intuitiva y fácil de aprender, lo que permitió la realización de las tareas propuestas en un tiempo máximo de 2 a 4 minutos, del mismo modo se evidenció que el software no presentó errores durante su manejo. Finalmente, la investigación indicó que aplicar la metodología Kanban en el desarrollo de proyectos de software es beneficioso, permite entregar un producto de calidad en el tiempo establecido.

Palabras claves: Desarrollo de software, Facilidad de aprendizaje, Metodología Kanban, Norma ISO/IEC 25000.

ABSTRACT

Software development is a complex process that involves effective management of resources and the correct organization of tasks in order to obtain a high quality product within a short timeframe. Due to its visual approach and as an effective methodology for project management, Kanban has become popular as a methodology for software development. Using the stages of the software development life cycle. Visualizing the workflow, limiting the work in progress and measuring the time spent are the rules of Kanban. **“THE AIM OF THE PRESENT RESEARCH PROJECT IS AT DEVELOPING EDUCATIONAL SOFTWARE APPLYING THE KANBAN METHODOLOGY FOR LEARNING BASIC ENGLISH VOCABULARY”**. This research was developed under the quantitative approach, bibliographic research and applied research. The data collection technique used was the survey and Likert scales were used to evaluate different questions, with the purpose of evaluating usability based on the ISO/IEC 25000 standard, the ease of learning criterion. Based on the results, it was found that the interface of the developed software is intuitive and easy to learn, which allowed the completion of the proposed tasks in a maximum time of 2 to 4 minutes, in the same way it was found that the software did not present any errors during its use. Finally, the research indicated that applying the Kanban methodology is beneficial in the development of software projects, therefore it allows to deliver a good quality product on time.

Keywords: Software development, Ease of learning, Kanban methodology, ISO/IEC 25000 standards.

DORIS ELIZABETH VALLE VINUEZA
Firma do digitalmente por DORIS ELIZABETH VALLE VINUEZA
Fecha: 2023.03.30 12:04:13 -05'00'

Reviewed by: Mgs. Doris Valle V.

ENGLISH PROFESSOR

c.c 0602019697

CAPÍTULO I. INTRODUCCION

El desarrollo de software ha recorrido un largo camino desde sus inicios por la década de 1940. Al principio, el software se usaba principalmente con fines científicos y militares, pero con el tiempo se ha expandido a múltiples campos de la industria y aplicaciones. A medida que la tecnología evoluciona y la demanda de soluciones digitales continúa aumentando, los desarrolladores de software deben adaptarse constantemente a las necesidades cambiantes de los usuarios. Desde la creación de aplicaciones simples hasta sistemas complejos de inteligencia artificial, el desarrollo de software es una industria en constante evolución que requiere habilidades técnicas, creativas y una comprensión profunda de las necesidades del usuario final. En este contexto, los desarrolladores deben mantenerse al tanto de las últimas tendencias y herramientas para crear soluciones innovadoras que satisfagan las necesidades cambiantes de los usuarios.

El desarrollo de software, como cualquier otro proyecto, tiene un ciclo de vida que consta de varios pasos que se completan a lo largo del proceso de desarrollo del mismo. El ciclo de vida depende directamente de la metodología utilizada. Una metodología de desarrollo de software no es más que una serie de pasos que deben seguirse rigurosamente para producir un software nuevo o modificado de acuerdo a requerimientos de los usuarios finales.

Debido a la constante demanda de nuevos programas, la industria del software se ha visto en la necesidad de utilizar nuevas metodologías para acelerar el proceso de desarrollo sin perder la calidad del producto final, por lo que la industria del software ha adoptado el método Kanban para el desarrollo de software. Kanban tiene sus orígenes en la industria manufacturera, específicamente en el fabricante de automóviles Japonés Toyota. El método Kanban es una metodología de desarrollo de software lean que se enfoca en entregar funciones a tiempo, con un fuerte énfasis en la gestión de cargas de trabajo. Usando Kanban, se describe en primera instancia las tareas a realizar y permite cambiar su prioridad. Con Kanban, los equipos pueden visualizar el estado actual de cada actividad.

En esta demanda de nuevos programas también se encuentra inmerso el software educativo, cuya función es proveer de herramientas a profesores y/o alumnos con la finalidad de facilitar el aprendizaje. Por tal motivo, en este proyecto de investigación se desarrolló un software educativo para el aprendizaje del vocabulario básico de inglés aplicando la metodología Kanban, ya que en la actualidad saber hablar el idioma inglés resulta imprescindible para el desarrollo en el ámbito educativo y para el desarrollo humano.

1.1 PLANTEAMIENTO DEL PROBLEMA

1.1.1 Problema y Justificación

De acuerdo al ranking EF EPI (EF English Proficiency Index), del año 2021 presentado por EF Education First, clasifica a los países de acuerdo al nivel de dominio del idioma inglés, en este ranking Ecuador se ubica en el puesto 90 de entre 100 países que fueron examinados y a nivel de Latinoamérica Ecuador ocupa el puesto 19, manteniéndose como uno de los países con un nivel de inglés bajo. En referencia a provincias del Ecuador, Azuay se encuentra en primer puesto con 485 puntos superando a Pichincha y Guayas. La provincia con el nivel más bajo de inglés es Bolívar con 332 puntos. Tomando este ranking y las puntuaciones de las provincias es fundamental promover el desarrollo de aplicativos de enseñanza del idioma inglés que permitan de forma efectiva contribuir a desarrollar el vocabulario básico.

En el área de desarrollo de software una de las etapas importantes es la planificación de las actividades de cada fase del desarrollo que se deben llevar a cabo para construir el producto final y entregar en la fecha establecida, los retrasos en la entrega del producto final en muchos de los casos se deben a la mala gestión de las actividades, para el cumplimiento de estas actividades el equipo de desarrollo tiene una mayor carga de trabajo, esto afecta directamente a la calidad del producto final.

En consecuencia, de lo antes mencionado se plantea como solución desarrollar un software educativo para facilitar el aprendizaje de vocabulario básico de inglés y gestionar de mejor forma las actividades de cada una de las etapas del desarrollo mediante la aplicación de la metodología Kanban, el software estará dirigido a niños entre 10 a 12 años.

1.2 Objetivos

1.2.1 General

- Desarrollar un software educativo aplicando la metodología Kanban, para el aprendizaje de vocabulario básico de inglés.

1.2.2 Específicos

- Analizar la metodología Kanban y el uso de tablero Kanban para el control de actividades de la fase de desarrollo.
- Implementar un software educativo aplicando la metodología Kanban.
- Evaluar la usabilidad del software educativo mediante la Norma ISO/IEC 25000, criterio facilidad de aprendizaje.

CAPÍTULO II. MARCO TEÓRICO

2.1 Metodología de desarrollo de software

El desarrollo de software y su complejidad aumenta de acuerdo al tamaño del proyecto y del equipo de trabajo. Llevar a cabo el desarrollo de un proyecto sin una metodología conlleva a problemas, errores, retrasos en la entrega dando como resultado un producto poco satisfactorio para el usuario. Las metodologías de desarrollo de software surgen por la necesidad de contar con procesos y métodos, ya que en principio el desarrollo de software se llevaba a cabo de manera empírica llegando a fracasar un gran porcentaje de proyecto de desarrollo de software. El fracaso de estos proyectos se debía a que el producto final no cubría las expectativas de los usuarios, la entrega en muchos casos estaría fuera del tiempo estimado y esto conlleva a exceder el presupuesto establecido para el desarrollo del mismo. A lo largo de los años las metodologías de desarrollo de software han ido evolucionando pasando de ser un simple documento de organización a convertirse en la base para el desarrollo de software eficiente y productivo. Las metodologías de desarrollo de software son un conjunto de técnicas y métodos, los mismos que son empleados para diseñar y construir un sistema informático que cubra las expectativas de los usuarios y al imponer un proceso sistemático busca que el desarrollo de software sea más eficiente.

Una metodología de desarrollo de software consta de fases o procesos seguidos sistemáticamente para diseñar, desarrollar, implementar y mantener un software desde la fase inicial en el que se plantea la necesidad del producto software hasta cumplir con el objetivo de su desarrollo (Maida & Pacienza, 2015). Se emplea una metodología de desarrollo de software para alcanzar el objetivo de trabajo en equipo y llevarlo de manera organizada (Santander Universidades, 2020).

Otro de los beneficios por el cual es ampliamente utilizado una metodología, es llevar un mejor control en el desarrollo del trabajo, consiguiendo minimizar los márgenes de errores y de acuerdo a cada situación no favorable anticiparse a estos. Como ventajas al utilizar una metodología se puede mencionar una mejor gestión de los recursos disponibles así también como el ahorro de tiempo. Existen varias metodologías que pueden ser empleadas para el desarrollo de software.

Cada una de las metodologías tienen sus propias reglas y principios. Elegir una metodología para el desarrollo de software depende de varios factores como: el tamaño del proyecto, tipo de software, tamaño del equipo, entre otros, por lo tanto, elegir una metodología queda a decisión del equipo de trabajo (Solera, 2022).

2.1.1 Tipos de metodologías de desarrollo de software

Las metodologías de desarrollo de software se encuentran agrupadas en dos grandes grupos: las metodologías tradicionales y las metodologías ágiles. Según Tinoco, Rosales y Salas

(2010), los requisitos que deben tener las metodologías de desarrollo, esto de acuerdo a varios autores, son:

- Objetivo del producto.
- Relación con el cliente.
- Fijar un modelo de ciclo de vida.
- Gestión de requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas de evaluación de la calidad.
- Métricas para medir el riesgo.
- Gestión de cambios.
- Establecer una línea de meta.

2.1.2 Metodología tradicional

Las metodologías tradicionales son, como su nombre indica, aquellas que se han utilizado durante toda la vida. Su objetivo es disciplinar el proceso de desarrollo de software y, por lo tanto, hacerlo predecible y eficiente (Demera, 2021).

La principal característica por la que se diferencia la metodología de desarrollo de software tradicional de la ágil, es que esta define al principio del proyecto de ingeniería de software los requisitos con los que debe contar el producto final, definir los requisitos tiene un gran impacto ya que de esto depende los recursos que se empleará en el proyecto. Los ciclos que sigue esta metodología de desarrollo son poco flexibles y no permite realizar cambios en los mismos.

A mediados de la década de los 60, se lanzó el concepto de ciclo de vida de desarrollo de software (SDLC siglas en inglés), este concepto surgió por la necesidad de administrar de manera más efectiva el desarrollo de software. Además, este concepto proporcionaría las pautas en las que se establecería los estados o fases por los cuales debería pasar el producto software desde que nace por una necesidad, hasta su entrega al usuario final (Garcés & Egas, 2013).

A medida que avanza los años, los clientes tienen nuevos requerimientos ya sea el tiempo de entrega del proyecto, la disminución de costos o la flexibilidad durante el desarrollo del proyecto, debido a estos nuevos requerimientos los equipos de desarrollo tienen la necesidad de implementar una metodología de desarrollo (Alvarez et al., 2021).



Figura 1: Metodologías tradicionales

Recuperado de: <http://ingenieriaabril.blogspot.com/2016/09/metodologias-para-el-desarrollo-de.html>

En la Figura 1 se visualiza las fases del ciclo de vida de las metodologías tradicionales, este conjunto de fases permite que el trabajo se organice y se realice de manera lineal, es decir, las etapas se desarrollan una seguida de otra y no se puede dar inicio a una nueva etapa sin completar la anterior. Una vez que ha iniciado una nueva etapa no se puede volver hacia atrás, es decir no se puede retroceder a las etapas ya terminadas.

Según Ginzo Technologies (2021), las principales metodologías tradicionales o clásicas son:

- En cascada
- Incremental
- Prototipo
- Espiral
- Desarrollo rápido de aplicaciones (RAD)

2.1.3 Metodología Ágil

Es una forma innovadora de organizar el trabajo y los flujos, divide el proyecto en pequeñas partes lo que permite correcciones, adiciones y fases de resolución inmediatas en un corto período de tiempo (SALESFORCE LATINOAMÉRICA, 2021).

El término “ágil” enfocado al desarrollo de software, nace en el año 2001, en la reunión celebrada en EEUU, en la cual fueron participes 17 expertos del ámbito del software. Esta metodología se crea como respuesta al enfoque en cascada, donde los proyectos se organizan como una serie de secuencias lineales y los resultados de los proyectos no son los esperados. El manifiesto ágil para el desarrollo de software fue redactado por un grupo de desarrolladores de software (Red Hat, 2022).

Las metodologías ágiles para la gestión de proyectos tienen su origen en TI, estas fueron creadas a partir de la insatisfacción de los desarrolladores frente a la poca fluidez de las metodologías tradicionales, ya que estas se caracterizan principalmente por ser rígidas. La

metodología ágil corrige esta característica y establece los valores fundamentales. Desde sus inicios, a las metodologías ágiles se han ido agregado varias técnicas y marcos de gestión de proyectos (Gabaldón, 2021). En la Figura 2 se visualiza las fases de las metodologías ágiles.



Figura 2: Metodologías ágiles

Recuperado de: <https://www.tithink.com/es/2018/10/16/metodologias-agiles-que-son-y-para-que-sirven/>

En la actualidad, la metodología ágil es empleada en la mayor parte de proyectos de desarrollo debido a los cambios frecuentes y las nuevas necesidades de productos software por parte de los usuarios. Según Calvo (2018), las metodologías ágiles desde sus inicios hasta la actualidad han tratado de agilizar las respuestas frente a los cambios suscitados durante el desarrollo de un proyecto.

Las pautas, principios y técnicas que son proporcionadas por las metodologías ágiles, permiten obtener un producto sin errores y entregar el proyecto sin mayores complicaciones, incrementando el nivel de satisfacción de los clientes y del equipo de trabajo (Hernández, 2016).

2.1.4 Valores del manifiesto ágil

Los modelos de organización que fueron promovidos por los creadores del manifiesto ágil, se centran en la persona y la colaboración. Estos modelos cambian el enfoque del entorno de trabajo, dejando de hablar de las personas como uno de los activos más importantes de la compañía, en enfocarse en las necesidades de los mismos con el fin de llevar un ambiente laboral agradable para si poder ofrecer productos de calidad a los clientes (Sentrío, 2021).

Las metodologías ágiles se fundamentan en 4 valores:

1. Individuos e interacción por encima de los procesos y herramientas.
2. El software por encima de la documentación.
3. Colaborar con el cliente por encima de la negociación contractual.
4. Respuesta al cambio por encima de seguir con un plan.

2.1.5 Principios de la metodología ágil

Las metodologías ágiles cuentan con 12 principios básicos para su desarrollo (Sáez, 2021), estos principios son:

1. Satisfacer al cliente con la entrega de trabajo continuo y de calidad.
2. Dividir las tareas complejas para poder completar con facilidad.
3. Reconocer que los equipos organizados realizan mejores trabajos.
4. Construir proyectos en entornos de trabajo propicios y motivar al grupo de trabajo mediante el voto de confianza.
5. Apoyar el desarrollo del trabajo sostenible mediante procesos.
6. Mantener un trabajo constante en la organización.
7. Ajustarse a nuevos cambios incluso al final del proyecto.
8. Mantener reuniones constantes entre el equipo de trabajo y los directivos para tratar los cambios y los avances del proyecto.
9. Recordar a los empleados los objetivos para llevar un trabajo más efectivo.
10. Medir el rendimiento conforme avanza en el trabajo.
11. Mantener la excelencia continua.
12. Adaptarse a circunstancias cambiantes.

2.1.6 Principales metodologías ágiles

- **SCRUM**

Esta metodología permite resolver proyectos complejos que se desarrollan en ambientes cambiantes, esto de acuerdo a su principal característica la flexibilidad (Sáez, 2021). Esta metodología se fundamenta en el desarrollo incremental, el ciclo de desarrollo del producto se divide en pequeños proyectos por etapas como son: análisis, desarrollo y testing (Toledo, 2022). Al aplicar scrum se busca la mejora del trabajo colaborativo entre el personal.

Los principales términos utilizados en esta metodología son: producto owner, scrum master, producto backlog, historias de usuario, tareas y equipo. Al aplicar esta metodología se considera en primera instancia una planificación ligera del trabajo a realizar, el número de sprints y la duración que tendrán cada uno.

- **Extreme Programming XP**

Se enfoca en mejorar las relaciones interpersonales con el fin de llevar con éxito el desarrollo de software, fomentando principalmente el trabajo en equipo, promoviendo y cuidando el aprendizaje de los desarrolladores y fomentando un buen ambiente de trabajo. Con la metodología XP la retroalimentación cliente y equipo de desarrollo es continua, además, la comunicación es flexible entre todos los involucrados, promueve la implementación de soluciones simples y estar prestos para los cambios. Esta metodología es ideal para aquellos proyectos que cuenten con

requisitos imprecisos y cambiantes y para proyectos con alto riesgo técnico (Letelier & Penadés, 2006).

La metodología XP cuenta con valores como: la comunicación, simplicidad, retroalimentación, coraje y respeto. Además, se basa en principios como: realimentación rápida, asumir simplicidad, cambio incremental, ajustarse al cambio y trabajo de calidad (López, 2020).

- Lean

Es un enfoque de desarrollo de software que combina elementos básicos de metodologías ágiles y lean manufacturing. Se enfoca en entregar valor al cliente de manera rápida y eficiente, haciendo énfasis en la mejora continua y la eliminación de desperdicios. Se fundamenta en la colaboración y la flexibilidad, permitiendo una adaptación rápida a los cambios y una mejor comunicación entre los equipos de desarrollo y los clientes. Esta metodología permite una mayor eficiencia y productividad, resultando en soluciones de alta calidad y una mayor satisfacción del cliente.

Esta metodología busca la reducción de tiempos de entrega, la mejora en la calidad y la reducción de los precios de los productos finales. La metodología Lean se base en 7 principios: lo fundamental es lo esencial, optimizar, la calidad óptima, aprendizaje continuo, reaccionar con fluidez, mejora constante y equipo de trabajo (INESDI, 2021).

2.1.7 Comparación entre metodología tradicional y ágil

Los autores Navarro, Fernández y Morale (2013), listan las principales características de las metodologías tradicionales y ágiles, contrastando cada una de ellas. Esta comparativa se muestra en la Tabla 1:

Tabla 1: Comparativa entre las metodologías tradicionales y ágiles

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	El proyecto es dividido en varios subproyectos
Poca comunicación con el cliente	Comunicación contante con el cliente

Entrega del software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Fuente: (Navarro, Fernández y Morales, 2013)

2.2 Kanban en la industria de producción

Uno de los factores que más influye en el sector de la producción es la planificación. Con una buena planificación se asegura la existencia de inventario suficiente de acuerdo a la demanda del mercado. Otro factor a considerar es la productividad mediante la vigilancia de los procesos de producción, estos procesos deben tener un ritmo adecuado y así conseguir incrementar la productividad. No cumplir con una de estas funciones puede generar problemas, afectando directamente a la finanza de la empresa. Para solventar es necesario la flexibilidad en las áreas de producción, debido a que las demandas del mercado son fluctuantes. Las empresas deben tener la capacidad de adaptarse a estos cambios (Melanie, 2022).

Kanban es un sistema de producción en el cual se utilizan señales visuales (tarjetas o tableros) para indicar la necesidad de producir o reponer un producto. El objetivo es asegurar un flujo continuo de materiales y evitar la acumulación de inventario innecesario. Este sistema es comúnmente utilizado en la industria automotriz y de electrónica, y se ha extendido a otros campos como la manufactura en general y en el ámbito de la tecnología de información y comunicación en el campo de desarrollo de software.

2.2.1 Historia

Kanban se hizo popular por primera vez en la década de 1940 cuando el gigante automotriz japonés Toyota buscó optimizar sus procesos de fabricación y reducir los costos relacionados con el inventario. Taiichi Ohno (ver Figura 3), ingeniero de Toyota, desarrolló un sistema de producción basado en la visualización de tareas y reducción de inventario, que permitía producir solo lo necesario en el momento adecuado. Kanban, se traduce al español como "tarjetas visuales", es el nombre de este sistema. El propósito del sistema Kanban es mantener un flujo de producción estable evitando almacenamientos innecesarios y minimizando el tiempo de espera en la cadena de producción. Como una herramienta de gran

prestigio para la gestión de proyectos y la mejora de procesos, el sistema Kanban finalmente se extendió a otros mercados e industrias (kanbantool, 2020).



Figura 3: Taiichi Ōno

Recuperado de: <https://www.digite.com/es/kanban/que-es-kanban/>

El método Kanban es desarrollado por el ingeniero industrial japonés de Toyota Taiichi Ohno, por la década de los 50. El objetivo de este método es controlar el flujo de materia prima en la fabricación de productos. Al ser un método eficiente, efectivo y mejorable, permite que el ámbito de trabajo sea el más óptimo, mejorando la competitividad de la empresa en el mercado. En la actualidad las compañías buscan minimizar la pérdida de material y el tiempo que se emplea para su fabricación, sin perder la calidad del producto (Pinto de los Ríos, 2015).

2.2.2 Método Kanban de producción

El Kanban de producción cuenta con los siguientes enfoques (Melanie, 2022):

- Mejora continua en la producción. Producción instantánea.
- Dar instrucciones basadas en las condiciones del ambiente de trabajo.
- Prevenir la acumulación o exceso de trabajo en la línea de producción.
- Identificar las debilidades o retrasos de los procesos de producción mediante controles.
- Eliminar la sobreproducción.

2.2.3 Principios básicos

Según Vergara (2017), el sistema Kanban cuenta con varios principios básicos, los mismos que se enumeran a continuación:

1. Calidad garantizada

El resultado de la producción debe salir de la mejor manera, sin margen de error. Para obtener este resultado Kanban vela por la calidad y no por la rapidez con la que las tareas son realizadas.

2. Reducción de desperdicios

Busca realizar lo justo y necesario, reduciendo todo lo superficial y secundario, pero hacerlo de la mejor manera.

3. Mejora continua

De acuerdo a los objetivos planteados, este sistema permite realizar mejoras en el desarrollo del proyecto y es, además, un método de gestión.

4. Flexibilidad

Del conjunto de tareas entrantes permite priorizar su realización según la necesidad del momento.

2.2.4 Tipos de Kanban

En el método Kanban empleado por Toyota, existe dos tipos de tarjetas Kanban (Salazar, 2019):

- **Kanban de retiro:** en esta tarjeta se detalla la relación y cantidad de producto que debe ser retirado del proceso terminado o a su vez del contenedor de producto para dar inicio al siguiente proceso. Las relaciones y detalles de esta tarjeta se visualizan en la Figura 4.

Proceso anterior:	Pulido de rebaba
Proceso posterior:	Ensamble de suela y capellada
Contenedor:	Almacén proceso 2
Referencia:	F-026-39
Nombre de la pieza:	Suelas de EVA
Tipo de calzado:	Sandalia talla 39 color azul Ref: 26
Capacidad del contenedor	Tipo de contenedor
40 unidades	A

Figura 4: Kanban o tarjeta de retiro

Recuperado de: <https://www.ingenieriaindustrialonline.com/lean-manufacturing/kanban-control-de-materiales-y-produccion/>

- **Kanban de producción:** este tipo de tarjeta se emplean principalmente para definir la relación y cantidad de producto que debe ser producido por un proceso. En la Figura 5 se muestra los elementos que componen la tarjeta de producción.

Proceso:	Ensamble de suela y capellada
Depositar piezas en:	Almacén proceso 2 (AI-2)
Referencia:	F-026-39
Nombre de la pieza:	Sandalia talla 39 color azul Ref: 26
Cantidad a producir	
40 unidades	

Figura 5: Kanban o tarjeta de producción

Recuperado de: <https://www.ingenieriaindustrialonline.com/lean-manufacturing/kanban-control-de-materiales-y-produccion/>

2.3 Metodología ágil Kanban en el desarrollo de software

El panorama del software cambia constantemente, debido a las nuevas necesidades de los usuarios, se ve en la necesidad de implementar nuevas metodologías ágiles como Kanban. Otro factor que ha contribuido a la adopción de Kanban es el Internet, debido a la facilidad de distribución de software y la descarga de actualizaciones. Esto ha acelerado el proceso de desarrollo de software hasta el punto en que la velocidad se ha convertido en un importante factor competitivo.

El padre y fundador de Kanban, David J. Anderson, aplica inicialmente la metodología Kanban en muchas empresas pertenecientes o afiliadas a Microsoft Corporation como son XIT y Corbis. La información sobre el proceso de desarrollo de productos de Donald Reinersen mejora y complementa aún más esta información. Codificando analíticamente muchos enfoques de gestión lean. El movimiento Kanban se da inicio tras la publicación de libro “Kanban. Successful Evolutionary Change for your Technology Business” que fue publicado por David J. Anderson en el año 2009. Posteriormente fundaría Lean Kanban University, rebautizada como Kanban University desde 2019, universidad que ofrece cursos formales en el método Kanban (Coloma, 2020).

Los años de pruebas, experimentación y colaboración de figuras líderes en Lean y Ágil como David J. Anderson, Dan Vacanti, Darren Davis, Corey Ladas, Dominica DeGrandis, Rick Garber y otros, a principios del año 2007 da como resultado lo que se conoce en la actualidad como método Kanban (Kanbanize, 2021).

2.3.1 Principios

Son cuatro los principios básicos para la implementación de la metodología Kanban, los mismo que son guía para el equipo de trabajo (Martins, 2022). Estos principios son enumerados a continuación:

1. **Empezar con lo que se hace ahora**

A diferencia de otros procesos ágiles, la flexibilidad que brinda Kanban permite implementar a cualquier proceso o flujo de trabajo y se adapta con facilidad a las prácticas de equipo de trabajo.

2. **Comprometidos con la búsqueda e implementación de cambios incrementales y evolutivos**

Los cambios grandes y repentinos son tan dañinos, llegando hasta el punto que el sistema nuevo no funciona correctamente. Con Kanban se fomenta la mejora continua y se buscan cambios incrementales con el objetivo de mejorar los procesos del equipo de trabajo y que estos procesos evolucionen con el tiempo.

3. **Respetar los procesos, los roles y las responsabilidades actuales**

Al implementar Kanban no es recomendable realizar modificaciones en el sistema de trabajo establecido, ya que se perderían excelentes elementos del proceso. Kanban no integra roles y funciona perfectamente con la estructura actual de tu equipo.

4. **Impulsar el liderazgo en todos los niveles**

Kanban fomenta la mejora continua, en el que los cambios pueden provenir de cualquier dirección. La participación activa de los miembros del equipo con nuevas formas e ideas para lograr la evolución de los procesos, es alentado por Kanban.

2.3.2 Prácticas de Kanban

El método Kanban cuenta con seis prácticas (Costa, 2021), las mismas que se describen a continuación:

- **Visualización**

Se refiere a la representación gráfica del proceso de desarrollo de software utilizando un tablero Kanban ya sea físico o digital. Por otro lado, la transparencia de los procesos, estados, recursos e ítems del proyecto es importante para llevar una mejor gestión. Las visualizaciones ayudan a los miembros del equipo a comprender el progreso del proyecto e identificar los cuellos de botella en el proceso de desarrollo.

- **Limitar el WIP**

Para un óptimo desarrollo del proyecto es fundamental definir la cantidad de tareas a realizar en la columna de trabajo en proceso. Una cantidad excesiva de tareas en este proceso puede conllevar a los famosos cuellos de botella, los problemas que se generan a raíz de esto es que el equipo de trabajo pierde el foco de las tareas que tienen prioridad, lo que conlleva a un posible cambio de tareas.

Varios son los beneficios que obtiene el equipo de trabajo tras definir la cantidad de tareas a realizar en cada proceso, como uno de los beneficios se menciona que el equipo se evita el desperdicio de tiempo e ineficiencia, otro de los beneficios que se

obtiene es asegurar la capacidad del sistema para la óptima realización de cada una de las tareas. Además, ayuda a identificar problemas en el flujo de trabajo.

- **Gestionar el flujo**

Una vez que se define el flujo de trabajo, el equipo puede ver el estado de cada tarea de una manera más intuitiva, lo que le permite comprender de la mejor manera el estado del proyecto. El objetivo de la gestión de procesos es mantener un ritmo adecuado de los procesos y mejorarlos, lo que se logra mediante el uso y análisis de mediciones, la aplicación de estrategias adecuadas y la implementación de mejoras.

- **Hacer explícitas las políticas**

Se refiere a la definición y comunicación clara de las reglas y procedimientos para la gestión y actualización del tablero Kanban. Esto incluye las reglas para las tarjetas en el tablero, como cuándo y cómo mover las tarjetas entre columnas y cómo manejar los cambios de tema. También puede incluir disposiciones para la asignación de tareas, la priorización de tareas y la gestión de problemas y riesgos.

El objetivo de una política clara es garantizar que todos los miembros del equipo entiendan y sigan el mismo flujo de trabajo, lo que ayuda a garantizar un desarrollo de proyectos eficiente y productivo. También facilita la comunicación y la colaboración entre los miembros del equipo porque todos usan el mismo conjunto de reglas y procedimientos.

- **Implementar ciclos de retroalimentación**

Los bucles de retroalimentación provienen del término inglés "feedback" y se refieren a mecanismos que regularmente involucran evaluar y mejorar el proceso de desarrollo de software. Esto incluye revisar y analizar el tablero Kanban para identificar debilidades, cuellos de botella y áreas que necesiten de mejoras e implementar cambios para abordarlos.

Los ciclos de retroalimentación pueden ser diarios, semanales o mensuales, según el proyecto y el equipo, y pueden incluir la revisión de las métricas del proyecto, la discusión de problemas y desafíos, y la planificación de acciones para mejorar el proceso. La retroalimentación también puede incluir cambios en las políticas del tablero Kanban para mejorar la eficiencia y la eficacia del proceso.

La práctica de implementar circuitos de retroalimentación es importante para garantizar que el proceso de desarrollo de software evolucione y mejore constantemente para satisfacer las necesidades del proyecto y del equipo. También ayuda a garantizar que los proyectos estén encaminados para cumplir con los objetivos y plazos establecidos.

- **Mejorar la colaboración y evolución experimental**

Se refiere a la implementación de estrategias y técnicas para promover la colaboración y la innovación entre los miembros del equipo. Esto incluye fomentar la comunicación abierta y la retroalimentación continua, y adoptar un enfoque de prueba y error para probar nuevas ideas y tecnologías.

Las prácticas avanzadas de desarrollo colaborativo y experimental ayudan a garantizar que los equipos trabajen de manera eficaz y eficiente y estén abiertos a

nuevas ideas y enfoques para mejorar el proceso de desarrollo de software. También ayuda a garantizar que el proyecto evolucione y mejore constantemente para satisfacer las necesidades del proyecto y del equipo.

2.3.3 Tablero Kanban

Los sistemas Kanban son visualizados típicamente mediante los tableros Kanban. En todos los tableros el recorrido es de izquierda a derecha, las nuevas tareas son ingresadas al tablero por la izquierda y una vez completadas salen por la derecha para ser finalmente entregadas al cliente (Kanban University, 2021).

Es una herramienta de visualización que brinda al equipo de trabajo una descripción general del estado actual del proyecto. Con esta herramienta el equipo tiene la capacidad de conocer el estado de cada tarea y realizar un seguimiento de su desarrollo en las diferentes etapas del proceso (Rocha, 2021).

El tablero Kanban (ver Figura 6) además de permitir visualizar en qué estado se encuentra cada tarea, este facilita la comunicación entre los miembros del equipo de trabajo y brinda la ayuda necesaria para dar seguimiento al flujo de trabajo (Stsepanets, 2022).



Figura 6: Tablero Kanban

Recuperado de: <https://agileisnow.com/2021/02/21/por-que-usar-el-tablero-kanban/>

Tener la posibilidad de visualizar el flujo de las tareas en diferentes estados, permite no solo tener una visión más amplia y clara del estado, sino también del objetivo del proyecto, el éxito del tablero Kanban se debe a que este aprovecha el poder de la información visual. Como lo señala los estudios realizados por 3M en el cual menciona que el cerebro humano procesa 60000 veces más rápido información visual que texto. La información visual provee

de la mayor cantidad de datos que llegan al cerebro, esto de acuerdo a Eric Jensen (Alcaraz, 2020).

Componentes del tablero Kanban (Ver Figura 7)

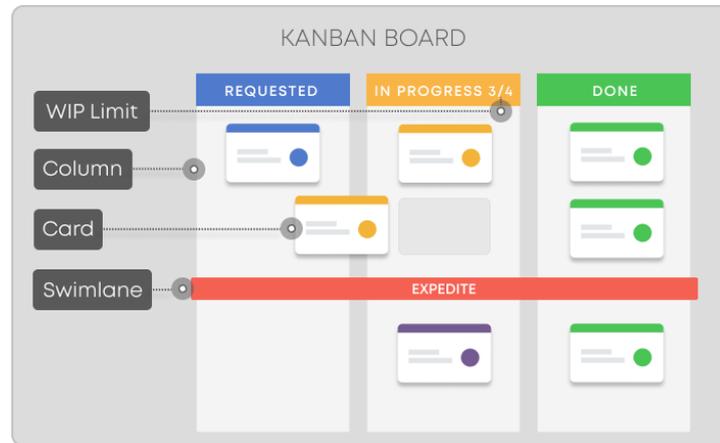


Figura 7: Componentes del tablero Kanban

Recuperado de: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban>

Los componentes y características principales del tablero Kanban son: tarjetas, columnas, límites de trabajo en curso. Este conjunto de características y componentes brinda al equipo la capacidad de gestionar de manera eficaz el flujo de trabajo (Donetonic, 2022). A continuación, se describe cada una de ellas:

- **Tarjetas**
El término tarjetas Kanban es utilizado para describir la representación visual del trabajo. En cuanto a la información que debe poseer una tarjeta Kanban es principalmente el título, descripción y el nombre del propietario. Además, esta puede contener información adicional como la fecha de entrega o la estimación del tiempo de duración de la actividad.
- **Columnas**
Tanto las fases del proyecto como del proceso son divididas en columnas verticales. El número de columnas lo define el equipo en base al tipo de proyecto y el alcance. Las columnas base del tablero Kanban son: To-do, Doing, Done.
- **Límites de trabajo en curso**
Limitar la cantidad máxima de tareas a realizar en cada etapa del flujo de trabajo permite culminar las tareas en tiempos más cortos y evita que se generen los cuellos de botella. Para aumentar la fluidez del flujo de trabajo, a las tareas complejas se deben considerar desglosar.

Funcionamiento del tablero Kanban

En primera instancia se define las tareas a realizar. Las tarjetas con las tareas son colocadas en la primera columna de la izquierda y este se mueve a la siguiente columna de la derecha cuando es completada en la etapa anterior. Es este mecanismo el que mantiene el flujo de

trabajo lo más fluido posible y mantiene al equipo informado sobre cambios y nuevas tareas. Al visualizar el trabajo en un tablero Kanban la productividad aumenta, mejora la administración de recursos y evita el desorden en la realización de las tareas o actividades del trabajo (Stsepanets, 2022).

Tipos del tablero Kanban

- **Tablero Kanban tradicional**

Con el inicio de Kanban las pizarras físicas eran utilizadas y consideradas como los primeros tableros Kanban, para una mejor visualización de las tareas se utilizaron tarjetas de colores. Debido a la facilidad de su implementación, el método Kanban se popularizó en equipos de desarrollo de software (Donetonic, 2022).

La principal ventaja de contar con este tipo de tableros es la disponibilidad, ya que no es necesario contar con un ordenador ni mucho menos internet. En cualquier momento se realiza los ajustes necesarios y se discute los puntos de interés entre los miembros del equipo de trabajo (Stsepanets, 2022)

- **Tablero Kanban online**

A medida que Kanban fue popularizándose entre los desarrolladores e ingenieros de software, aparecieron las pizarras virtuales. Este tipo de tableros son utilizados para los equipos de trabajo que se encuentran ubicados en lugares diferentes. Las ventajas de este tipo de tablero es la velocidad de configuración, facilidad de compartir entre los miembros, además, no es necesario reunirse ya que pueden dejar comentarios en las tareas, las mismas que son visibles para el resto del equipo de trabajo en cualquier momento (Stsepanets, 2022).

2.3.4 Herramientas Kanban

Permiten acceder a la información de manera remota y la seguridad es alta en comparación a los tableros tradicionales basados en papel o pizarra, la ventaja principal de utilizar estas herramientas en la gestión de proyectos es la mejora continua del flujo de trabajo definido (Salesdorado, 2022).

Las herramientas Kanban mejoran el proceso de gestión del flujo de trabajo y permite alcanzar de manera eficiente los objetivos diarios con el mínimo esfuerzo. Dependiendo del proyecto y los requisitos del equipo existen varias herramientas Kanban que se ajustan a las necesidades de cada proyecto (Medewar, 2022).

Las mejores herramientas Kanban en línea para la gestión de proyecto son: Trello, Kanbanpad, TargetProcess, Lean Kit, Kanbanize, Kanban Flow (SOLVING AD HOC, 2018).

2.4 Norma ISO/IEC 25000

Los autores (Calero, Moraga, & Piattini, 2010) mencionan que el objetivo básico de un modelo de calidad es especificar y evaluar la calidad de un producto de software a través de mediciones internas y directas de las propiedades del programa o mediciones externas e indirectas del rendimiento del sistema.

La norma ISO/IEC 25000 es reconocida también por el nombre de SQuare (Requisitos y Evaluación de Calidad de Productos de Software), esta norma surge y es la base para crear modelos, métricas, procesos y herramientas que se emplean para la evaluación de la calidad de un producto software, esta evaluación se realiza por medio de los requerimientos. Esta norma tiene como objetivo principal la creación de un marco de trabajo para evaluar la calidad del software (Roa, Morales, & Gutiérrez, 2015).

La evolución de normas anteriores como las normas ISO/IEC 9126 que describe particularidades de un modelo de calidad del software como producto y la norma ISO/IEC 14598 que trata específicamente del proceso de evaluación del producto software, ha dado como resultado la norma ISO/IEC 25000. La familia de normas ISO/IEC 25000 está compuesta por cinco divisiones (ver Figura 8).



Figura 8: División de la Familia ISO/IEC 25000

Recuperado de:

https://repositorio.upao.edu.pe/bitstream/20.500.12759/8136/1/REP_ING.SIST_ESTEFANY.MINCHOLA_EVALUACIÓ_N.CALIDAD.SISTEMA.INTEGRADO.ADMINISTRACIÓN.FINANCIERA.MUNICIPALIDAD.DISTRITAL.CALAMARCA.ISO.IEC.25000.2021.pdf

2.4.1 ISO/IEC 2501n - Modelo de calidad

En la división ISO/IEC 2501n se presentan modelos de calidad que incluyen características para la calidad interna, externa y el uso del producto software. Esta división está conformada

principalmente por dos normas las cuales son: ISO/IEC 25010 e ISO/IEC 25012 (Ramírez, 2018).

- **ISO/IEC 25010 - Modelos del sistema y calidad del software**

Los autores Roa, Morales y Gutiérrez (2015), mencionan que la norma ISO/IEC 25010 detalla el modelo de la calidad del producto software y está conformada por ocho características (ver Figura 9) que se deben tomar en cuenta a la hora de evaluar el producto software.



Figura 9: Características del modelo de calidad

Recuperado de: https://www.mvconsulting.com.ar/novedades_248

- **ISO/IEC 25012 - Modelo de calidad de datos**

Esta norma describe un modelo general para la calidad de datos, es aplicable para datos que se encuentran almacenados de forma estructurada y son parte de un sistema de información (Ramírez, 2018).

2.4.2 Característica - Usabilidad

La usabilidad hace referencia a la capacidad que posee un sistema informático para ser aprendido, entendido y usado, además, con esta característica se mide el grado de aceptación por parte del usuario en relación a su aspecto visual. La evaluación general lo llevan a cabo los usuarios finales del sistema (Ortega, 2018).

2.4.3 Subcaracterística - Facilidad de aprendizaje

El usuario comprende de manera rápida y sencilla el funcionamiento del producto software (López, 2021). En otras palabras, es la capacidad del producto que permite al usuario aprender la aplicación o software.

CAPÍTULO III. METODOLOGIA

La presente investigación fue desarrollada desde un enfoque cuantitativo ya que, se evaluó la usabilidad - criterio facilidad de aprendizaje del software educativo de vocabulario básico de inglés, el mismo que se desarrolló aplicando la metodología Kanban.

3.1 Tipo y diseño de investigación

3.1.1 Según la fuente de investigación

La investigación es de tipo bibliográfica ya que se recopiló e interpretó la información obtenida de diversas fuentes secundarias como libros, artículos, tesis y otros documentos.

3.1.2 Según el objeto de estudio

Investigación de tipo aplicada ya que dio como resultado un producto de tipo tecnológico denominado software educativo para el aprendizaje de vocabulario básico de inglés.

3.2 Identificación de Variables

3.2.1 Variable Independiente

Software educativo aplicando Metodología Kanban.

3.2.2 Variable dependiente

Evaluar la facilidad de aprendizaje del software educativo de vocabulario básico de inglés.

3.2.3 Operacionalización de variables

En la Tabla 2 se presenta la operacionalización de variables.

Tabla 2: Operacionalización de variables

Problema de investigación	Tema	Objetivos	Variables	Conceptualización	Dimensión	Indicadores
¿Es necesario el desarrollo de un software educativo aplicando la metodología KANBAN, para apoyar el aprendizaje de vocabulario básico de inglés?	Desarrollo de un software educativo aplicando la metodología KANBAN, para el aprendizaje de vocabulario básico de inglés	<p>General:</p> <p>Implementar un software educativo aplicando la metodología kanban, para el aprendizaje de vocabulario básico de inglés.</p> <p>Específicos:</p> <ul style="list-style-type: none"> • Analizar la metodología Kanban y el uso de tablero Kanban para el control de actividades de la fase de desarrollo. • Desarrollar un software educativo aplicando la metodología Kanban. 	<p>Independiente</p> <ul style="list-style-type: none"> • Software educativo aplicando Metodología a Kanban <p>Dependiente</p> <ul style="list-style-type: none"> • Evaluar la facilidad de aprendizaje del software educativo de vocabulario 	<p>Software educativo aplicando Metodología Kanban</p> <p>Evaluar la facilidad de aprendizaje del software educativo de vocabulario básico de inglés.</p>	<p>Software</p> <p>Metodología</p>	<ul style="list-style-type: none"> • Número de módulos implementados. • Ítem del proyecto Kanban. • Tiempo de espera. • Tiempo de bloqueo. • Porcentaje de eficiencia del Flujo. • Porcentaje de tareas de aprendizaje cumplidas. • Velocidad con que se realiza una tarea • Numero de errores que se cometen

-
- Evaluar la usabilidad del software educativo mediante la Norma ISO/IEC 25000, criterio facilidad de aprendizaje.

básico de inglés.

- Porcentaje de satisfacción del usuario

Elaborado por: Sahiry Zhuño, 2022.

3.3 Población y Muestra

3.3.1 Población

La población de estudio estuvo conformada por los alumnos del noveno EGB de la Unidad Educativa Básica Reino Quito de Tablo (ver Tabla 3):

Tabla 3: Población de estudio

EEB REINO DE QUITO DE TABLÓN	
Año Básico / Curso	Estudiantes
9vo EGB	13

Elaborado por: Sahiry Zhuño, 2022.

3.3.2 Muestra

Se trabajó con la totalidad de la población.

3.4 Técnicas e instrumentos de recolección de datos

3.4.1 Encuesta

Se aplicó esta técnica para evaluar el grado de facilidad de uso del software, además, conocer el porcentaje de tareas cumplidas.

3.5 Técnicas de análisis e interpretación de la información

Las encuestas realizadas fueron generadas de forma automatizada y óptima mediante la herramienta de encuestas Google Forms, con esta herramienta se pudo, analizar los resultados de cada pregunta y ordenarlos en hojas de cálculo y generar los gráficos en tiempo real.

Obtenido los datos de la encuesta se procesó y analizó la información y utilizando el método de la escala de Likert se evaluó la usabilidad del software, de la Norma ISO/IEC 25000, el criterio facilidad de aprendizaje. Los resultados proporcionados por este método fueron de manera rápida y eficaz.

3.6 Procedimientos

3.6.1 Análisis de requerimientos

Requerimientos funcionales ver Tabla 4.

Tabla 4: Requerimientos funcionales

REQUISITO	DESCRIPCIÓN
Crear cuentas de usuarios	El software permitirá crear cuentas de usuarios para acceder a los contenidos.
Visualizar actividades	El software permitirá visualizar al usuario las actividades que puede realizar.
Test de conocimiento	El software permitirá acceder a un test de conocimiento
Visualizar números de test realizados	El software permitirá visualizar el número de los test realizados

Elaborado por: Sahiry Zhuño, 2022.

Requerimientos no funcionales ver Tabla 5.

Tabla 5: Requerimientos no funcionales

REQUISITO	DESCRIPCIÓN
Funcionalidad	El software deberá responder a las peticiones solicitadas por el usuario
Integridad de datos	El sistema deberá mantener los datos íntegros del usuario, sin presentar alteraciones cuando el usuario los solicite
Usabilidad	El sistema deberá ser intuitivo, fácil de aprender y debe brindar ayuda en el manejo del mismo

Elaborado por: Sahiry Zhuño, 2022

3.6.2 Diseño

Interfaz gráfica de usuario

- Splash screen
En la Figura 10 se presenta el diseño del splash screen.



Figura 10: Diseño - Splash screen

Elaborado por: Sahiry Zhuño, 2022

- Ventana de inicio de sesión

En la Figura 11 se presenta el diseño de la ventana de inicio de sesión.



Figura 11: Diseño - ventana de inicio de sesión

Elaborado por: Sahiry Zhuño, 2022

- Ventana principal – Categorías

En la Figura 12 se visualiza el diseño de la ventana principal que contiene las diferentes categorías de las palabras e información del usuario.



Figura 12: Diseño - ventana menú principal

- Ventana Lección 1

El diseño de la ventana de lección 1 (ver Figura 13) consta de los siguientes componentes: imágenes, botones y textos.



Figura 13: Diseño - ventana lección 1 – Escuchar

- Ventana Lección 2

En la Figura 14 se presenta el diseño de la ventana de lección 2, que está conformada por los siguientes componentes: imágenes, texto, botones.

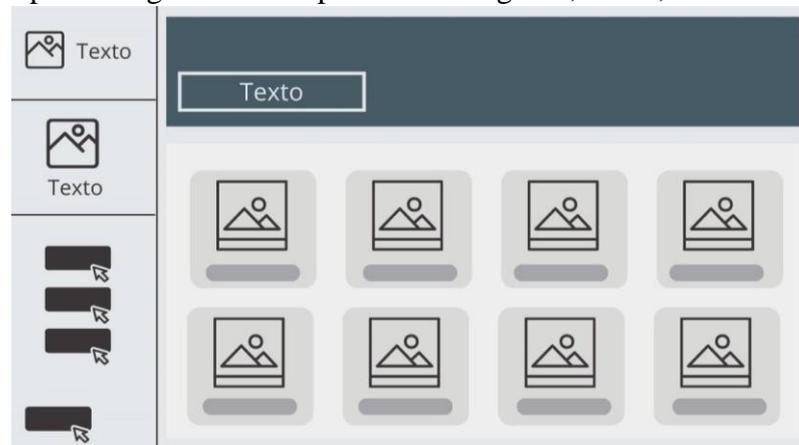


Figura 14: Diseño - ventana lección 2 – Escribir

- Ventana Lección 3

El diseño de la ventana de Lección 3 (ver Figura 15), lo conforman los componentes: imágenes, texto, botones.



Figura 15: Diseño - ventana lección 3 – Selección

Elaborado por: Sahiry Zhuño, 2022

Diagrama de caso de uso

En la Figura 16 visualizamos el diagrama de caso de uso. Como actor principal el usuario y se definió 5 acciones que debe realizar el sistema para dar respuesta de acuerdo a las solicitudes requeridas por el actor.

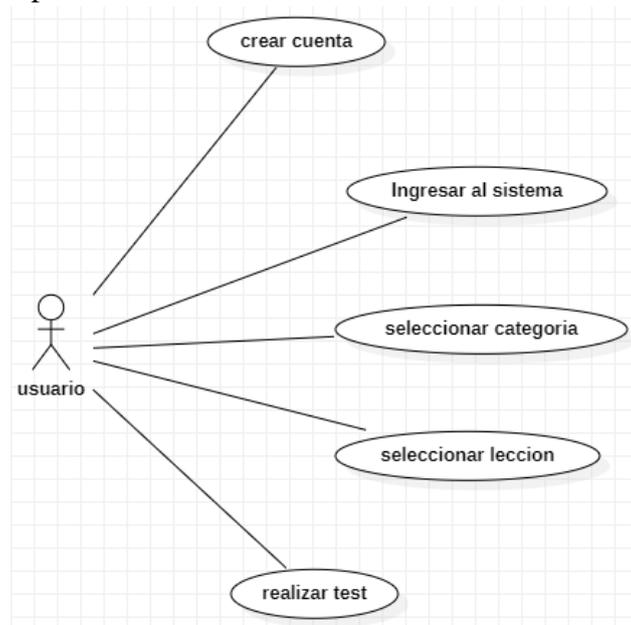


Figura 16: Diagrama de caso de uso

Elaborado por: Sahiry Zhuño, 2022

Diseño de arquitectura n capas

Para desarrollar el software se empleó la arquitectura n capa, está arquitectura es un modelo de diseño de software que separa la lógica en distintas capas acopladas, facilitando la escalabilidad y el mantenimiento del software, las capas que se definieron son: acceso a datos, lógica de negocio, presentación y entidades, ver Figura 17.

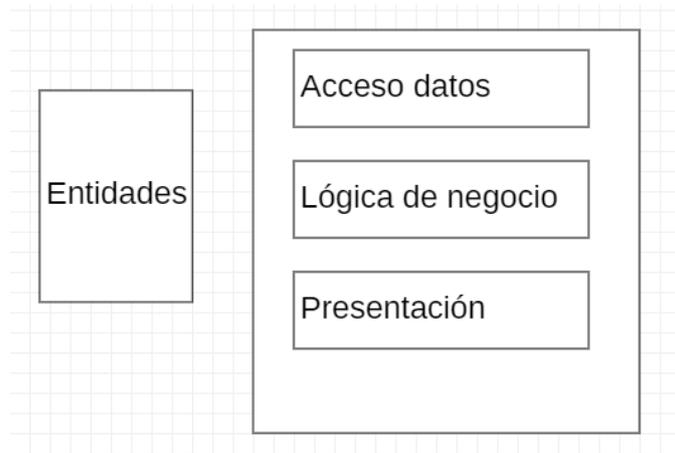


Figura 17: Arquitectura N capas

Elaborado por: Sahiry Zhuño, 2022

Modelado

- Diagrama Entidad Relación

Este diagrama está conformado por las entidades y las relaciones establecidas entre ellas, ver Figura 18.

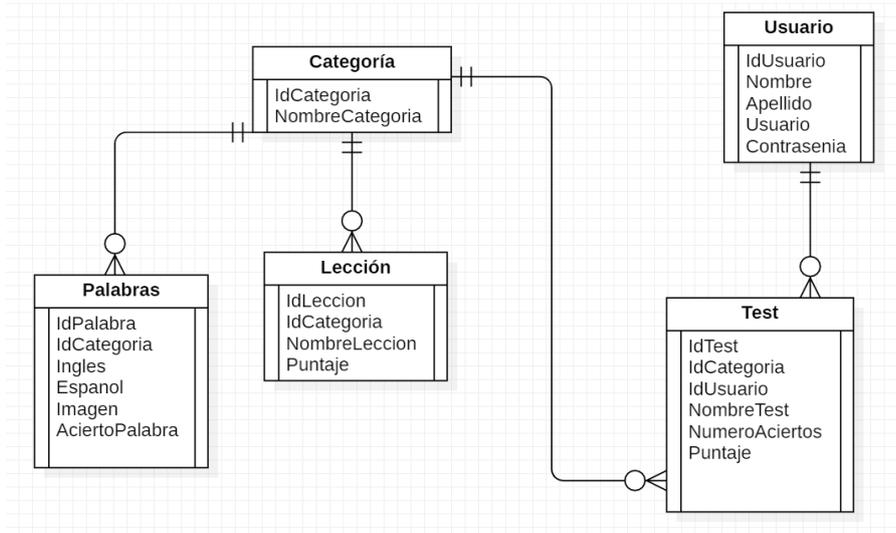


Figura 18: Diagrama entidad-relación (Diagrama ER)

Elaborado por: Sahiry Zhuño, 2022

- Diagrama Físico

El diagrama de la base de datos se muestra en la Figura 19. Las tablas con sus atributos y las relaciones son los componentes que conforman el diagrama físico de la base de datos implementado.

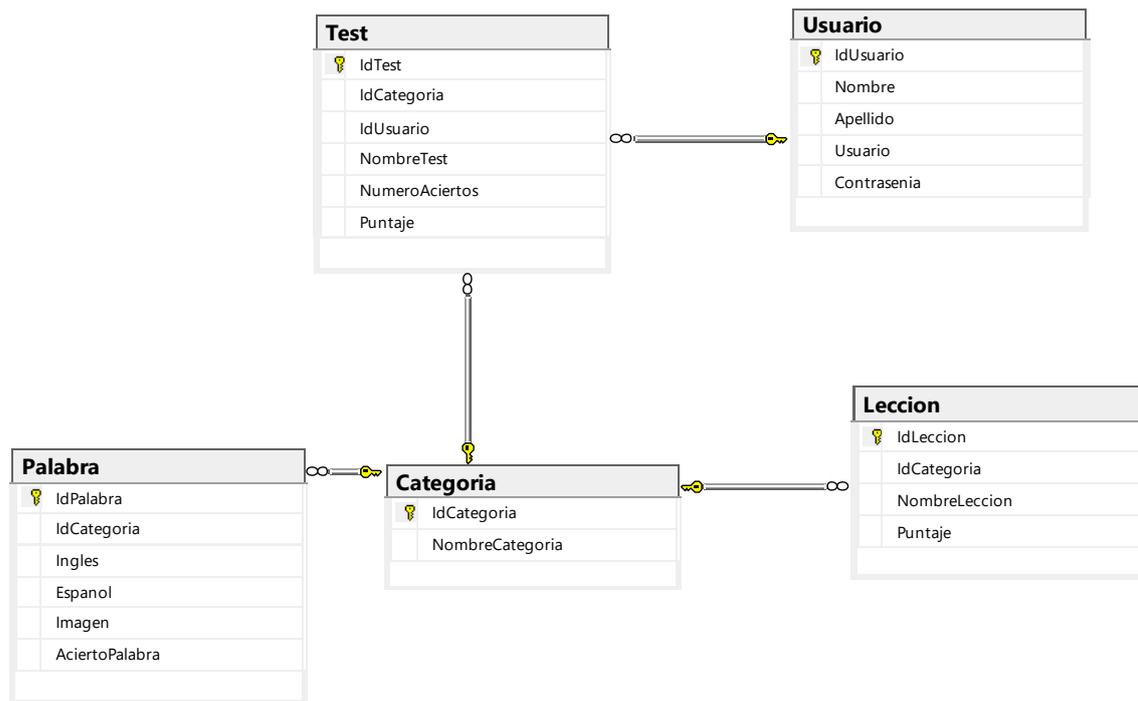


Figura 19: Diagrama de base de datos

Elaborado por: Sahiry Zhuño, 2022

3.6.3 Desarrollo del software educativo aplicando Kanban

Para del desarrollo del software educativo se definió las actividades a realizar en la fase de diseño y desarrollo de software, para la representación de las actividades mediante tarjetas se utilizó un tablero Kanban virtual, como es la herramienta basada en la web Trello.

Actividades del tablero Kanban

Se definió tres actividades para el tablero Kanban

1. **Lista de tareas:** en esta columna se añadió las tareas a realizar tanto en la fase de diseño como en la fase de desarrollo. La prioridad de cada tarea está definida por su posición, la tarea que se encuentra en la parte superior tiene una prioridad alta para su desarrollo.
2. **En proceso:** Las actividades que se encuentran en desarrollo se posicionan en esta columna. Para esta columna o actividad se estableció un máximo de 2 tareas, consiguiendo con esto evitar el famoso cuello de botella.
3. **Hecho:** a esta columna pasan todas las tareas que fueron completadas en la actividad anterior.

Para llevar un mejor control y gestión de las tareas se estableció un color diferente para cada columna o actividad. En la Figura 20 se visualiza las tarjetas que se encuentran en la columna de “lista de tareas” llevan un color celeste, mientras que las tarjetas en la columna de “En

proceso” pasan a llevar un color anaranjado. El color definido para la columna de “Hecho” fue el color azul, las actividades completadas tomarán este color.

Flujo de trabajo

El flujo inicia con la actividad del listado de tareas a realizar, pasando a la actividad en proceso para finalmente pasar a la última actividad de hecho. En la Figura 20 se muestra el flujo de trabajo establecido.



Figura 20: Flujo de trabajo

Elaborado por: Sahiry Zhuño, 2022

Las tareas definidas entre las fases de diseño y desarrollo del software fueron 9 tareas. De acuerdo a su prioridad se colocaron una encima de otro en la actividad “Lista de tareas” del tablero Kanban. La Figura 21 muestra el tablero Kanban en la etapa inicial con el listado de las tareas.

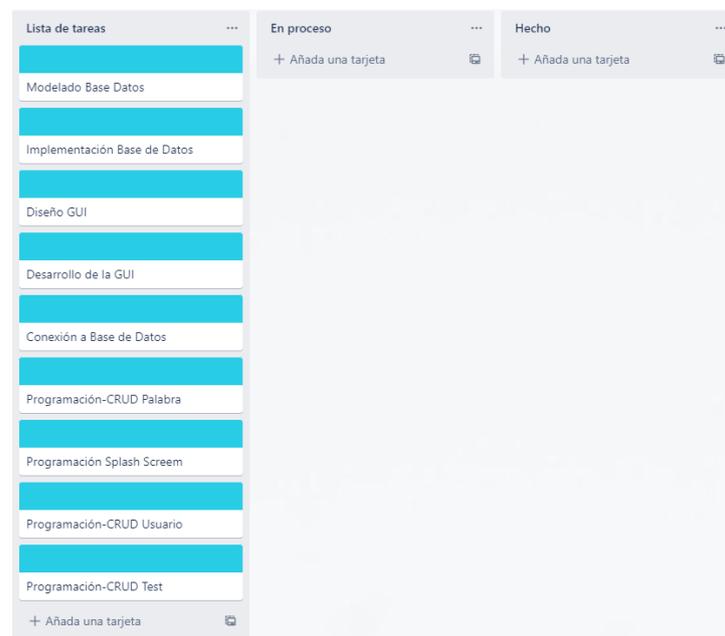


Figura 21: Etapa 1 - tablero Kanban

Elaborado por: Sahiry Zhuño, 2022

En la Figura 22 se muestra una instancia del tablero Kanban en la etapa dos, en esta etapa las tareas que se encuentran en la actividad de “En proceso” toman el color anaranjado. Como se ha mencionada anteriormente se estableció un máximo de 2 tareas para esta actividad.

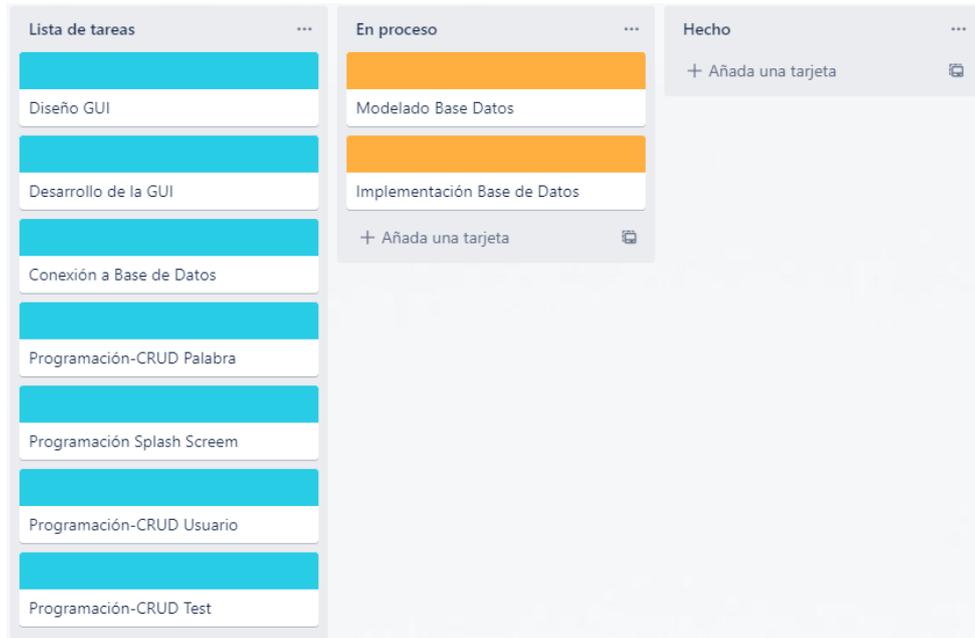


Figura 22: Etapa 2 - tablero Kanban

Elaborado por: Sahiry Zhuño, 2022

La Figura 22 muestra una instancia del tablero en el que se nota que las tarjetas de las tareas que se encuentran en desarrollo se mueven a la actividad de “En proceso”, estas tareas son: el modelado de la base de datos y la implementación de la base de datos. Las tarjetas en esta actividad son de color anaranjado.

En la etapa tres del tablero Kanban se muestra las actividades que fueron completadas. En esta columna las tarjetas de las tareas toman el color azul para indicar su estado. La Figura 23 refleja una instancia del tablero Kanban en la etapa 3 en el que se visualiza como las tareas que se encontraban en proceso ahora han pasado a la actividad de hecho, dando a entender que estas tareas fueron realizadas correctamente.

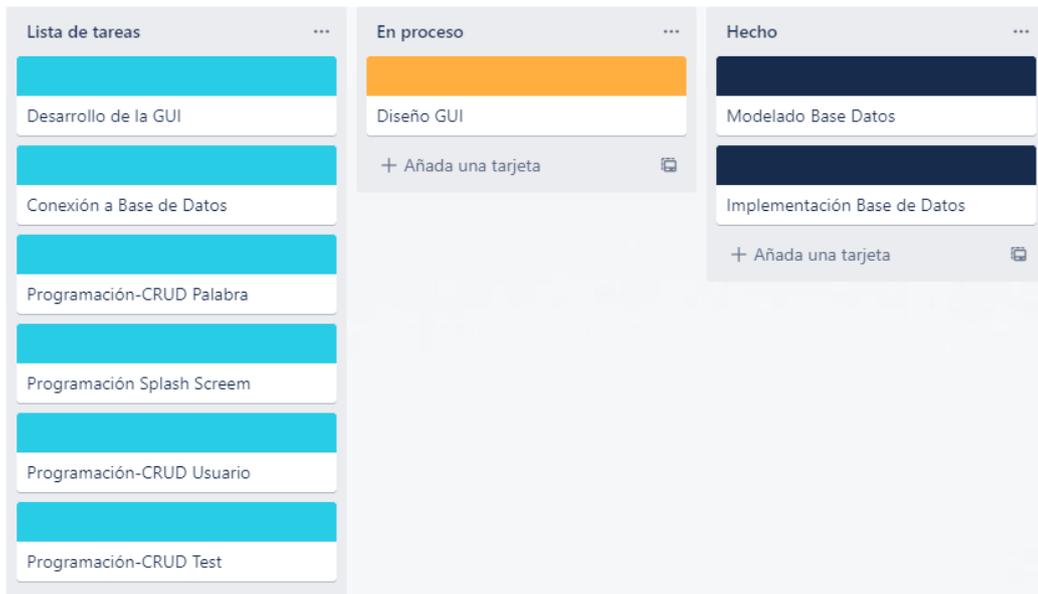


Figura 23: Etapa 3 - Tablero Kanban

Elaborado por: Sahiry Zhuño, 2022

Como se ha indicado anteriormente el desarrollo del software se realizó bajo la arquitectura n capas, debido que brinda la facilidad de mantenimiento. Las capas definidas se describen a continuación:

Capa de Acceso a Datos

En la capa de acceso se estableció la conexión con la base de datos. Dentro de esta clase se encuentra la función para establecer y validar la conexión, entre otras funciones. En la Figura 24 se visualiza parte del código de la clase denominada “ClsDataBase”.

```

#region Métodos privados
1 referencia
private void CrearConexionBaseDatos(ref ClsDataBase ObjDataBase)
{
    switch (ObjDataBase.NombreDB)
    {
        case "DB_SW_IN":
            ObjDataBase.ObjSqlConnection = new SqlConnection(
                Properties.Settings.Default.CadenaConexion_DB_SW_IN);
            break;
        default:
            break;
    }
}
3 referencias
private void ValidarConexionBaseDatos(ref ClsDataBase ObjDataBase)
{
    if (ObjDataBase.ObjSqlConnection.State == ConnectionState.Closed)
    {
        ObjDataBase.ObjSqlConnection.Open();
    }
    else
    {
        ObjDataBase.ObjSqlConnection.Close();
        ObjDataBase.ObjSqlConnection.Dispose();
    }
}
2 referencias
private void AgregarParametros(ref ClsDataBase ObjDataBase)
2 referencias
private void PrepararConexionBaseDatos(ref ClsDataBase ObjDataBase)
{
    CrearConexionBaseDatos(ref ObjDataBase);
    ValidarConexionBaseDatos(ref ObjDataBase);
}
1 referencia
private void EjecutarDataAdapter(ref ClsDataBase ObjDataBase)
{
    try
    {
        PrepararConexionBaseDatos(ref ObjDataBase);

        ObjDataBase.ObjSqlDataAdapter = new SqlDataAdapter(ObjDataBase.NombreSP, ObjDataBase.ObjSqlConnection);
        ObjDataBase.ObjSqlDataAdapter.SelectCommand.CommandType = CommandType.StoredProcedure;
        AgregarParametros(ref ObjDataBase);
        ObjDataBase.DsResultados = new DataSet();
        ObjDataBase.ObjSqlDataAdapter.Fill(ObjDataBase.DsResultados, ObjDataBase.NombreTabla);
    }
    catch (Exception ex)
    {
        ObjDataBase.MensajeErrorDB = ex.Message.ToString();
    }
}

```

Figura 24: Fracción de código –ClsDataBase

Elaborado por: Sahiry Zhuño, 2022

Capa de Entidades

- Palabra

En la capa de entidades se creó una clase que lleva la nomenclatura “Cls” más el nombre de la tabla, en esta clase se definió los atributos que posee la tabla palabra en la base de datos. Además, se definió los métodos que permiten el acceso a los atributos de la clase, adicional se definen dos atributos para el manejo de respuestas de la base de datos, ver Figura 25.

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Entidades.Palabras
{
    9 referencias
    public class ClsPalabra
    {
        #region Atributos privados
        private int _idPalabra, _idCategoria;
        private string _ingles, _espanol;
        private byte imagen;
        private bool _aciertoPalabra;

        //atributos de manejo de la base de datos
        private string _mensajeError, _valorEscalar;
        private DataTable _dtResultados;
        #endregion

        #region Atributos públicos
        3 referencias
        public int IdPalabra { get => _idPalabra; set => _idPalabra = value; }
        6 referencias
        public int IdCategoria { get => _idCategoria; set => _idCategoria = value; }
        3 referencias
        public string Ingles { get => _ingles; set => _ingles = value; }
        1 referencia
        public string Espanol { get => _espanol; set => _espanol = value; }
        1 referencia
        public byte Imagen { get => imagen; set => imagen = value; }
        1 referencia
        public bool AciertoPalabra { get => _aciertoPalabra; set => _aciertoPalabra = value; }
        9 referencias
        public string MensajeError { get => _mensajeError; set => _mensajeError = value; }
        1 referencia
        public string ValorEscalar { get => _valorEscalar; set => _valorEscalar = value; }
        13 referencias
        public DataTable DtResultados { get => _dtResultados; set => _dtResultados = value; }
        #endregion
    }
}

```

Figura 25: Código de la clase – ClsPalabra

Elaborado por: Sahiry Zhuño, 2022

- Usuario

En la capa de entidades se encuentra la clase denominada “ClsUsuario” que contiene los atributos que posee la tabla de usuario en la base de datos. Adicional, se definió los métodos que permiten el acceso a los atributos de la clase, se añadió dos atributos para el manejo de respuestas de la base de datos. En la Figura 26 se visualiza el código de esta clase.

```

using System.Data;

namespace Entidades.Usuarios
{
    13 referencias
    public class ClsUsuario
    {
        #region Atributos privados
        private int _idUsuario;
        private string nombre, apellido, usuario, contrasena;

        //atributos de manejo de la base de datos
        private string _mensajeError, _valorEscalar;
        private DataTable _dtResultados;

        #endregion

        #region Atributos publicos
        6 referencias
        public int IdUsuario { get => _idUsuario; set => _idUsuario = value; }
        5 referencias
        public string Nombre { get => nombre; set => nombre = value; }
        5 referencias
        public string Apellido { get => apellido; set => apellido = value; }
        7 referencias
        public string Usuario { get => usuario; set => usuario = value; }
        7 referencias
        public string Contrasena { get => contrasena; set => contrasena = value; }
        9 referencias
        public string MensajeError { get => _mensajeError; set => _mensajeError = value; }
        2 referencias
        public string ValorEscalar { get => _valorEscalar; set => _valorEscalar = value; }
        7 referencias
        public DataTable DtResultados { get => _dtResultados; set => _dtResultados = value; }

        #endregion
    }
}

```

Figura 26: Código de la clase – ClsUsuario

Elaborado por: Sahiry Zhuño, 2022

Capa Lógica de Negocios

- Palabra

En la capa lógica de negocios se encuentra la clase denominada “ClsPalabraNL” que contienen las funciones para leer, calificar y ejecutar. La función Read devuelve un conjunto de registros tras pasar como parámetro el código de la categoría. La función calificar realiza una comparación entre los registros de la tabla “Palabra” y el texto ingresado por el usuario tras pasar como parámetro el Id de la palabra y el texto a comparar. En la Figura 27 se muestra parte del código de la clase “ClsPalabraNL”.

```

#region CRUD Palabra
//Leer
4 referencias
public void Read(ref ClsPalabra ObjPalabra)
{
    ObjDataBase = new ClsDataBase()
    {
        NombreTabla = "Palabra",
        NombreSP = "[SP_Palabra_Read]",
        Scalar = false
    };
    ObjDataBase.DtParametros.Rows.Add("@IdCategoria", "4", ObjPalabra.IdCategoria);
    Ejecutar(ref ObjPalabra);
}

//Calificacion
1 referencia
public void Calificar(ref ClsPalabra ObjPalabra) {
    ObjDataBase = new ClsDataBase()
    {
        NombreTabla = "Palabra",
        NombreSP = "[SP_Palabra_Verificar]",
        Scalar = true
    };
    ObjDataBase.DtParametros.Rows.Add("@IdPalabra", "4", ObjPalabra.IdPalabra);
    ObjDataBase.DtParametros.Rows.Add("@Ingles", "17", ObjPalabra.Ingles);

    Ejecutar(ref ObjPalabra);
}
#endregion

#region Métodos privados
2 referencias
private void Ejecutar(ref ClsPalabra ObjPalabra)
{
    ObjDataBase.CRUD(ref ObjDataBase);
    if (ObjDataBase.MensajeErrorDB == null)
    {
        if (ObjDataBase.Scalar) {...}
        else
        {
            ObjPalabra.DtResultados = ObjDataBase.DsResultados.Tables[0];
            if (ObjPalabra.DtResultados.Rows.Count == 1)
            {
                foreach (DataRow item in ObjPalabra.DtResultados.Rows) {...}
            }
        }
    }
    else
    {
        ObjPalabra.MensajeError = ObjDataBase.MensajeErrorDB;
    }
}
#endregion
}

```

Figura 27: Código de la clase – ClsPalabraLN

Elaborado por: Sahiry Zhuño, 2022

- Usuario

En la capa lógica de negocios se encuentra la clase denominada “ClsUsuarioNL” que contienen las funciones para crear, leer, actualizar y borrar, en esta misma clase se encuentra la función de inicio de sesión. En la Figura 28 se muestra parte del código de esta clase.

```

using AccesoDatos.DataBase;
using Entidades.Usuarios;
using System;
using System.Data;

namespace Logica.Usuarios
{
    4referencias
    public class ClsUsuarioLN
    {
        #region Variables privadas
        private ClsDataBase ObjDataBase = null;

        #endregion

        #region Método Index
        0referencias
        public void Index(ref ClsUsuario ObjUsuario)
        {
            ObjDataBase = new ClsDataBase()
            {
                NombreTabla = "Usuario",
                NombreSP = "[SP_Usuario_Index]",
                Scalar = false
            };
            Ejecutar(ref ObjUsuario);
        }
        #endregion

        #region CRUD Usuario
        1referencia
        public void Create(ref ClsUsuario ObjUsuario)
        {
            ObjDataBase = new ClsDataBase()
            {
                NombreTabla = "Usuario",
                NombreSP = "[SP_Usuario_Create]",
                Scalar = true
            };
            ObjDataBase.DtParametros.Rows.Add(@"@Nombre", "17", ObjUsuario.Nombre);
            ObjDataBase.DtParametros.Rows.Add(@"@Apellido", "17", ObjUsuario.Apellido);
            ObjDataBase.DtParametros.Rows.Add(@"@Usuario", "17", ObjUsuario.Usuario);
            ObjDataBase.DtParametros.Rows.Add(@"@Contrasenia", "17", ObjUsuario.Contrasena);
            Ejecutar(ref ObjUsuario);
        }
        1referencia
        public void Read(ref ClsUsuario ObjUsuario)
        {
            ObjDataBase = new ClsDataBase()
            {
                NombreTabla = "Usuario",
                NombreSP = "[SP_Usuario_Read]",
                Scalar = false
            };
            ObjDataBase.DtParametros.Rows.Add(@"@IdUsuario", "4", ObjUsuario.IdUsuario);
            Ejecutar(ref ObjUsuario);
        }
    }
}

```

Figura 28: Fracción de código – ClsUsuarioLN

Elaborado por: Sahiry Zhuño, 2022

Capa de Presentación

- Splash Screen

Esta ventana se posiciona en el centro de la pantalla una vez ejecutado el programa, la duración de visualización es de 3 segundos. Este presenta el nombre del programa y por debajo se encuentra el slogan. La Figura 29 representa el diseño del splash screen ya implementado.



Figura 29: Splash screen

Elaborado por: Sahiry Zhuño, 2022

- Ventana de inicio de sesión y registro

La Figura 30 muestra la distribución de los componentes de la ventana de inicio de sesión, este contiene los campos para ingresar el nombre de usuario y la contraseña, el botón de inicio de sesión y la opción de crear una cuenta. A la derecha se visualiza parte del código.

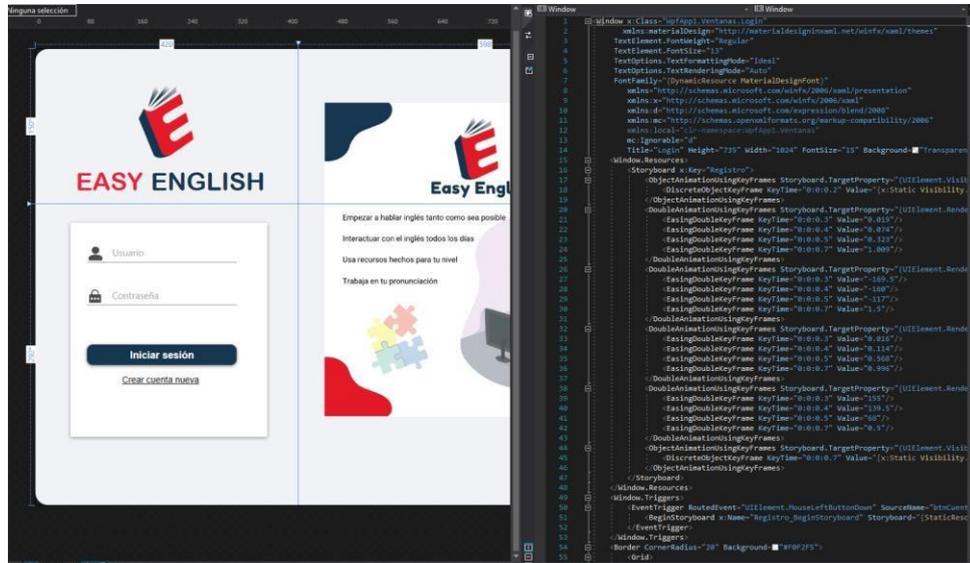


Figura 30: Interfaz y código - ventana de inicio de sesión y registro

Elaborado por: Sahiry Zhuño, 2022

- Ventana menú principal

Esta ventana contiene en el lateral izquierdo una imagen predeterminada para el usuario y sobre este la opción de editar la información básica del usuario, información del progreso y el botón para cerrar sesión. En la parte central se ubican las categorías que son opciones de selección. En la Figura 31 se visualiza parte de la interfaz y su código.

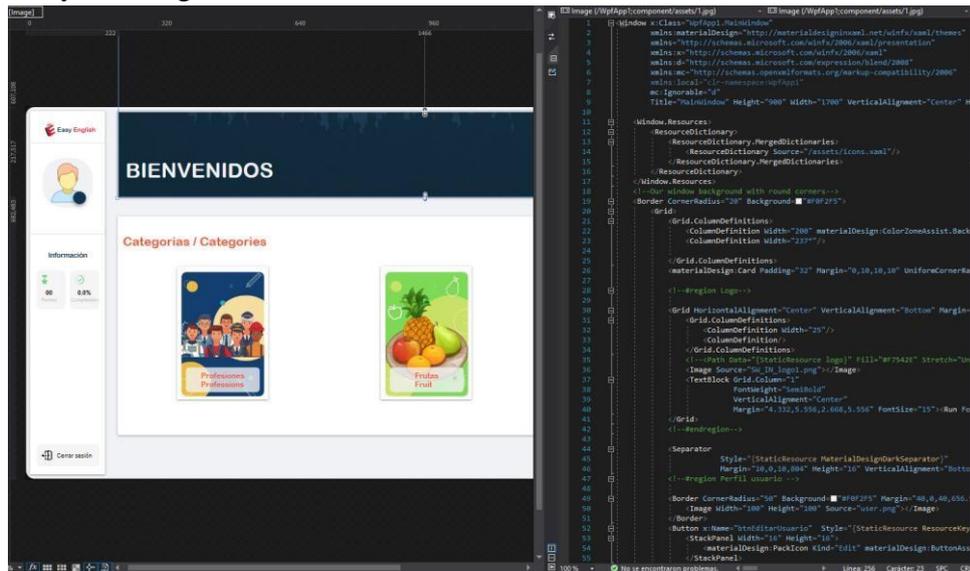


Figura 31: Interfaz y código - ventana menú principal

Elaborado por: Sahiry Zhuño, 2022

- Ventana lección 1 – Escuchar

Esta ventana contiene, en el lateral izquierdo información del usuario, las diferentes opciones de lecciones. En el centro de la ventana muestra un conjunto de imágenes con el nombre en inglés y español, el usuario tiene la capacidad de escuchar su pronunciación presionando sobre el botón de play que se ubica sobre cada imagen. En la Figura 32 se visualiza parte de la interfaz y el código.

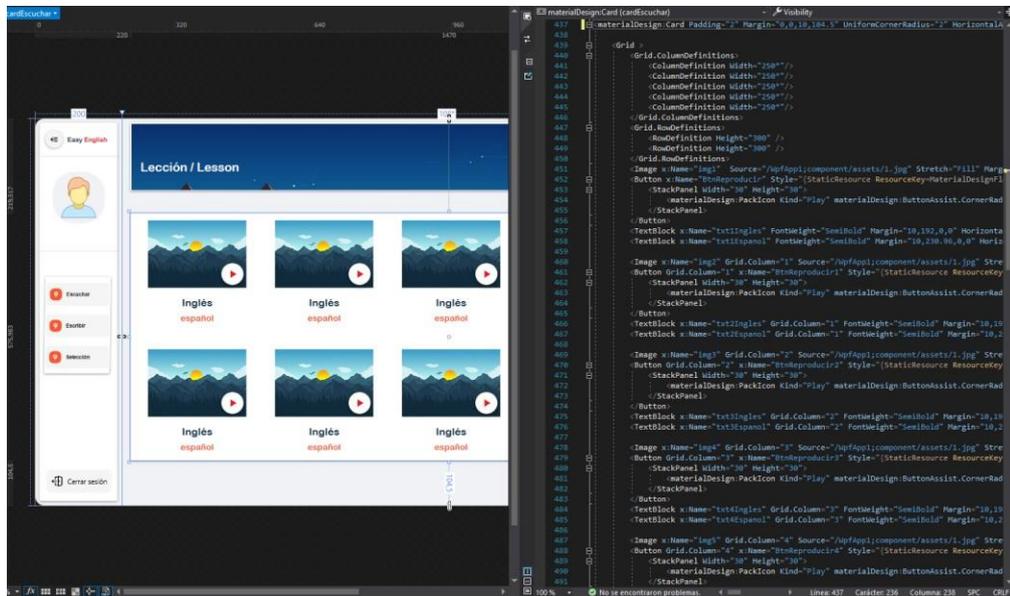


Figura 32: Interfaz y código - ventana lección 1

Elaborado por: Sahiry Zhuño, 2022

- Ventana lección 2

En el centro de la ventana se muestra un conjunto de imágenes con una entrada de texto cada una, el usuario en esta lección debe ingresar los nombres correspondientes a cada imagen y para finalizar presionar sobre el botón guardar progreso. El diseño y el código de esta ventana se muestra en la Figura 33.

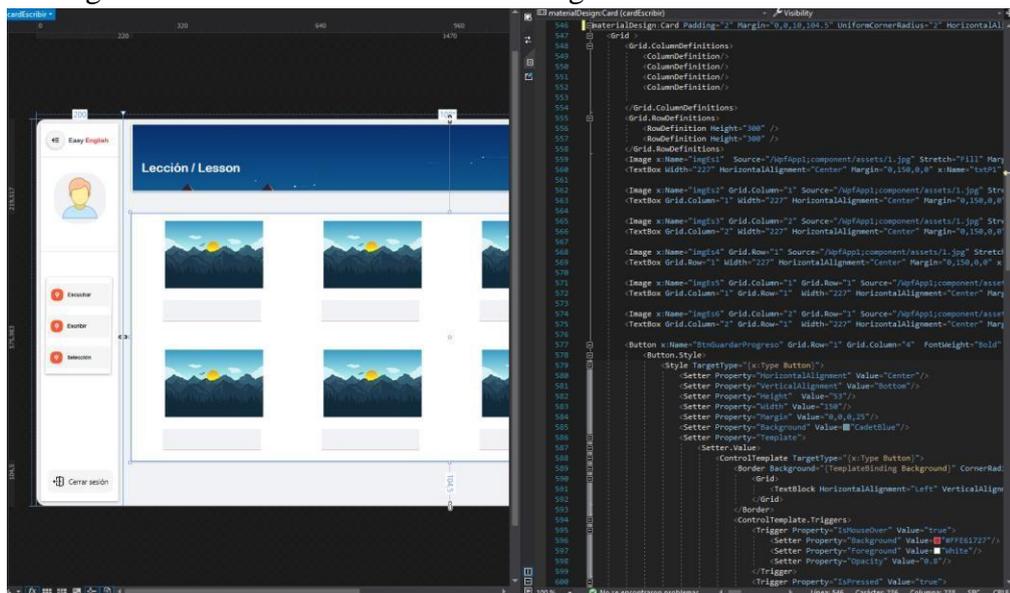


Figura 33: Interfaz y código - ventana lección 2

Elaborado por: Sahiry Zhuño, 2022

- Ventana lección 3

En esta lección el usuario debe seleccionar la imagen que corresponda con la palabra en ingles que se muestra a la izquierda de la pantalla. En la Figura 34 se muestra parte de la interfaz y del código.

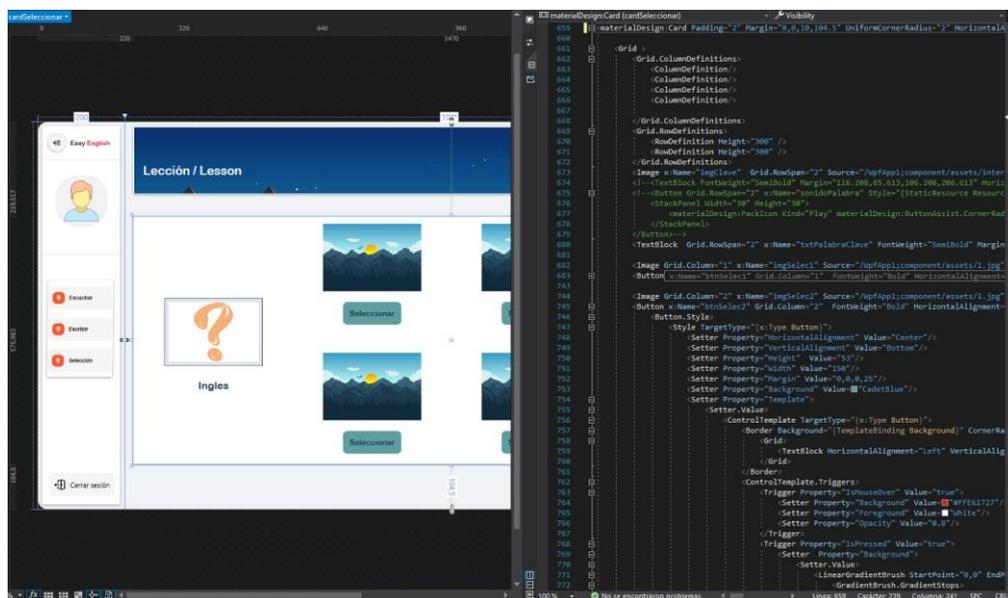


Figura 34: Interfaz y código - ventana lección 3

Elaborado por: Sahiry Zhuño, 2022

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

La aplicación de la metodología Kanban en el ámbito del desarrollo de software dio como resultado el software educativo para el aprendizaje de vocabulario básico de inglés, y finalmente se realizó la evaluación de la usabilidad el criterio facilidad de aprendizaje en base a la Norma ISO/IEC 25000.

Utilizando la herramienta Formularios de Google, se desarrolló una encuesta para evaluar el criterio facilidad de aprendizaje. Se aplicó una escala de Likert a cada respuesta, lo que nos permitió calcular y evaluar los resultados. Escalas de Likert establecidas, ver Tablas 6-9:

Tabla 6: Escala de Likert - Pregunta 1

Respuesta	Equivalencia
Muy difícil	1
Difícil	2
Regular	3
Fácil	4
Muy Fácil	5

Elaborado por: Sahiry Zhuño, 2022

Tabla 7: Escala de Likert - Pregunta 2

Respuesta	Equivalencia
Pésimo	1
Malo	2
Regular	3
Bueno	4
Excelente	5

Elaborado por: Sahiry Zhuño, 2022

Tabla 8: Escala de Likert - Pregunta 3 - 4

Respuesta	Equivalencia
Siempre	1
Casi siempre	2
A veces	3
Casi nunca	4
Nunca	5

Elaborado por: Sahiry Zhuño, 2022

Tabla 9: Escala de Likert - Pregunta 5

Respuesta	Equivalencia
------------------	---------------------

Más de 8 minutos	1
De 5 a 7 minutos	2
De 2 a 4 minutos	3
Menos de 1 minuto	4

Elaborado por: Sahiry Zhuño, 2022

4.1 Resultados de la encuesta de Evaluación de Usabilidad/Criterio facilidad de aprendizaje

¿Cómo calificaría la facilidad de uso del programa?

13 respuestas

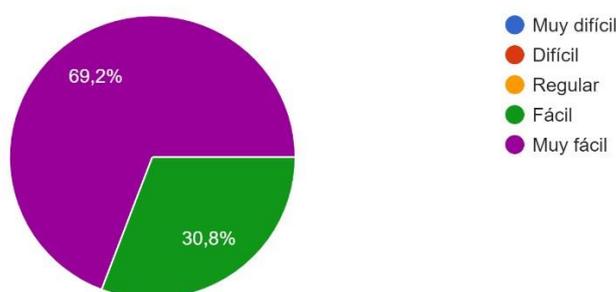


Figura 35: Pregunta 1 – Evaluación de usabilidad

Elaborado por: Sahiry Zhuño, 2022

Los resultados de la pregunta 1 son reflejados en la Figura 35, de acuerdo a la gráfica de resultados la mayoría de los encuestados indicó que la interacción con el software de aprendizaje de vocabulario básico de inglés es muy fácil.

¿Encuentra que el programa es intuitivo y fácil de entender?

13 respuestas

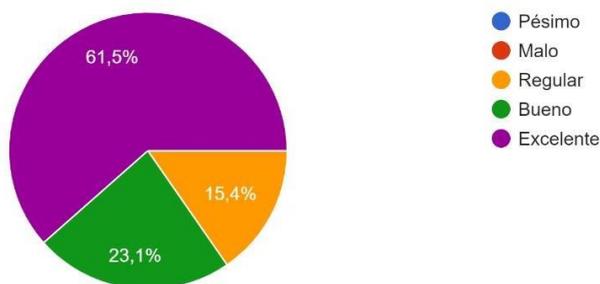


Figura 36: Pregunta 2 – Evaluación de usabilidad

Elaborado por: Sahiry Zhuño, 2022

Los resultados de la pregunta 2 se refleja en la Figura 36, donde la mayoría de los encuestados manifestaron que el software para el aprendizaje de vocabulario básico de inglés es intuitivo y fácil de entender.

¿Ha experimentado algún problema o error mientras utiliza el programa?

13 respuestas

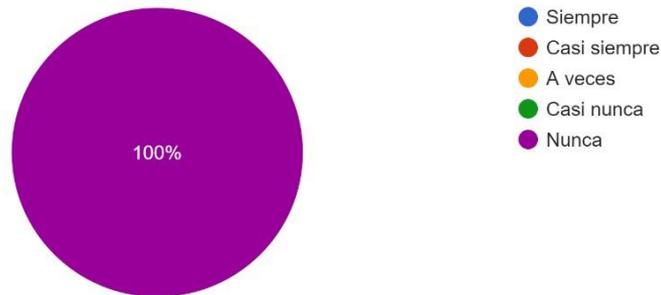


Figura 37: Pregunta 3 – Evaluación de usabilidad

Elaborado por: Sahiry Zhuño, 2022

Los resultados de la pregunta 3 se refleja en la Figura 37, donde el 100% de los encuestados manifiestan que nunca han experimentado problemas o errores mientras usan el software para el aprendizaje de vocabulario básico de inglés.

¿Tuviste que pedir ayuda a alguien más para aprender a utilizar el programa?

13 respuestas

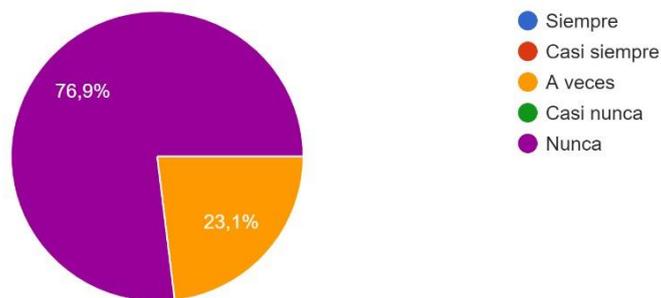


Figura 38: Pregunta 4 – Evaluación de usabilidad

Elaborado por: Sahiry Zhuño, 2022

En la Figura 38 se refleja los resultados de la pregunta 4, en el cual el 76.9% de los encuestados manifestaron que nunca pidieron ayuda para aprender a utilizar el programa de aprendizaje de vocabulario básico de inglés

¿Cuánto tiempo te llevó realizar una tarea del programa?

13 respuestas

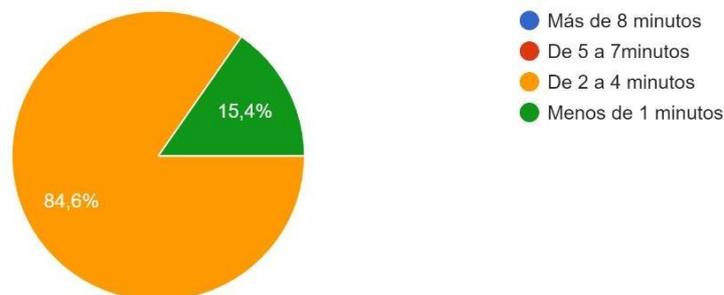


Figura 39: Pregunta 5 – Evaluación de usabilidad

Elaborado por: Sahiry Zhuño, 2022

En la Figura 39 se visualiza que la mayoría de los encuestados indicaron que el tiempo estimado para realizar una actividad en el software para el aprendizaje de vocabulario básico de inglés, está entre 2 a 4 minutos, este tiempo se pudo verificar con la consulta generada a la base de datos, ver Table 10.

Tabla 10: Reporte-tiempo

Categoría Palabra	Lección	Tiempo	Alumno
2	Listening	00:02:42.0000000	Cinthy Armijos
1	Selection	00:03:09.0000000	Stalin Romero
2	Writing	00:02:12.0000000	Alison Romero
2	Listening	00:03:51.0000000	Alejandra Liendres
1	Selection	00:03:20.0000000	Erika Carche
3	Selection	00:03:07.0000000	Gissela Ochoa
2	Listening	00:02:48.0000000	Valeria Armijos
3	Writing	00:03:35.0000000	Alex Villa
1	Listening	00:03:28.0000000	Jean Camacho
1	Listening	00:00:57.0000000	Cristian Toledo
3	Writing	00:00:52.0000000	Anthony González
2	Selection	00:03:17.0000000	Carlos Ramon

Elaborado por: Sahiry Zhuño, 2022

4.2 Discusión de los resultados Evaluación de Usabilidad – Criterio facilidad de aprendizaje

De acuerdo a los resultados obtenidos de la encuesta, la mayoría siendo el 69.2% del total de los encuestados manifestaron que el uso del programa es muy fácil. Se notó que el diseño del software es eficiente y se adapta a los usuarios finales, permitiendo una interacción sencilla que facilita el aprendizaje, como lo manifestaron el 61.5% de los encuestados.

En referencia al tiempo empleado en realizar una tarea del programa, el 84.6% de los encuestados manifestaron que el tiempo utilizado está entre 2 a 4 minutos (ver Tabla 11), siendo este un tiempo óptimo para realizar la tarea. Los usuarios que interactuaron con el software cumplieron con la totalidad de las tareas de aprendizaje propuestas.

Tabla 11: Resultado de encuesta – Tiempo empleado

Criterio	Pregunta	Equivalencia Escala de Likert	Porcentaje obtenido	Equivalencia de encuestados	Total de encuestados
Facilidad de aprendizaje	¿Cuánto tiempo te llevó realizar una tarea del programa?	1			13
		2			
		3	84.6%	11	
		4	15.4%	2	

Elaborado por: Sahiry Zhuño, 2022

El 100% de los encuestados manifestaron que no han experimentado ningún problema o error mientras utilizaban el programa.

En Tabla 12 se presenta los resultados de la encuesta de evaluación de usabilidad – criterio facilidad de aprendizaje mediante las escalas de Likert definidas.

Tabla 12: Resultados de la encuesta

Criterio	Pregunta	Equivalencia Escala de Likert	Porcentaje obtenido	Equivalencia de encuestados	Total de encuestados
-----------------	-----------------	--------------------------------------	----------------------------	------------------------------------	-----------------------------

Facilidad de aprendizaje	¿Cómo calificaría la facilidad de uso del programa?	5	69.2%	9	13
	¿Encuentra que el programa es intuitivo y fácil de entender?	5	61.5%	8	13
	¿Ha experimentado algún problema o error mientras utiliza el programa?	5	100%	13	13
	¿Tuviste que pedir ayuda a alguien más para aprender a utilizar el programa?	5	79.2%	10	13

Elaborado por: Sahiry Zhuño, 2022

En base a la investigación y los resultados se evidenció que el software educativo desarrollado aplicando la metodología Kanban, significó un apoyo en el proceso de aprendizaje de vocabulario básico de inglés.

CAPÍTULO V. CONCLUSIONES y RECOMENDACIONES

5.1 Conclusiones

- La metodología Kanban ha demostrado ser una herramienta útil en el desarrollo de software, ya que permite que los equipos de desarrollo trabajen de manera más productiva y transparente durante todo el proceso de desarrollo debido a que esta metodología ofrece un enfoque visual y colaborativo para la gestión de proyectos. Al implementar Kanban en el desarrollo de software, los equipos pueden identificar y resolver cuellos de botella, mejorar la calidad del producto final y reducir los tiempos de entrega. Además, el método Kanban promueve la cultura de la mejora continua al brindar a los miembros del equipo la libertad de experimentar con diferentes formas de trabajar y adaptar sus prácticas según sea necesario.
- Mediante la aplicación de la metodología Kanban en el desarrollo del software para el aprendizaje de vocabulario básico de inglés, se obtuvo beneficios que permitieron el desarrollo satisfactorio del mismo. Al aplicar Kanban, se logró una gestión más eficiente y efectiva de las actividades de las distintas fases de desarrollo, lo que permitió concentrarse en las actividades de mayor prioridad y completar el trabajo de manera más rápida.
- El software posee una interfaz intuitiva y fácil de aprender, y la funcionalidad se ajusta a las necesidades de los usuarios finales, según los resultados de la encuesta de evaluación de usabilidad, criterio de facilidad de aprendizaje, basado en la norma ISO/IEC 25000.

5.2 Recomendaciones

- Se recomienda comenzar con un enfoque simple al aplicar el método Kanban en el desarrollo de software. Se pueden utilizar herramientas como pizarrones o tarjetas para visualizar las tareas y el progreso.
- Utilizar el software de aprendizaje de vocabulario básico de inglés como una herramienta de ayuda y fortalecimiento de conocimientos en las demás instituciones educativas en las que se imparten el idioma inglés.
- Realizar actualizaciones del software e incorporar nuevas funcionalidades que ayuden y mejoren el proceso de aprendizaje del vocabulario básico de inglés.

BIBLIOGRAFÍA

- Alcaraz, M. (2020). *getbillage*. Obtenido de https://www.getbillage.com/es/blog/tablero-gestion-kanban#question_3
- Alvarez, M., Marcelino, M., Macías, A., & Novoa, J. C. (2021). METODOLOGÍA TRADICIONAL VS ÁGIL PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE. Obtenido de <https://www.boletin.upiita.ipn.mx/index.php/ciencia/925-cyt-numero-83/1901-metodologia-tradicional-vs-agil-para-la-gestion-de-proyectos-de-software#:~:text=Las%20metodologías%20tradicionales%20se%20caracterizan,actividades%20y%20artefactos%20claramente%20>
- Calero, C., Moraga, M. Á., & Piattini, M. G. (2010). *Calidad del productop y proceso software*. (C. Calero, M. Á. Moraga, & M. G. Piattini, Edits.) España: RA-MA.
- Calvo, D. (2018). *Calvo, Diego*. Obtenido de <https://www.diegocalvo.es/metodologias-tradicionales-y-metodologias-agiles/>
- Coloma, D. (2020). *Itmove Business Agility Experts*. Obtenido de <https://itnove.com/blog/kanban/manager/cual-es-el-origen-del-metodo-kanban/>
- Costa, R. (2021). *adaptworks Emphasys Group*. Obtenido de <https://blog.adapt.works/es/6-practicas-kanban/>
- Demera, R. (2021). *tribalyte*. Obtenido de https://tech.tribalyte.eu/blog-metodologias-tradicional-vs-agil#Metodologias_tradicionales
- Donetonic. (2022). *donetonic*. Obtenido de https://donetonic.com/es/que-es-un-tablero-kanban/#Elementos_de_un_tablero_Kanban
- Gabaldón, J. (2021). *Linkedin*. Obtenido de <https://es.linkedin.com/pulse/metodologías-tradicionales-vs-ágiles-josé-gabaldón>
- Garcés, L. R., & Egas, L. M. (2013). Evolución de las Metodologías de desarrollo de la Ingeniería de software en el proceso la Ingeniería de Sistemas Software. *Revista Científica Y Tecnológica UPSE, 1*(3). doi:<https://doi.org/10.26423/rctu.v1i3.29>
- Ginzo Technologies. (2021). *Ginzo Technologies*. Obtenido de <https://ginzo.tech/blog/metodologia-clasica-desarrollo-software/>
- Hernández, A. (2016). *ingenieriaabril*. Obtenido de <http://ingenieriaabril.blogspot.com/2016/09/metodologias-para-el-desarrollo-de.html>

- INESDI. (2021). *https://www.inesdi.com/blog/que-son-las-metodologias-agiles-tipos-y-ejemplos/*. Obtenido de INESDI BUSINESS TECHSCHOOL: *https://www.inesdi.com/blog/que-son-las-metodologias-agiles-tipos-y-ejemplos/*
- Kanban University. (2021). *LA GUÍA OFICIAL DEL MÉTODO KANBAN (Vol. I)*. Obtenido de *https://kanban.university/wp-content/uploads/2021/11/The-Official-Kanban-Guide_Spanish_A4.pdf*
- Kanbanize. (2021). *Kanbanize*. Obtenido de *https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban*
- kanbantool. (2020). *kanban tool*. Obtenido de *https://kanbantool.com/es/guia-kanban/historia-de-kanban*
- Letelier, P., & Penadés, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *TÉCNICA ADMINISTRATIVA*, V(26). Obtenido de *http://www.cyta.com.ar/ta0502/b_v5n2a1.htm*
- López Mendoza, M. (2020). *OpenWebinars*. Obtenido de *https://openwebinars.net/blog/extreme-programming-que-es-y-como-aplicarlo/*
- López Tibanta, C. E. (2021). *DESARROLLO DE UN SISTEMA WEB PARA LA AUDITORÍA DE*. Machala.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Tesis de Licenciatura en Sistemas y Computación, Universidad Católica Argentina, Facultad de Química e Ingeniería “Fray Rogelio Bacon”. Obtenido de *https://repositorio.uca.edu.ar/handle/123456789/522*
- Martins, J. (2022). *asana*. Obtenido de *https://asana.com/es/resources/what-is-kanban*
- Medewar, S. (2022). *Geekflare*. Obtenido de *https://geekflare.com/es/best-kanban-board-tools/*
- Melanie. (2022). *Blog - ComparaSoftware*. Obtenido de *https://blog.comparasoftware.com/kanban-de-produccion/#Kanban-en-el-sector-de-produccion*
- Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*, 11(2), 30-39. Obtenido de *https://www.redalyc.org/pdf/4962/496250736004.pdf*
- Ortega Moreno, M. (2018). *SISTEMA DE EVALUACIÓN DE LA CALIDAD DE LOS COMPONENTES WEB CENTRADO EN LOS USUARIOS FINALES*. Madrid.

- Pinto de los Ríos, J. S. (2015). *Implementación del método Kanban en las empresas constructoras pequeñas y medianas en la ejecución de un proyecto en Colombia*. UNIVERSITAT POLITÈCNICA DE VALÈNCIA, E.T.S.I. CAMINOS, CANALES Y PUERTOS. Obtenido de https://riunet.upv.es/bitstream/handle/10251/51733/MEMORIA_TFM%20Pinto%20de%20los%20Rios%20Juan%20Sebastian.pdf?sequence=1
- Ramírez, S. L. (2018). *studocu*. Obtenido de <https://www.studocu.com/pe/document/universidad-nacional-del-santa/sistemas-de-informacion/iso-iec-25000-norma-de-calidad-en-software/2033616>
- Red Hat. (2022). *Red Hat*. Obtenido de <https://www.redhat.com/es/devops/what-is-agile-methodology#:~:text=La%20metodología%20ágil%20como%20la,el%20desarrollo%20ágil%20de%20software>.
- Roa, P., Morales, C., & Gutiérrez, P. (2015). Norma ISO/IEC 25000. *Tecnología Investigación y Academia*, 3(2).
- Rocha Garrido, G. (2021). *profile*. Obtenido de <https://profile.es/blog/como-hacer-tablero-kanban-ejemplo/>
- Sáez, J. (2021). *IEBS*. Obtenido de <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
- Salazar López, B. (2019). *Ingeniería Industrial Online*. Obtenido de <https://www.ingenieriaindustrialonline.com/lean-manufacturing/kanban-control-de-materiales-y-produccion/>
- Salesdorado. (2022). *Salesdorado*. Obtenido de <https://salesdorado.com/es/automatizacion/logiciels-kanban/>
- SALESFORCE LATINOAMÉRICA. (2021). *salesforce*. Obtenido de <https://www.salesforce.com/mx/blog/2021/12/que-son-metodologias-agiles-y-como-pueden-ayudar-a-tus-equipos-de-trabajo.html>
- Santander Universidades. (2020). *Santander | Becas*. Obtenido de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Sentrio. (2021). *Sentrio*. Obtenido de https://sentrio.io/blog/valores-principios-agile-manifiesto-agil/#Los_4_valores_del_Manifiesto_Agile
- Solera, S. (2022). *OCCAM*. Obtenido de <https://www.occamagenciadigital.com/blog/las-mejores-metodologias-para-un-correcto-desarrollo-de-software>
- SOLVING AD HOC. (2018). *solvingadhoc*. Obtenido de <https://solvingadhoc.com/herramientas-kanban-una-gestion-tareas-eficaz/>

- Stsepanets, A. (2022). *GANTTPRO*. Obtenido de <https://blog.ganttpro.com/es/metodo-kanban-para-mejorar-el-flujo-de-trabajo/>
- Tinoco Gómez, O., Rosales López, P. P., & Salas Bacalla, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial Data*, 13(2), 70-74. Obtenido de <http://www.redalyc.org/articulo.oa?id=81619984009>
- Toledo, R. (2022). *Grupo cibernos*. Obtenido de <https://www.grupocibernos.com/blog/conoce-las-metodologias-de-desarrollo-agil-mas-usadas>
- Vergara Pineda, J. M. (2017). *coriaweb*. Obtenido de <https://www.coriaweb.hosting/metodo-kanban-cuales-principios-basicos/>

ANEXOS

- **Anexo 1:** Encuesta

Encuesta_Evaluación de Usabilidad - Facilidad de aprendizaje

[Iniciar sesión en Google](#) para guardar lo que llevas hecho. [Más información](#)

***Obligatorio**

¿Cómo calificaría la facilidad de uso del programa? *

Muy difícil

Difícil

Regular

Fácil

Muy fácil

¿Encuentra que el programa es intuitivo y fácil de entender?

Pésimo

Malo

Regular

Bueno

Excelente

¿Ha experimentado algún problema o error mientras utiliza el programa?

- Siempre
- Casi siempre
- A veces
- Casi nunca
- Nunca

¿Tuviste que pedir ayuda a alguien más para aprender a utilizar el programa?

- Siempre
- Casi siempre
- A veces
- Casi nunca
- Nunca

¿Cuánto tiempo te llevó realizar una tarea del programa?

- Más de 8 minutos
- De 5 a 7 minutos
- De 2 a 4 minutos
- Menos de 1 minutos