



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE COMPUTACIÓN

**CONSTRUCCIÓN DE UNA APLICACIÓN MÓVIL PARA EVALUAR EL
DESEMPEÑO DOCENTE DE LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTOR: BRYAN DAVID MORALES TOAPANTA

TUTOR: HOLGER RAÚL ORTEGA MARTÍNEZ

Quito – Ecuador

2024

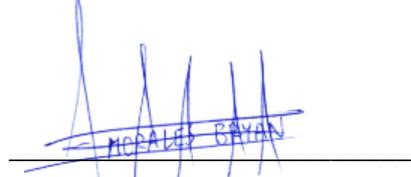
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Bryan David Morales Toapanta con documento de identificación N°1725084113;
manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la
Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total
parcial el presente trabajo de titulación.

Quito, 05 de febrero de 2024

Atentamente,



Bryan David Morales Toapanta

1725084113

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Bryan David Morales Toapanta con documento de identificación N° 1725084113, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: “Construcción de una aplicación móvil para evaluar el desempeño docente de la Universidad Politécnica Salesiana.”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 05 de febrero de 2024

Atentamente,



Bryan David Morales Toapanta

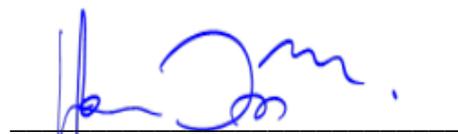
1725084113

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Holger Raúl Ortega Martínez con documento de identificación N° 1708182728, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: CONSTRUCCIÓN DE UNA APLICACIÓN MÓVIL PARA EVALUAR EL DESEMPEÑO DOCENTE DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, realizado por Bryan David Morales Toapanta con documento de identificación N° 1725084113, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 05 de febrero de 2024

Atentamente,



Físico. Holger Raúl Ortega Martínez, MSc

DEDICATORIA

Repleto de felicidad por este gran logro realizado, quiero agradecer en primera instancia a Dios y a mis padres, Morales Guachamin Manuel y Toapanta Tarco María Cleotilde, quienes con mucho amor y paciencia supieron llevarme por el camino correcto, brindándome su apoyo incondicional para culminar mi carrera profesional.

Sin olvidarme de mis hermanos y familiares, hermano Morales Jonathan, hermana Morales Daniela y madrina Morales Sonia, ellos también fueron parte de mi camino, les agradezco desde el fondo de mi corazón por ese cariño y ayuda que supieron brindarme.

ÍNDICE DE CONTENIDOS

CAPÍTULO I.....	1
ANTECEDENTES Y GENERALIDADES.....	1
1.1 Introducción	1
1.2 PROBLEMÁTICA.....	2
1.3 JUSTIFICACIÓN.....	3
1.4 OBJETIVOS GENERAL Y ESPECÍFICOS.....	4
1.4.1 Objetivo general	5
1.4.2 Objetivos específicos.....	5
CAPÍTULO II.....	6
MARCO TEÓRICO.....	6
1.5 EVALUACIÓN DEL DESEMPEÑO DOCENTE.....	6
1.6 DISPOSITIVOS MÓVILES	7
1.7 APLICACIONES MÓVILES	8
1.7.1 Aplicaciones nativas.....	9
1.7.2 Aplicaciones híbridas	10
1.7.3 Aplicaciones Web.....	10
1.7.4 Progressive Web Application (PWA)	11
1.8 SISTEMAS OPERATIVOS MÓVILES	11
1.9 LENGUAJES DE PROGRAMACIÓN PARA APLICACIONES MÓVILES	12
1.9.1 Kotlin.....	13

1.9.2	Swift	13
1.9.3	JavaScript	14
1.9.4	Dart.....	15
1.10	FRAMEWORK DE FLUTTER PARA DART	17
1.11	BASES DE DATOS	20
1.12	BASES DE DATOS NO RELACIONALES	21
1.12.1	Apache Cassandra	22
1.12.2	MongoDB.....	22
1.12.3	Firebase	23
1.13	BASES DE DATOS RELACIONALES	24
1.13.1	MariaDB.....	24
1.13.2	PostgreSQL	25
1.13.3	SupaBase	25
1.14	ARQUITECTURA WEB	26
1.14.1	Arquitectura monolítica.....	27
1.14.2	Arquitectura cliente servidor.....	28
1.14.3	Arquitectura física de 3 niveles.....	28
1.14.4	Arquitectura física multinivel.....	29
1.14.5	Arquitectura de microservicios	30
1.15	METODOLOGÍAS TRADICIONALES PARA EL DESARROLLO DE SOFTWARE.....	31
1.16	METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE	32

1.16.1	Metodología ágil XP	33
1.16.2	Metodología ágil Kanban	34
1.16.3	Metodología ágil SCRUM	36
CAPÍTULO III		38
METODOLOGÍA		38
1.17	Recopilación documental	38
1.17.1	Metodología SCRUM	39
1.18	DIAGRAMA DE BLOQUES	40
1.19	SPRINT 0	40
1.19.1	Especificación de roles	41
1.19.2	Declaración del producto	41
1.19.3	Creación del área de trabajo	42
1.19.4	Product backlog o pila de producto	42
1.19.5	Refinamiento del product backlog	44
1.19.6	Planificación de los sprints	45
1.19.7	Definición de finalizado	48
1.19.8	Diagrama de casos de uso general	49
1.19.9	Diagrama de casos de uso para el ingreso al sistema	50
1.19.10	Diagrama de casos de uso para la parametrización del sistema	51
1.19.11	Diagrama de casos de uso para el ingreso de evaluaciones	52
1.19.12	Diagrama de casos de uso para la consulta de evaluaciones	54
1.19.13	Diagrama de casos de uso para el reporte de evaluación	55

1.20	IMPLEMENTACIÓN	56
1.20.1	Desarrollo font-end	56
1.20.2	Sprint 1	63
1.20.3	Sprint 2	67
1.20.4	Sprint 3	80
CAPÍTULO IV		82
RESULTADOS		82
1.21	PRUEBAS DE FUNCIONAMIENTO	82
1.21.1	Pruebas para la creación de usuarios	83
1.21.2	Pruebas del Login.....	84
1.21.3	Pruebas de ingreso de información en los módulos de la evaluación	85
1.21.4	Pruebas de revisión de la información de las evaluaciones del lado del docente	86
1.22	PRUEBAS DE USABILIDAD.	87
CONCLUSIONES.....		93
RECOMENDACIONES.....		94
REFERENCIAS		95

ÍNDICE DE TABLAS

Tabla 1 Principales sistemas operativos móviles.....	12
Tabla 2 Principales características de DART	16
Tabla 3 Principales características de Flutter	19
Tabla 4 Principales tipos de bases de datos	21
Tabla 5 Comparación de metodología tradicional vs ágil	33
Tabla 6 Fases de la metodología Kanban para el desarrollo de software	36
Tabla 8 Definición de roles del proyecto.....	41
Tabla 9 Definición de herramientas para el desarrollo.....	42
Tabla 10 Pila de productos	42
Tabla 11 Refinamiento del producto backlog.....	44
Tabla 12 Ítems del producto backlog para el primer sprint.....	46
Tabla 13 Ítems del producto backlog para el segundo sprint	47
Tabla 14 Ítems del producto backlog para el tercer sprint.....	47
Tabla 15 Actividades identificadas para ejecutar el sprint 1	63
Tabla 16 Actividades identificadas para ejecutar el sprint 2	68
Tabla 17 Resultado de la encuesta del buen docente.....	69
Tabla 18 Actividades identificadas para ejecutar el sprint 3	80
Tabla 19 Resultados de las pruebas de funcionamiento para la creación de usuarios.....	83
Tabla 20 Resultados de las pruebas de funcionamiento para el login	84
Tabla 21 Resultados de las pruebas de ingreso de información en los módulos de la evaluación	85
Tabla 22 Resultados de las pruebas de visualización de los docentes de las evaluaciones recibidas	86

ÍNDICE DE FIGURAS

Figura 2 <i>Evaluación docente</i>	6
Figura 3 <i>Evolución de los teléfonos celulares</i>	7
Figura 4 <i>Aplicaciones para dispositivos móviles</i>	8
Figura 5 <i>Las 10 app más descargadas en el 2022</i>	9
Figura 6 <i>Logo de Kotlin</i>	13
Figura 7 <i>Logo de Swift</i>	14
Figura 8 <i>Logo de JavaScript</i>	15
Figura 9 <i>Logo de Dart</i>	15
Figura 10 <i>Características de Dart</i>	17
Figura 11 <i>Logo de Flutter</i>	18
Figura 12 <i>Funcionamiento y Estructura de Flutter</i>	20
Figura 13 <i>Logo de Apache Cassandra</i>	22
Figura 14 <i>Logo de MongoDB</i>	23
Figura 15 <i>Logo de Firebase</i>	23
Figura 16 <i>Logo de MariaDB</i>	24
Figura 17 <i>Logo de PostgreSQL</i>	25
Figura 18 <i>Logo de Supabase</i>	26
Figura 19 <i>Modelo de arquitectura monolítica</i>	27
Figura 20 <i>Modelo de arquitectura cliente servidor</i>	28
Figura 21 <i>Modelo de arquitectura física de 3 niveles</i>	29
Figura 22 <i>Modelo de arquitectura de microservicios</i>	30
Figura 23 <i>Fases del modelo en cascada</i>	32
Figura 24 <i>Fases de la metodología XP</i>	34

Figura 25 <i>Tablero Kanban</i>	35
Figura 26 <i>Fases para la ejecución de los sprints</i>	37
Figura 1 <i>Fases de la metodología de trabajo</i>	38
Figura 27 <i>Fases de evolución de la aplicación</i>	40
Figura 28 <i>Listado para definir la finalización de un sprint</i>	48
Figura 29 <i>Diagrama de casos de uso general</i>	49
Figura 30 <i>Diagrama de casos de uso de ingreso al sistema</i>	50
Figura 31 <i>Diagrama de casos de uso para la parametrización del sistema</i>	51
Figura 32 <i>Diagrama de casos de uso para el ingreso de evaluaciones</i>	52
Figura 33 <i>Diagrama de casos de uso para la consulta de evaluaciones</i>	54
Figura 34 <i>Diagrama de casos de uso para el reporte de evaluación</i>	55
Figura 35 <i>Líneas de fragmento de código instalación librerías</i>	57
Figura 36 <i>Organización lógica MVC proyecto</i>	58
Figura 37 <i>Organización lógica MVC proyecto</i>	59
Figura 38 <i>Fragmentos de líneas de código calificación con estrella</i>	60
Figura 39 <i>Fragmentos de línea de código para el desarrollo del back-end</i>	61
Figura 40 <i>Diagrama de casos de uso para el reporte de evaluación</i>	62
Figura 35 <i>Descripción de los sprints</i>	62
Figura 36 <i>BurnDown Chart del sprint 1</i>	63
Figura 37 <i>Modelo entidad relación</i>	64
Figura 38 <i>Ingreso al sistema de información del lado del alumno</i>	66
Figura 39 <i>Ingreso al sistema de información del lado del docente</i>	67
Figura 40 <i>BurnDown Chart del sprint 2</i>	68
Figura 41 <i>Listado de materias a evaluar</i>	70
Figura 42 <i>Listado de categorías que forman parte de la evaluación</i>	71

Figura 43 <i>Categoría de estrellas</i>	72
Figura 44 <i>Categoría Ranking Don Bosco</i>	73
Figura 45 <i>Categoría Insignias</i>	74
Figura 46 <i>Categoría Reseña</i>	75
Figura 47 <i>Listado de categorías para revisión de la evaluación</i>	76
Figura 48 <i>Categoría de estrellas</i>	77
Figura 49 <i>Categoría insignias</i>	77
Figura 50 <i>Categoría Ranking Don Bosco</i>	78
Figura 51 <i>Categoría Reseña</i>	79
Figura 52 <i>BurnDown Chart del sprint 3</i>	80
Figura 53 <i>Post para motivar a la descarga de la App</i>	81
Figura 54 <i>Descarga de la App de Play Store</i>	82
Figura 55 <i>Resultados de la pregunta relacionada las interfaces del aplicativo móvil</i>	87
Figura 56 <i>Resultados de la pregunta relacionada con la frecuencia de uso de la aplicación móvil</i>	88
Figura 57 <i>Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil</i>	88
Figura 58 <i>Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil</i>	89
Figura 59 <i>Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil</i>	90
Figura 60 <i>Resultados del uso de la pantalla de perfil de usuario</i>	91
Figura 61 <i>Resultados del uso de la pantalla de los componentes de la evaluación</i>	91
Figura 62 <i>Resultados del uso de la pantalla para revisar los resultados de la calificación por estrellas</i>	92

RESUMEN

El presente estudio de investigación tiene como objetivo construir una aplicación móvil dirigida a la evaluación del desempeño docente en la Universidad Politécnica Salesiana que le permita a los alumnos evaluar a sus docentes de manera voluntaria una vez que la clase ha terminado, generando información de utilidad para la mejora continua de los procesos de enseñanza-aprendizaje en el aula.

El cumplimiento de los objetivos establecidos demandó que la presente investigación se la realice en dos fases, en una primera fase la revisión bibliográfica que permita contar con los insumos teóricos necesarios y mediante la metodología SCRUM realizar el desarrollo en sí de la aplicación móvil. El resultado principal obtenido está relacionado con la propuesta de aplicativo móvil que les permita a los docentes llevar a cabo sus procesos de evaluación.

ABSTRACT

The objective of this research study is to build a mobile application aimed at evaluating teaching performance at the Salesian Polytechnic University that allows students to voluntarily evaluate their teachers once the class has finished, generating useful information for continuous improvement of teaching-learning processes in the classroom.

The fulfillment of the established objectives demanded that the present investigation be carried out in two phases, a first bibliographic review that allows having the necessary theoretical inputs and through the SCRUM methodology to carry out the development of the mobile application itself. The main result obtained is related to the mobile application proposal that allows the University to carry out its teacher evaluation processes.

CAPÍTULO I

ANTECEDENTES Y GENERALIDADES

1.1 Introducción

El origen de la evaluación de las acciones que realizamos es tan antiguo como la misma sociedad, cuyos pilares se asientan en la curiosidad del ser humano y en la necesidad de superación (haciendo mejor las cosas) tratando de responder a la pregunta ¿cuán bien hemos hecho una actividad?, de la misma se deriva la necesidad de identificar los criterios que permitan cuantificar el desempeño realizando una actividad.

Para el caso de la evaluación docente, sus primeros indicios se refieren a la década de los 70 en los Estados Unidos, quienes lo utilizaron como un mecanismo que permita la profesionalización docente y de esta manera influir en el incremento de los índices de calidad educativa. En el Ecuador la normativa para evaluar a los docentes de centros de educación superior se incorporó recién para el año 2000 a través del artículo 53 de la LOES (Ley Orgánica de Educación Superior) que entró en vigencia en mayo del mencionado año en la que se dispuso la obligatoriedad de evaluar a los docentes una vez al año.

Para el caso de la Universidad Politécnica Salesiana, para la aplicación de esta normativa de evaluación cuenta con un sistema informático mismo que permite realizar las evaluaciones al finalizar cada período académico, esta periodicidad impide que los docentes cuenten con retroalimentación frecuente de la percepción de su accionar en las aulas por parte de sus alumnos. Adicionalmente, acorde a los resultados del primer período académico de 2020 en la evaluación a los docentes 7 de cada 10 alumnos de la universidad participaron dando su opinión de manera confidencial, existiendo un porcentaje importante de ausencia principalmente porque consideran que es un proceso innecesario por la periodicidad en la que se la realiza (al final del semestre) (Cárdenas, 2020).

Lo antes expuesto abona en la necesidad de mejorar los procedimientos de evaluación, permitiendo que los mismos sean frecuentes y no solo a final de semestre donde confluyen otros factores que no permitan que la evaluación sea objetiva (pérdida de la materia, acumulación de tareas y exámenes, etc.); por lo que se requiere contar con una herramienta tecnológica capaz de adaptarse a los nuevos cambios asociados a la era de la información.

1.2 PROBLEMÁTICA

A nivel mundial, según la UNESCO en todo el mundo existe más de 69 millones de docentes que carecen de cualificaciones y formación básica para adaptarse al ritmo de los cambios educativos (Souza, 2022) evidenciándose una carencia tanto en su formación académica como en la calidad de su enseñanza.

A pesar de la importancia que tiene la evaluación docente en todos los niveles educativos no está claramente marcado el accionar de las universidades ante resultados negativos, acorde a lo afirmado por Itlizaliturii "evaluar a los docentes es una manera de elevar la calidad de la enseñanza, pero las consecuencias laborales que tienen los resultados obtenidos por los docentes dividen a los gobiernos" (Montoya & Espinoza, 2017) .

En los países menos desarrollados, se genera escasez de auto crítica en el personal docente, lo que genera la necesidad imprescindible que periódicamente se implementen procesos de evaluación del desempeño. En referencia a este tema, Pearlman sostiene que, "La evaluación del educador no debe verse como una estrategia de inspección jerárquica que controla las actividades de los profesores, sino como una forma de fomentar y favorecer el perfeccionamiento del profesorado" (Montoya & Espinoza, 2017).

En Ecuador, según el Instituto de Evaluación de la Educación (INEVAL) y el Ministerio de Educación (MINEDUC), para el año 2016, 139.474 docentes fueron evaluados en conocimiento, destrezas y habilidades para cada uno de los perfiles profesionales establecidos,

la evaluación demuestra que más del 70% de los docentes obtuvo menos de 700 puntos, es decir insuficiencia. (Torres, s. f.).

Entre las diversas causas que pueden estar originando este problema de estudio, se ha identificado que los resultados de la evaluación llegan muy tarde al docente (posterior a finalizar el año lectivo); y la otra en la que, la poca credibilidad de los estudiantes reduce el interés de participar en los procesos de evaluación a los docentes. Para el caso de los efectos que puede traer este problema, principalmente se relacionan con que el docente no puede realizar una corrección oportuna, privándole de la posibilidad de usar esa información como retroalimentación, rápido y eficaz; y, por último, la poca participación de los estudiantes, lo cual hace que la recopilación de información se prolongue por más tiempo del planificado y no se obtengan datos verídicos de dicha evaluación.

Lo antes expuesto abona en la necesidad de contar con un proceso continuo de evaluación continua durante todo el semestre, el cual se involucran a los alumnos con el objetivo de que los docentes reciban una retroalimentación continua que les permita inmediatamente realizar correctivos mejorando sus procesos de enseñanza aprendizaje; De este requerimiento se desprende contar con una aplicación móvil amigable para los estudiantes a la cual puedan acceder inmediatamente terminada la clase y evaluar la gestión del docente; en este sentido, los resultados finales de la evaluación serán el resultado del cúmulo de evaluaciones realizadas durante cada una de sus clases.

Finalmente, es importante indicar que la aplicación móvil objeto del presente proyecto de investigación aportará con la agilidad necesaria para dinamizar el proceso; además, tendrá la ventaja de enfocarse en las buenas prácticas y en los aspectos emocionales, mismos que guardan mayor relación con los resultados educativos.

1.3 JUSTIFICACIÓN

Desde la década de los 40's, el desarrollo de sistemas ha ido evolucionado hasta convertirse en una profesión con distintas especialidades, cuyo objetivo principal es la implementación de soluciones tecnológicas que permitan dar respuesta de manera eficiente a las necesidades institucionales (Garrcés & Egas, 2018), lo que convierte a los datos como la principal fuente que permite tomar decisiones (Soares, 2010).

En este sentido, la importancia de este proyecto radica en brindar una solución tecnológica capaz de incorporar preguntas con enfoque emocional que permitan evaluar a los docentes permitiendo involucrar a más alumnos al proceso evaluativo durante todo el semestre, mejorando adicionalmente las falencias detectadas en el proceso de evaluación que aplica la universidad.

La incorporación de un aplicativo móvil al proceso de evaluación le permitirá a la universidad llegar de mejor manera a los estudiantes que en la actualidad están acostumbrados a realizar todo a través de su teléfono celular, identificándose a los estudiantes como los primeros beneficiarios del desarrollo propuesto, los siguientes beneficiados serían los docentes quienes de manera frecuente podrán consultar sus evaluaciones para corregir en los ámbitos que reciban baja calificación; finalmente, la universidad en su conjunto se verá beneficiada al contar con información periódica que le permita tomar correctivos inmediatos.

En cuanto a la factibilidad técnica, se debe considerar que en el presente desarrollo se aplicarán los conocimientos adquiridos durante toda la trayectoria universitaria relacionadas con el ciclo de vida de los sistemas de información. Sobre la factibilidad económica se debe considerar que para el desarrollo del aplicativo móvil no se requiere del pago de licencias de lenguajes de programación, bases de datos o de infraestructura tecnológica.

1.4 OBJETIVOS GENERAL Y ESPECÍFICOS

1.4.1 Objetivo general

Construir una aplicación móvil dirigida a la evaluación del desempeño docente en la Universidad Politécnica Salesiana.

1.4.2 Objetivos específicos

- Determinar una metodología de desarrollo de software para elaborar la app móvil.
- Formular un número de preguntas centralizadas en las emociones.
- Diseñar la app móvil, que cuente con un entorno amigable para evaluar al docente.
- Construir la app móvil empleando herramientas de software libre.
- Evaluar la aplicación móvil para garantizar su funcionamiento.

CAPÍTULO II

MARCO TEÓRICO

1.5 EVALUACIÓN DEL DESEMPEÑO DOCENTE

Este tipo de evaluación se origina en la necesidad de asegurar la calidad educativa y garantizar que los profesores cumplan con los estándares requeridos en la enseñanza. La calidad educativa está directamente relacionada con la calidad de la enseñanza, la cual se logra a través de la dedicación, los métodos de enseñanza y los mecanismos utilizados por los docentes para transmitir sus conocimientos a los alumnos en las diferentes materias que imparten (Morales , 2001).

Los primeros mecanismos relacionados con la evaluación del desempeño docente data de los años 50 (Urriola, 2013) a través de la implementación de mecanismos que permitían brindar estímulos económicos a los docentes que obtienen resultados positivos y en contraposición para aquellos cuya evaluación no era satisfactoria eran excluidos de tales beneficios e incluso siendo sancionados con el pago de penalidades económicas (ver Figura 1). Siendo éste como un modelo punitivo que no generaba sinergias de colaboración entre los docentes sino más bien una competitividad individualista por obtener estímulos económicos (Moreno, 2018).

Figura 1

Evaluación docente



Nota. Se describe la manera en que se realizan la evaluación a los docentes por parte de sus alumnos. Fuente: (Gestión Educativa, 2016).

En Ecuador, la Ley Orgánica de Educación Superior experimentó su última reforma en el año 2010, en dicha reforma, el artículo 151 establece que los profesores deben someterse a evaluaciones periódicas, cuyos mecanismos serán definidos por cada Institución de Educación Superior (IES). Estas evaluaciones deben incluir como uno de los parámetros la retroalimentación que los alumnos proporcionen sobre sus docentes.

1.6 DISPOSITIVOS MÓVILES

El 11 de mayo de 1994 el Ecuador le dio la bienvenida al servicio de telefonía móvil (Revista Líderes, 2014) permitiendo de esta manera la llegada de los teléfonos celulares cuyo principal objetivo fue el de mejorar la comunicación entre las personas. La popularización de estos dispositivos hizo que su uso se masificó progresivamente incrementando la necesidad de satisfacer mayores exigencias generando la aparición de los teléfonos inteligentes o smartphone, considerándose (hasta el momento) como la mayor evolución de los teléfonos móviles, la Figura 2 describe la evolución de los teléfonos celulares desde su primera versión.

Figura 2

Evolución de los teléfonos celulares



Nota. Se describe la evolución de los teléfonos celulares desde su aparición. Fuente: (Ros, 2015).

Para Malave y Beauperthuy (2011) un teléfono inteligente es “un dispositivo electrónico que permite a su usuario llevar consigo un objeto de dimensiones pequeñas, que cumpla funciones similares a las de un computador personal” cuyo desarrollo ha permitido contar con un sin número de aplicaciones que permiten el acceso a redes sociales, realizar transacciones bancarias, enviar mensajes de texto, navegar en internet, pedir un taxi, etc.

1.7 APLICACIONES MÓVILES

La popularidad ganada por los dispositivos móviles se centra en el desarrollo de las aplicaciones existentes para optimizar funcionamiento, mismas que logran que los usuarios cada vez estén más conectados realizando actividades diarias desde un teléfono celular. El origen de las App se remonta a la aparición del sistema operativo BlackBerry que permitió a los usuarios decidir las aplicaciones que desea tener instaladas en su teléfono; sin embargo, los teléfonos de esa época tenían pantallas pequeñas y no táctiles lo que generaba que las aplicaciones sean limitadas en su funcionalidad.

Figura 3

Aplicaciones para dispositivos móviles



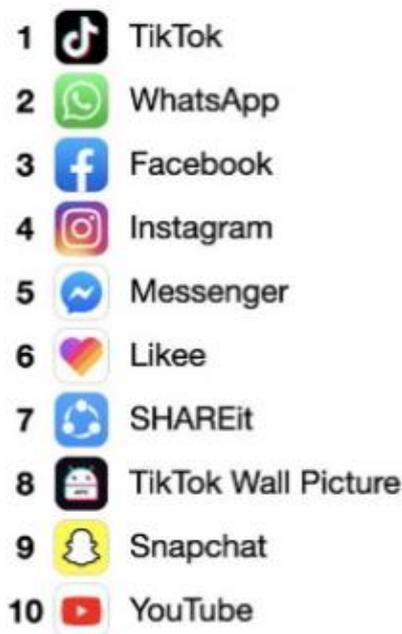
Nota. Se describe el sin número de aplicaciones disponibles para los dispositivos móviles.

Fuente: (Payares, 2022).

En la actualidad los teléfonos cuentan con pantallas táctiles que en promedio calzan en la palma de la mano, sus sistemas operativos permiten la instalación de App que se integran con las características del equipo ampliando sus funciones, la Figura 3 describe las múltiples acciones que los usuarios pueden realizar como navegar por páginas web, realizar video llamadas, compartir su ubicación en tiempo real, acceder a aplicaciones con reconocimiento facial, etc. Así también, la Figura 4 muestra los resultados del estudio “Q1 2022: Store Intelligence Data Digest” realizado por la empresa Sensor Tower, especialista en estudios de mercado digitales, en la que concluye que TikTok fue la App con más de 3.500 millones de descargas lo que la convierte en la número uno dejando por detrás a las de propiedad de la empresa Meta (Payares, 2022).

Figura 4

Las 10 app más descargadas en el 2022



Nota. Se enlistan las 10 App más descargadas en el año 2022. Fuente: Payares (2022).

1.7.1 Aplicaciones nativas

Este tipo de aplicaciones se desarrollan específicamente para funcionar en una sola de las plataformas móviles existentes como iOS o Android, su desarrollo se realiza a través de

lenguajes de programación nativos proporcionados por el fabricante de la plataforma, como Objective-C o Swift para iOS y Java o Kotlin para Android. Se destacan por tener acceso completo a todas las características y funcionalidades del teléfono inteligente como la cámara, los sensores, el GPS y las notificaciones push; adicionalmente, se integran optimamente con el sistema operativo lo que les permite ofrecer un rendimiento optimizado, ya que están diseñadas específicamente para la plataforma objetivo (Rodríguez, Vera, Martínez, & Dogliotti, 2020).

Aunque desarrollar aplicaciones nativas en la mayoría de casos puede requerir más tiempo y esfuerzo que el desarrollo de las híbridas, ofrecen una experiencia de usuario más fluida y personalizada, aprovechando las últimas características y actualizaciones del sistema operativo, brindando a los usuarios una experiencia nativa y optimizada en su plataforma móvil preferida.

1.7.2 Aplicaciones híbridas

Este tipo de aplicaciones se desarrollan utilizando tecnologías web como HTML, CSS y JavaScript, y luego se encapsulan dentro de un contenedor nativo que les permite ejecutarse sobre diferentes plataformas móviles como iOS y Android. Estas aplicaciones aprovechan Frameworks como Apache Cordova, Ionic, React Native, Flutter, entre otros, que les permiten acceder a funcionalidades del dispositivo, como la cámara, los sensores y la geolocalización. Aunque las aplicaciones híbridas pueden ofrecer una mayor portabilidad y un desarrollo más rápido al compartir gran parte del código entre plataformas, pueden presentar algunas limitaciones en rendimiento y acceso a características específicas del sistema operativo en comparación con las aplicaciones nativas (Asociación Iberoamericana de Educación Superior a Distancia, 2017).

1.7.3 Aplicaciones Web

Este tipo de aplicaciones se ejecutan a través de un navegador web sin necesidad su instalación, utilizan tecnologías web como HTML, CSS y JavaScript para crear interfaces interactivas y funcionales, se caracterizan por ser fácilmente accesibles desde diferentes dispositivos y

plataformas, lo que proporciona una mayor flexibilidad y portabilidad. Las aplicaciones web pueden ofrecer una amplia gama de funcionalidades, desde sitios web interactivos hasta aplicaciones más complejas que permiten la gestión de datos, la integración con servicios externos y la colaboración en tiempo real. Además, las actualizaciones y mejoras se implementan centralmente en el servidor, lo que permite a los usuarios acceder siempre a la versión más reciente sin necesidad de actualizar manualmente (Rodríguez, Vera, Martínez, & Dogliotti, 2020).

1.7.4 Progressive Web Application (PWA)

Son aplicaciones web que se caracterizan por su capacidad de combinar lo mejor de las características de las aplicaciones web y de las aplicaciones nativas, ofreciendo de esta manera una mejor experiencia a los usuarios, están diseñadas para funcionar en cualquier dispositivo o plataforma y para su acceso se lo hace mediante cualquier navegador web utilizando tecnologías web como HTML, CSS y JavaScript. Es por esto que las PWA pueden ser instaladas en los dispositivos móviles como una aplicación nativa, lo que les permite funcionar sin conexión a Internet proporcionando una experiencia de usuario similar como si se estuviera utilizando aplicaciones nativas. Las PWA ofrecen beneficios como una instalación sencilla, actualizaciones automáticas y una amplia compatibilidad, convirtiéndolas en una opción para desarrolladores y usuarios que desean una experiencia de aplicación móvil similar en la web.

1.8 SISTEMAS OPERATIVOS MÓVILES

Los sistemas operativos móviles son plataformas diseñadas específicamente para dispositivos móviles, como smartphones y tablets, los más conocidos incluyen iOS de Apple, Android de Google y Windows Phone de Microsoft (actualmente discontinuado). Cada uno de estos sistemas operativos ofrece un conjunto de características y funcionalidades únicas que se describen en la Tabla 1 e incluyen interfaces de usuario distintiva; así también, los sistemas operativos móviles proporcionan una base para ejecutar aplicaciones móviles ofreciendo

servicios esenciales como la gestión de recursos del dispositivo, comunicación inalámbrica, seguridad y acceso a servicios en la nube. Estos sistemas operativos también permiten a los usuarios personalizar sus dispositivos mediante la instalación de aplicaciones, widgets y ajustes de configuración según sus preferencias.

Tabla 1

Principales sistemas operativos móviles

Característica	Android	iOS
Kernel	Linux	OS X
Tipo de Sistema Operativo	Abierto	Cerrado
Lenguaje de programación nativo	Java	Objective C
Multitarea	Si	Si
Estándares soportados	GSM, CDMA	GSM, CDMA
Hardware soportado	Amplia gama de dispositivos móviles	iPhone, iPad, iPod touch
Tienda de Software	Google Play	App Store
Asistente de Voz	S-Voice	Siri

Nota. Se describen las principales características de cada uno de los sistemas operativos móviles. Elaborado por: El autor.

Aunque ambos sistemas operativos comparten características similares, una de las fortalezas distintivas de Android radica en su naturaleza de ser código abierto, esto brinda a los fabricantes de tecnología la flexibilidad de realizar modificaciones y adaptaciones según las necesidades de sus equipos, lo que ha llevado a una amplia adopción de Android en dispositivos móviles a nivel mundial, al ser de código abierto. Android fomenta la colaboración y la innovación lo que ha contribuido en su crecimiento y popularidad en la industria tecnológica.

1.9 LENGUAJES DE PROGRAMACIÓN PARA APLICACIONES MÓVILES

Este tipo de lenguajes desempeñan un papel fundamental en la implementación de aplicaciones móviles, permitiendo a los programadores crear aplicaciones funcionales con una interfaz atractiva para los usuarios, cada lenguaje tiene sus propias características y ventajas, y la

elección del lenguaje depende del sistema operativo objetivo, las necesidades del proyecto y las habilidades del desarrollador. Los lenguajes de programación para aplicaciones móviles proporcionan un sin número de herramientas que facilitan el desarrollo aplicaciones móviles innovadoras y satisfacer las demandas de los usuarios en un mercado en constante evolución.

1.9.1 Kotlin

Es un lenguaje desarrollado por JetBrains y lanzado oficialmente en el año 2016 se caracteriza por ser moderno, seguro y conciso, cuyo principal uso es para el desarrollo de aplicaciones para el sistema operativo Android (ver Figura 5). Este lenguaje se caracteriza por estar diseñado para ser interoperable con Java ofreciendo muchas características avanzadas, como la inferencia de tipos, la programación orientada a objetos y la nulabilidad para evitar errores comunes de NullPointerException (Kotlin, 2022 a).

Figura 5

Logo de Kotlin



Nota. Se muestra el logo del lenguaje de programación para dispositivos móviles Kotlin.

Fuente: (Kotlin, 2022 a).

Kotlin también se destaca por su sintaxis más simple y legible en comparación con Java, facilitando la escritura y lectura del código permitiendo adicionalmente la implementación de aplicaciones de escritorio, para servidores y aplicaciones web, debido a su crecimiento y adopción rápida ha llegado a ser una opción a considerar por los programadores que buscan una alternativa moderna y eficiente a Java (Kotlin, 2022 b).

1.9.2 Swift

Se caracteriza por ser moderno, seguro y potente, es desarrollado por Apple y fue introducido en el mercado tecnológico en el año 2014 cuyo logo se muestra en la Figura 6. Desde su

aparición se convirtió en el lenguaje principal para el desarrollo de aplicaciones del ecosistema de Apple (iOS, macOS, watchOS y tvOS). Su potencialidad radica en la mezcla de las potencialidades de lenguajes como C, Objective-C, Rust y otros, permitiendo de esta manera ofrecer una sintaxis más simple y legible, al tiempo que proporciona un rendimiento rápido y de sólida seguridad (Apple, 2022).

Figura 6

Logo de Swift



Nota. Se muestra el logo del lenguaje de programación para dispositivos móviles Swift. Fuente: (Apple, 2022).

Este lenguaje se enfoca en la seguridad, con inferencia automática de tipos y manejo de opcionales para prevenir errores comunes, también introduce conceptos innovadores como protocolos y extensiones que facilitan la reutilización de código y la modularidad. Adicionalmente, cuenta con una comunidad en crecimiento y el apoyo activo de Apple, Swift continúa evolucionando y siendo adoptado por desarrolladores alrededor del mundo que se encargan de crear potentes aplicaciones que se han posicionado en el mercado de aplicaciones Apple (Apple, 2022).

1.9.3 JavaScript

Se trata de un lenguaje versátil y ampliamente utilizado para la implementación de aplicaciones web, se caracteriza por ser un lenguaje de scripting interpretado por los navegadores web lo que permite agregar interactividad y dinamismo a las páginas web cuyo logo se muestra en la Figura 7. Es conocido por ser orientado a objetos y basado en prototipos que permite: 1) la manipulación y modificación del contenido de una página web, 2) la interacción con el usuario

a través de formularios y eventos, 3) la validación de datos y la creación de animaciones; y, 4) creación de efectos visuales (Zubikarai, 2021)

Figura 7

Logo de JavaScript



Nota. Se muestra el logo del lenguaje de programación JavaScript. Fuente: (Mozilla, 2022).

Además, JavaScript se puede utilizar en entornos de servidor a través de Node.js, lo que permite el desarrollo de aplicaciones web completas del lado del cliente y del servidor, adicionalmente es compatible con una amplia gama de Frameworks y bibliotecas populares, como React, Angular y Vue.js, que facilitan el desarrollo de aplicaciones web complejas. Por su amplia adopción y su comunidad activa, JavaScript sigue siendo uno de los lenguajes más influyentes y relevantes en el desarrollo web actual. (Mozilla, 2022).

1.9.4 Dart

Es un lenguaje desarrollado por Google que se utiliza para la implementación de aplicaciones móviles, aplicaciones web y aplicaciones de escritorio cuyo logo se muestra en la Figura 8. Se caracteriza por ser un lenguaje de alto rendimiento que proporciona características modernas como tipado estático opcional, recolección de basura y soporte para programación asíncrona. Una de las principales ventajas es su capacidad de ser compilado tanto a código nativo como a código JavaScript, lo que permite su ejecución eficiente tanto en dispositivos móviles como en navegadores web (DART, 2022).

Figura 8

Logo de Dart



Dart

Nota. Se muestra el logo del lenguaje de programación para dispositivos móviles DART.

Fuente: (DART, 2022).

Dart ha mostrado un creciente reconocimiento en la comunidad de desarrollo debido a su facilidad de uso, rendimiento y capacidad de construir aplicaciones nativas de alto nivel en múltiples plataformas (DART, 2022). A continuación, en la Tabla 2 se describen sus principales características:

Tabla 2

Principales características de DART

Característica	Detalle
Seguridad	Utiliza la verificación de tipo estático para garantizar que el valor de una variable siempre coincida con el tipo estático de la variable. A veces, esto se conoce como escritura de sonido.
Escritura de código	El sistema de escritura DART también es flexible, lo que permite el uso de un “dynamic tipo” con comprobaciones en tiempo de ejecución, lo que puede ser útil durante la experimentación o para el código que necesita ser especialmente dinámico.
Seguridad nula sólida	Significa que los valores no pueden ser nulos a menos que se configure que pueden serlo. Con una sólida seguridad nula, DART permite proteger a los sistemas de información de excepciones nulas en tiempo de ejecución a través del análisis de código estático.
Bibliotecas	DART tiene un amplio conjunto de bibliotecas principales, que proporcionan elementos esenciales para muchas tareas de programación diarias.

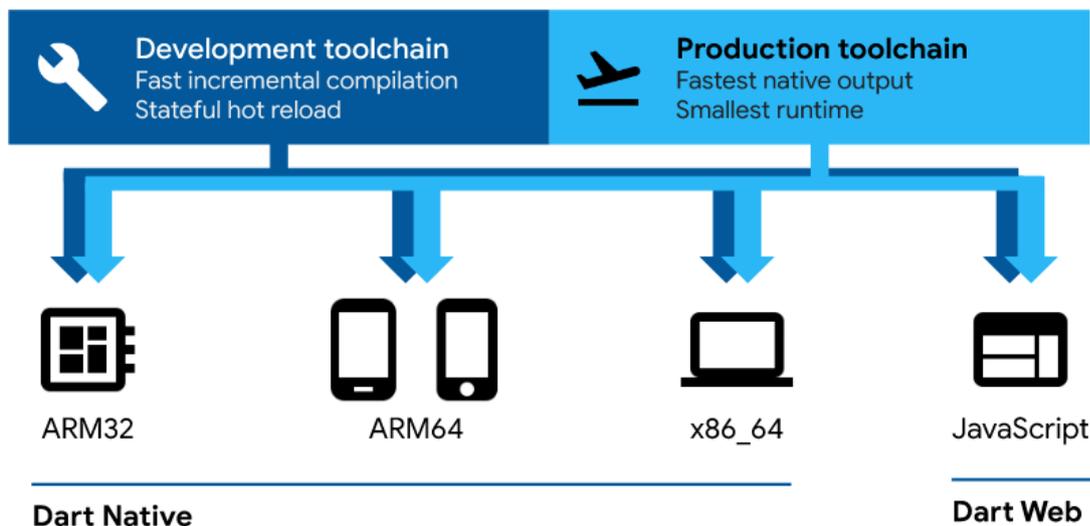
Nota. Se describen las principales características de cada uno de los sistemas operativos móviles. Elaborado por: El autor.

La Figura 9 muestra la manera de compilación de DART lo que le permite ejecutar código de dos maneras diferentes:

- Plataforma nativa: se usa para aplicaciones orientadas a dispositivos móviles y de escritorio; para lo cual, incluye una máquina virtual de Dart con compilación justo a tiempo (JIT) y un compilador adelantado (AOT) para producir código de máquina.
- Plataforma web: se usa para aplicaciones orientadas a la web a través de su compilador web que traduce las sintaxis de Dart a JavaScript.

Figura 9

Características de Dart



Nota. Se describe la tecnología de compilación del lenguaje de programación DART. Fuente: (DART, 2022).

Considerando que Google es el propietario del sistema operativo de mayor popularidad a nivel de dispositivos móviles para el presente proyecto de investigación se ha optado por elegir a DART como el lenguaje en el cual se desarrollará el sistema de evaluación docente propuesto.

1.10 FRAMEWORK DE FLUTTER PARA DART

Es un Framework de código abierto desarrollado por Google para la implementación de aplicaciones móviles, se encuentra diseñado para desarrollar aplicaciones nativas para plataformas iOS y Android utilizando el lenguaje de programación Dart. Se caracteriza principalmente por su enfoque en el diseño de interfaces amigables y atractivas para el usuario, conocidas como "widgets", que permiten una experiencia de usuario consistente en diferentes dispositivos y sistemas operativos. Flutter utiliza un enfoque de "pintura" para renderizar la interfaz de usuario, lo que permite un rendimiento rápido y suave. Además, brinda a los programadores la opción de descargar widgets personalizables para facilitar la creación de interfaces ricas en contenido y animaciones (Flutter España, 2022).

Figura 10

Logo de Flutter



Nota. Se muestra el logo del Framework Flutter para la programación en lenguaje DART.

Fuente: (Flutter España, 2022).

La principal de las características de Flutter es su capacidad de hot-reload, que permite a los programadores ver los resultados de los cambios realizados en el código en tiempo real, evitando el tener que reiniciar la aplicación, brindando agilidad al proceso de programación mejorando la experimentación y facilitando la depuración. Ofrece también acceso directo a las API y características del sistema operativo subyacente, lo que permite un alto nivel de personalización y funcionalidades avanzadas. La Tabla 3 muestra las principales características que en la actualidad posicionan a Flutter como el Framework preferido para el desarrollo de aplicaciones móviles.

Tabla 3*Principales características de Flutter*

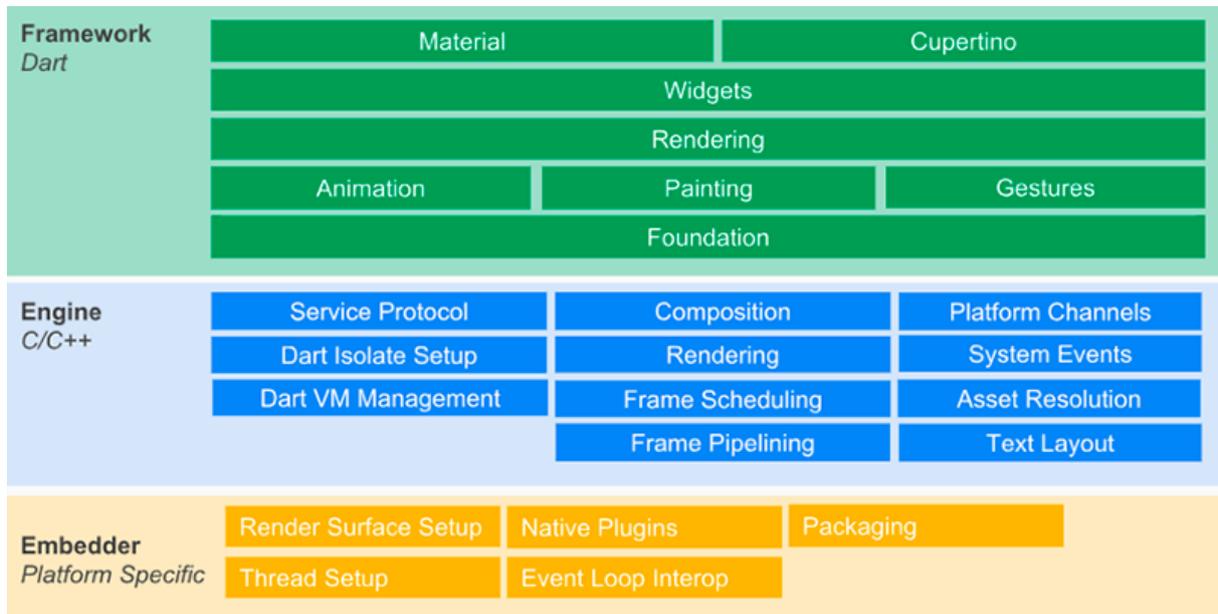
Característica	Detalle
Desarrollo rápido	El hot reload de Flutter ayuda al desarrollador a que de manera fácil y rápida experimente y construya UIs, añada funcionalidades pre establecidas, y corrija los bugs más rápido. Experimenta tiempos de recarga por debajo de un segundo, sin perder el estado, en emuladores, simuladores, y dispositivos para iOS y Android.
Rendimiento Nativo	Los widgets de Flutter incorporan todas las diferencias críticas entre plataformas como el scrolling, navegación, iconos y fuentes para promocionar un rendimiento totalmente nativo tanto en iOS como en Android.
UI Expresiva y Flexible	La arquitectura en capas les permite a los desarrolladores una completa personalización, que resultan en un renderizado increíblemente rápido y diseños expresivos y flexibles. Adicionalmente posee una galería de widgets pre construidas con Material Design y Cupertino (estilo-iOS) lo que permite realizar aplicaciones con mejor estilo.

Nota. Se describen las principales características de cada uno de los sistemas operativos móviles. Elaborado por: El autor.

La Figura 11 muestra la estructura y los componentes bajo los cuales funciona Flutter, el primer segmento se encuentra enfocado en el desarrollo de aplicaciones utilizando Dart. En el segundo segmento se encuentra el motor de ejecución de las aplicaciones (C/C++), la iteración entre los dos segmentos está dada por Flutter que las gestiona automáticamente. Además, Flutter cuenta con una comunidad activa y en crecimiento, lo que facilita la colaboración y la disponibilidad de paquetes y complementos.

Figura 11

Funcionamiento y Estructura de Flutter



Nota. Se muestra la estructura de funcionamiento de Flutter. Fuente: (Flutter España, 2022).

1.11 BASES DE DATOS

Se tratan de sistemas de almacenamiento organizados y estructurados que permiten almacenar, gestionar y recuperar datos de manera eficiente, son utilizadas para almacenar información permitiendo su posterior lectura, actualización y eliminación de registros de manera controlada y segura. Por su gran utilidad son utilizadas en todo tipo de aplicaciones y entornos, desde pequeñas aplicaciones personales hasta grandes sistemas empresariales; para lo cual, proporcionan un método centralizado y seguro para almacenar y acceder a la información, lo que permite un manejo eficiente de los datos y una mayor capacidad de análisis. La Tabla 4 describe los principales tipos de bases de datos según ORACLE (2022).

Tabla 4*Bases de datos*

Tipo	Descripción
Bases de datos relacionales	Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada.
Bases de datos no relacionales (NoSQL)	Permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas.
Bases de datos OLTP.	Es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones.
Bases de datos distribuidas	Una base de datos distribuida consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes.

Nota. Se describen las principales características de cada uno de los tipos de bases de datos.

Elaborado por: El autor.

1.12 BASES DE DATOS NO RELACIONALES

Son sistemas para el almacenamiento de datos que difieren de las bases antes descritas en su estructura y en la manera en la que almacenan y administran los datos. A diferencia de las relacionales que para administrar los datos lo hacen mediante tablas y relaciones definidas para organizar y estructurar la información las bases de datos no relacionales se basan en modelos de datos flexibles y escalables.

En este sentido, las bases no relacionales ofrecen mayor flexibilidad en el manejo de todo tipo de datos (no estructurados o semiestructurados), lo que hace que se enfoquen especialmente para aplicaciones web, análisis de big data, IoT y aplicaciones móviles, entre otros casos de

uso. Estas bases de datos permiten un rápido acceso a los datos, ya que suelen distribuirse en clústeres o sistemas distribuidos, ofreciendo además una mayor tolerancia a fallos pudiendo manejar grandes volúmenes de datos en entornos de alta carga (Castro, González, & Callejas, 2012).

1.12.1 Apache Cassandra

Se trata de una base distribuida y escalable, diseñada para administrar grandes volúmenes de datos a través de múltiples nodos y clústeres, utiliza un modelo de datos basado en columnas y se destaca por su alta disponibilidad, rendimiento y capacidad de escalabilidad horizontal (ver Figura 12). Esta base de datos ofrece tolerancia a fallos al replicar los datos en varios nodos y permite consultas rápidas y eficientes, especialmente en casos que requieren búsquedas por rango y consultas complejas. Es ampliamente utilizado en entornos de big data y aplicaciones que necesitan manejar grandes cargas de trabajo y ofrecer alta disponibilidad (Apache Cassandra, 2022).

Figura 12

Logo de Apache Cassandra



Nota. Se muestra el logo de la base de datos Apache Cassandra. Fuente: (Apache Cassandra, 2022).

1.12.2 MongoDB

Se enfoca en el almacenamiento y manejo de documentos en formato JSON a través de un modelo de datos documental que proporciona una forma flexible y escalable de organizar, almacenar y consultar datos. Una de las características principales es que no se requiere un esquema fijo o predefinido lo que permite adaptarse fácilmente a cambios en la estructura de los datos sin tener que modificar la base de datos (ver Figura 13). Para lo cual, MongoDB se

base en una estructura de datos basado en colecciones y documentos, donde una colección es el sinónimo de una tabla en una base de datos relacionales y contiene un conjunto de documentos, donde cada documento es un objeto en formato JSON que puede tener campos y valores de diferentes tipos (MongoDB, 2021).

Figura 13

Logo de MongoDB



Nota. Se muestra el logo de MongoDB. Fuente: (MongoDB, 2021).

1.12.3 Firebase

Es una base de datos que permite administrar los datos en tiempo real y la sincronización de la información entre clientes y servidores, se caracteriza por ser altamente escalable y ofrecer una fácil integración con aplicaciones web y móviles a través de sus SDKs (ver Figura 14). Firebase es apreciada por la facilidad de uso y su capacidad para crear aplicaciones en tiempo real con una latencia mínima, llegando a ser una opción para desarrolladores que buscan una plataforma completa y confiable para construir y escalar aplicaciones en la nube (Firebase, 2022).

Figura 14

Logo de Firebase



Nota. Se muestra el logo de la base de datos FireDB. Fuente: (Firebase, 2022).

1.13 BASES DE DATOS RELACIONALES

Son sistemas de gestión de bases de datos que organizan los datos en tablas que se componen de filas y columnas; para lo cual, utilizan el lenguaje de consulta estructurado conocido como SQL (Structured Query Language) que permite almacenar, procesar y recuperar los datos. Este tipo de bases de datos son altamente estructuradas, lo que implica que las relaciones establecidas entre las tablas es través de claves primarias y claves externas lo que garantiza la integridad de los datos y facilita la consulta y la combinación de información de diferentes tablas. Este tipo de bases de datos son ampliamente utilizadas en casi todas las aplicaciones y sectores, ya que ofrecen una sólida consistencia de datos, soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y cuentan con herramientas eficientes para el acceso y análisis de información.

1.13.1 MariaDB

Es una base de código abierto que proviene del proyecto MySQL, actualmente ofrece una amplia gama de características avanzadas, como almacenamiento en caché de consultas, replicación en tiempo real, particionamiento de tablas y seguridad mejorada, destacándose por su enfoque en la escalabilidad y el rendimiento, con optimizaciones de consultas y soporte para múltiples motores de almacenamiento (MariaDB Foundation, 2023).

Figura 15

Logo de MariaDB



Nota. Se muestra el logo de la base de datos MariaDB. Fuente: (MariaDB Foundation, 2023).

Al ser parte de un proyecto de código abierto, cuenta con una comunidad activa que contribuye con mejoras y actualizaciones continuas, por lo que es ampliamente utilizado en diversas

aplicaciones y entornos, desde pequeñas empresas hasta grandes empresas, gracias a su confiabilidad, escalabilidad y flexibilidad (MariaDB Foundation, 2023).

1.13.2 PostgreSQL

Cuenta con características de alto rendimiento llegando a ser una base de datos ampliamente reconocida por su estabilidad, confiabilidad y robustez. En la actualidad cuenta con una amplia gama de características como el soporte para transacciones ACID, integridad referencial, replicación, particionamiento de tablas y consultas complejas; además, se destaca por su arquitectura modular y extensible, lo que permite a los usuarios personalizar y ampliar sus funcionalidades según sus necesidades, la Figura 16 muestra su logo. PostgreSQL es altamente escalable siendo ideal para almacenar y administrar grandes volúmenes de datos y cargas de trabajo exigentes, su comunidad activa de desarrolladores y su enfoque en estándares abiertos la hacen una opción popular tanto para pequeñas aplicaciones como para sistemas empresariales de misión crítica (PostgreSQL, 2023).

Figura 16

Logo de PostgreSQL



Nota. Se muestra el logo de la base de datos PostgreSQL. Fuente: (PostgreSQL, 2023)

1.13.3 SupaBase

Supabase es una plataforma de desarrollo de aplicaciones que ofrece una completa solución para el desarrollo, implementación y despliegue de aplicaciones web y móviles, basado en PostgreSQL y con una API propia, esta base combina una estructura de datos relacional con una interfaz sencilla y segura para interactuar con ella. Esta arquitectura permite a los desarrolladores construir aplicaciones de manera rápida y eficiente, aprovechando todas las

potentes capacidades de PostgreSQL, como la gestión de transacciones ACID y el soporte para datos geoespaciales (Supabase, 2023).

Además esta base de datos, brinda a los programadores una serie de servicios adicionales para simplificar y acelerar el desarrollo como los de autenticación de usuarios con proveedores externos como Google y GitHub, la autorización de acceso a recursos y la gestión de roles y permisos, también ofrece almacenamiento de archivos, permitiendo a los usuarios subir y gestionar documentos, imágenes y otros tipos de archivos de manera segura y eficiente, la Figura 17 muestra el logo de Supabase. Una característica destacada de esta base de datos es su facilidad de uso, la plataforma está diseñada para ser intuitiva y ofrece una documentación detallada y ejemplos prácticos para guiar a los desarrolladores en el proceso de creación de aplicaciones (Supabase, 2023).

Figura 17

Logo de Supabase



Nota. Se muestra el logo de la base de datos Supabase. Fuente: (Supabase, 2023)

Al ser de código abierto, permite el acceso al código fuente para revisarlo, estudiarlo y modificarlo según sus necesidades específicas, esto da lugar a una mayor flexibilidad y personalización, lo que se traduce en la posibilidad de adaptar la plataforma a proyectos particulares y complejos; así también, cuenta con una comunidad activa de desarrolladores que contribuyen y ofrecen soporte técnico.

1.14 ARQUITECTURA WEB

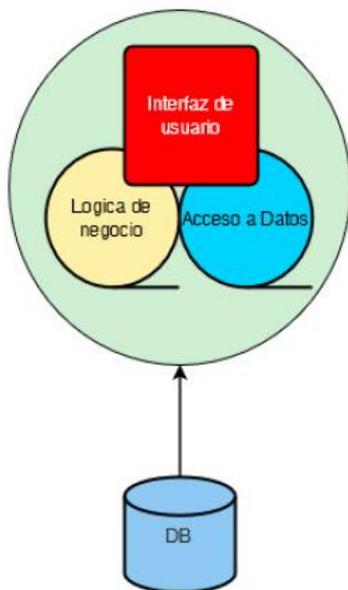
La arquitectura de sistemas de información es considerada como un paradigma al momento de diseñar sistemas informáticos, cuyo fin es el de analizar como el desempeño de cada

componente a nivel de hardware y software afectan directa o indirectamente en el correcto funcionamiento de los requerimientos de funcionamiento del sistema. Para el caso de la arquitectura web su fin es la de gestionar eficazmente la manera en que se encuentra distribuida o estructurada de una manera jerárquica el sistema de información, centrándose en la manera en que la manera en que se distribuyen los servicios a nivel de hardware (Leyva, Alarcón, & Ortigón, 2019). Acorde a lo afirmado por Jiménez, Tello y Ríos (2014) la arquitectura web es “la ciencia que trata el diseño de un sistema de información y del impacto de sus cualidades como: desempeño, seguridad, modificabilidad entre otras” (p. 37), destacándose los siguiente tipo: 1) Arquitectura monolítica, 2) Arquitectura de capas, 3) Arquitectura física de 3 niveles, 4) Arquitectura física multinivel; y, 5) Arquitectura de microservicios

1.14.1 Arquitectura monolítica

Figura 18

Modelo de arquitectura monolítica



Nota. Se muestra el modelo de la arquitectura monolítica. Elaborado por: El autor.

La arquitectura monolítica se popularizó en la década de los 70 cuenta con una estructura única en la que en un solo servidor se encuentra alojada la base de datos, el código fuente con sus distintos servicios como lo muestra la Figura 18. A estos servicios los usuarios acceden

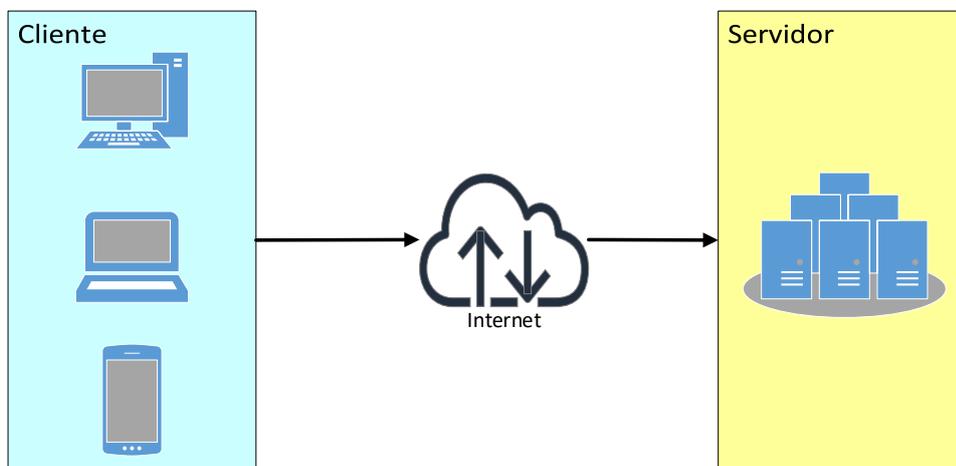
mediante páginas web o REST API que permiten la comunicación entre sistemas de información que usan el protocolo HTTP (Filgueira, 2014).

1.14.2 Arquitectura cliente servidor

En este tipo de arquitectura, el cliente hace referencia a la interfaz de usuario o a la aplicación que solicita y consume servicios, mientras que el servidor es el encargado de procesar y proporcionar los servicios solicitados por el cliente acorde a lo descrito en la Figura 19.

Figura 19

Modelo de arquitectura cliente servidor



Nota. Se muestra la arquitectura del modelo cliente servidor. Elaborado por: El autor.

Este tipo de arquitectura permite una distribución de tareas más eficiente permitiendo la separación de responsabilidades y una mayor escalabilidad, garantizando el procesamiento y almacenamiento de datos centralizado en el servidor, mientras que la interfaz para la presentación de datos se maneja del lado del cliente. Este tipo de arquitectura es ampliamente utilizada en aplicaciones web, donde el cliente ejecuta acciones en un navegador web y se comunica con el servidor a través de solicitudes HTTP (KYOCERA, 2020).

1.14.3 Arquitectura física de 3 niveles

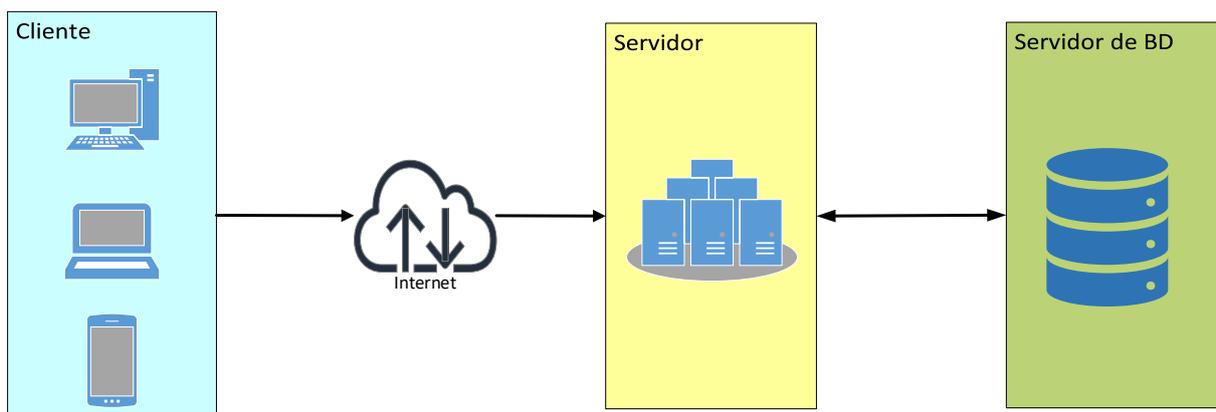
Este tipo de arquitectura se centra en un enfoque de diseño de sistemas en el que las funcionalidades se dividen en tres capas:

- Capa de presentación: se encarga de la interfaz de usuario y la interacción con los usuarios finales.
- Capa de lógica de negocio: se ocupa de la lógica y las reglas empresariales
- Capa de almacenamiento de datos: se encarga de la persistencia y recuperación de datos

En este sentido, cada capa tiene su propia responsabilidad y comunicación definida con las capas adyacentes, lo que permite la división de tareas facilitando la modificación, el mantenimiento y la escalabilidad del sistema. Esta arquitectura es comúnmente utilizada en aplicaciones empresariales y web, ya que permite un mayor modularidad, reutilización de código y flexibilidad en la implementación de sistemas (Acosta, Álvarez, & Gordillo, 2019).

Figura 20

Modelo de arquitectura física de 3 niveles



Nota. Se muestra el modelo de la arquitectura física de 3 niveles. Elaborado por: El autor.

1.14.4 Arquitectura física multinivel

La arquitectura física multinivel es un enfoque de diseño de infraestructuras de tecnología de la información que divide los componentes y servicios en múltiples niveles o capas físicas. En este tipo de arquitectura, cada nivel está dedicado a una función específica y se encarga de gestionar y proporcionar servicios relacionados, permitiendo que cada capa pueda estar compuesta por diferentes elementos físicos, como servidores, dispositivos de almacenamiento, enrutadores, conmutadores, entre otros. La arquitectura física multinivel permite una mayor

flexibilidad y escalabilidad al distribuir la carga de trabajo y los recursos en múltiples niveles, incrementando la capacidad de rendimiento, la disponibilidad y reducir considerablemente los tiempos de respuesta del sistema. Adicionalmente, facilita la administración y el mantenimiento al separar claramente las responsabilidades y las funciones en cada nivel.

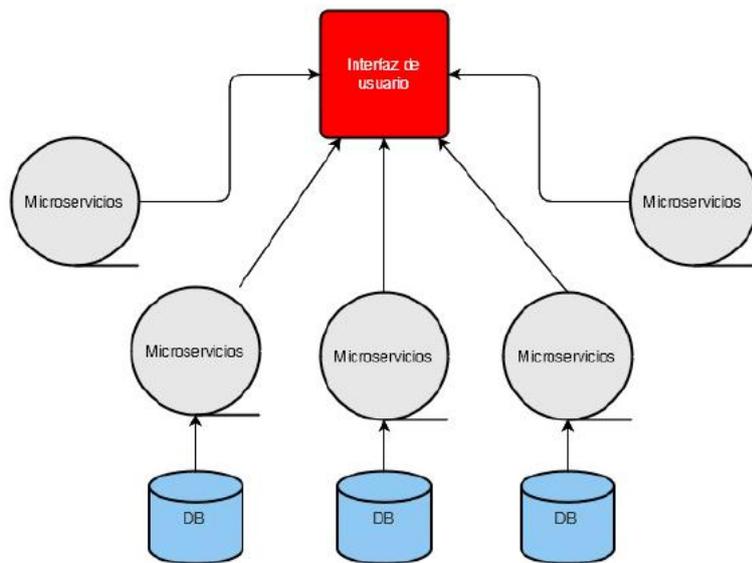
1.14.5 Arquitectura de microservicios

Es un enfoque que se centra en el desarrollo de aplicaciones como un conjunto armónico y sincronizado de pequeños servicios, independientes y autónomos, cada uno con su propia funcionalidad específica que son capaces de comunicarse entre ellos mediante interfaces claramente definidas. Cada servicio se desarrolla y se despliega de manera independiente, lo que permite una mayor flexibilidad y escalabilidad en comparación con las arquitecturas monolíticas tradicionales acorde a lo descrito en la Figura 21.

Los microservicios se pueden implementar utilizando diferentes tecnologías y lenguajes de programación, y cada uno puede tener su propia base de datos y su ciclo de vida individual. Esto facilita la mejora continua, el mantenimiento y la evolución de cada servicio sin afectar a los demás. Además, los microservicios promueven la modularidad, el desacoplamiento y la reutilización de código, lo que facilita el desarrollo ágil y permite a los equipos trabajar de manera independiente en diferentes servicios. Sin embargo, esta arquitectura también introduce desafíos adicionales, como la gestión de la comunicación entre servicios y el monitoreo y la administración de múltiples componentes distribuidos (Ordóñez, 2021).

Figura 21

Modelo de arquitectura de microservicios



Nota. Se muestra el modelo de la arquitectura de microservicios. Elaborado por: El autor.

1.15 METODOLOGÍAS TRADICIONALES PARA EL DESARROLLO DE SOFTWARE

Este tipo de metodologías vuelcan sus energías en el desarrollo de una planificación del proyecto que abarque todas las actividades que se van a realizar, tratando de identificar cada uno de los detalles, por más mínimo que sea, para su ejecución durante el ciclo de vida del proyecto tecnológico. Como una de sus desventajas es que este tipo de metodologías son incapaces de adaptarse rápidamente a los enfoques cambiantes en los que actualmente se desarrollan los sistemas informáticos ya que al tener una rígida planificación de actividades y tiempos su cambio se vuelve un trámite engorroso y demorado (Parra, 2021).

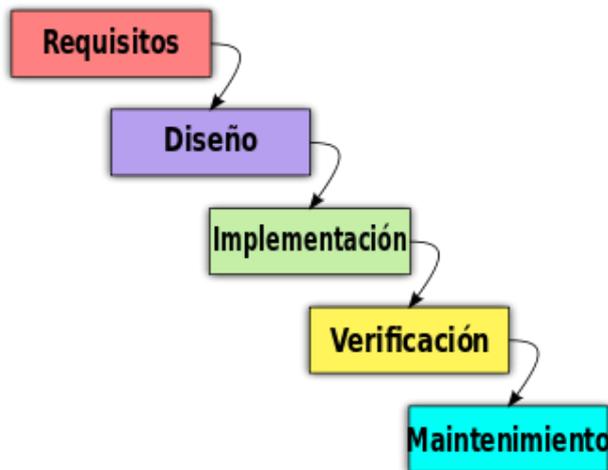
El primer modelo conocido de este tipo de metodologías fue denominado “modelo en cascada”, mismo que propone una estricta secuencia de pasos desde la identificación de requisitos hasta el mantenimiento del sistema ya desarrollado (Delgado, 2019) cómo se describe en la Figura 22. Sobre la base de este modelo se fueron construyendo otras metodologías que al igual que su predecesora estuvieron compuestas por procesos rígidos y secuenciales.

En general las metodologías tradicionales tienden a percibir a todo el proyecto como una sola actividad grande, en una sola dirección que no permite regresar a un punto anterior, ya que el

levantamiento de los requerimientos se los realiza una sola vez utilizando mucho tiempo tratando de garantizar que nada se escape por lo cual se espera que no haya cambios en el camino (Delgado, 2019).

Figura 22

Fases del modelo en cascada



Nota. Se muestran las fases del modelo en cascada. Elaborado por: El autor.

1.16 METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE

Este tipo de metodologías se centran en adecuarse a la forma de trabajo y a las condiciones del proyecto consiguiendo respuestas rápidas a los cambios (Garrido Sotomayor, 2021). Se caracterizan principalmente por su flexibilidad, capacidad de adaptación y por ser iterativos, repitiéndose las veces que se consideren necesarias durante todo el desarrollo del proyecto.

Desde su aparición en el mercado han ido ganando cada vez más espacio, según el estudio anual del Project Management Institute (PMI) para el 2020 a nivel mundial 7 de cada 10 organizaciones afirman haber utilizado al menos una vez en el desarrollo de sus proyectos tecnológicos alguna de las metodologías ágiles (Finanzas, 2021).

Acorde a lo descrito en la Tabla 5 se diferencian principalmente de las metodologías tradicionales en la capacidad que tienen de adaptarse en los cambios sobre los requisitos de los clientes y sobre la solución que se encuentra desarrollando debido a que sus procesos son

iterativos a través de entregas anticipadas en las que el cliente puede verificar los avances parciales sin tener que esperar que la solución se encuentre completamente finalizada.

Tabla 5

Comparación de metodología tradicional vs ágil

Criterio	Metodología Tradicional	Metodología Ágil
Se basa en	Proceso	Personas
Tipo de proceso	Proceso Predictivo	Proceso adaptativo
Estructura	Lineal/Secuencial	Iterativa
Recopilación de requisitos	Se recopila una sola vez al inicio del proyecto	Son dinámicos, aparecen en el transcurso del proyecto
Involucramiento del cliente	Es baja la participación del cliente	El involucramiento es alto
Gestión del proyecto	Su gestión es centralizada, Se encarga de la gestión solo el director de proyecto	Su gestión es descentralizada, hay varias personas del equipo que se encargan de la gestión.
Equipo	El equipo es especializado en cada tema en específico	El equipo es flexible y multidisciplinario
Preferencia	Prefiere la anticipación y el control estricto desde el inicio	Prefiere la adaptabilidad durante el desarrollo del proyecto
ROI	El ROI se realiza al final del proyecto	El ROI se desarrolla durante el proyecto
Retroalimentación	La retroalimentación es nula	La retroalimentación llega de acuerdo con las necesidades que van apareciendo

Nota. Se describen las principales características de las metodologías ágiles. Elaborado por:

El autor.

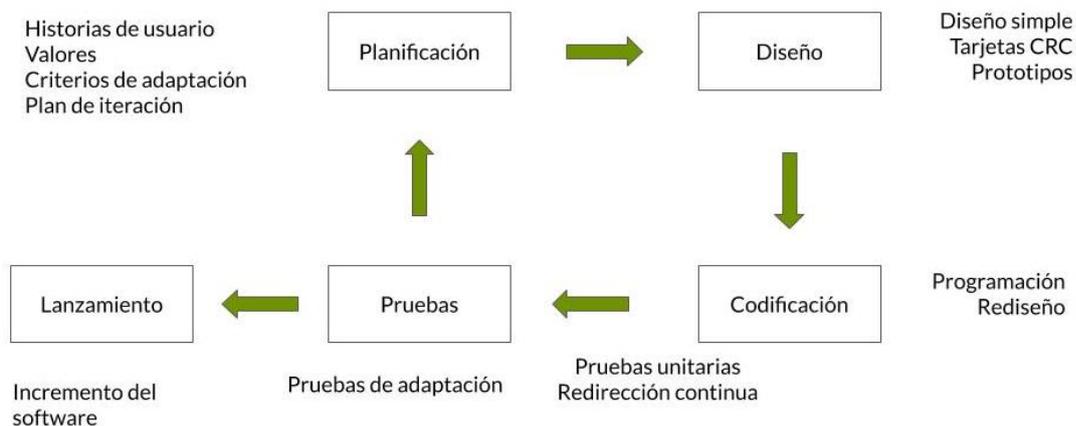
1.16.1 Metodología ágil XP

La metodología XP se caracteriza por realizar el levantamiento de los requisitos de los usuarios a modo de historias, centrandó su interés en describir detalladamente cada una de las

funcionalidades y acciones que el cliente requiere que el sistema realice para solucionar sus problemas como se describe en la Figura 23. Para lo cual, el desarrollo lo realiza a mediante ciclo cortos de desarrollo del sistema cuyos productos intermedios son presentados al cliente para recibir su retroalimentación, cada vez que se finaliza una iteración éste entra a la fase de pruebas en la que el usuario aprueba su completo funcionamiento. El uso de esta herramienta se ha popularizado en startups o empresas en proceso de integración, ayudando a mejorar la relación empresa-cliente (Garrido Sotomayor, 2021).

Figura 23

Fases de la metodología XP



*Nota. Se muestran las fases de la metodología ágil XP son sus interacciones. Elaborado por:
El autor.*

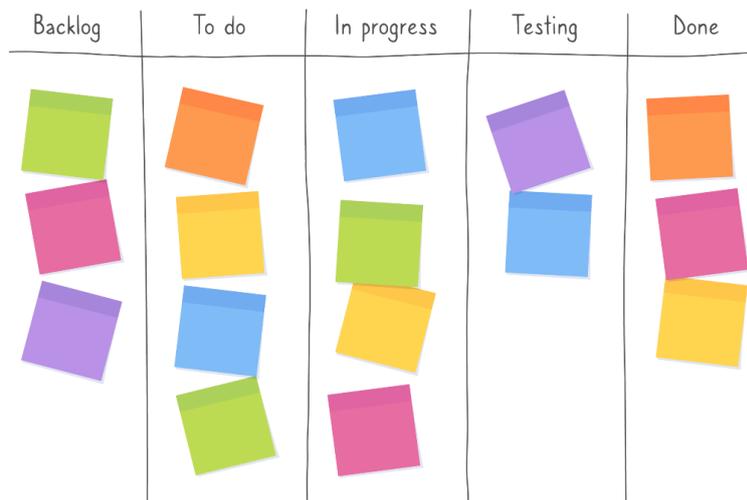
1.16.2 Metodología ágil Kanban

La metodología Kanban, conocida también como "Tarjetas visuales", les permite a los directores de proyectos desarrollar una tabla o gráfico de las tres columnas de tareas: pendientes, en progreso, en pruebas o completadas. Acorde a la descripción de la Figura 24 es una herramienta centrada en alertas visuales sobre el tablero de trabajo en el equipo en su conjunto interactúan, mitigando la duplicidad de tareas o el olvido de alguna. Por lo tanto, genera mejores canales de comunicación del equipo de trabajo incrementando su productividad (Garrido Sotomayor, 2021). En este sentido, esta metodología se enfoca en el desarrollo de

diagramas de flujo de valor para el desarrollo de proyectos, procurando mejorar la organización individual y conjunta del equipo de trabajo.

Figura 24

Tablero Kanban



Nota. Se muestran las fases de la metodología ágil Kanban. Fuente: (Zamora, 2021)

Kanban tiene como objetivo limitar al máximo la acumulación de tareas en el work in progress, lo que influye directamente en el incremento del nivel de productividad de los equipos de trabajo. En esta metodología se identifican dos roles fundamentales dentro de su flujo organizativo:

- Service Request Manager: su función es la de comprender las necesidades del cliente para posteriormente comunicarlas al equipo de trabajo de la manera más clara y precisa.
- Service Delivery Manager: su función es la de entender los requisitos del cliente y definir expectativas apegadas a la realidad sobre el alcance, la calidad y los factores de riesgo del servicio a ser desarrollado.

En Kanban se centra en cuatro fases:

Tabla 6

Fases de la metodología Kanban para el desarrollo de software

Fase	Descripción
Fase 1	Introducción al personal acerca de la metodología Kanban. En esta fase todos los miembros del equipo se familiarizan con el uso de la metodología Kanban y los beneficios de su aplicabilidad.
Fase 2	Implementación de la metodología Kanban con todos los componentes con más problemas. Se aplica a los componentes que tengas más dificultades, enfatizando así resaltarlos para garantizar la facilidad de ejecución y desarrollo.
Fase 3	Implementación de Kanban en los componentes restantes. Se implementa solo cuando se hayan encontrado soluciones a problemas que hayan sido detectados como exhaustos.
Fase 4	Se realiza una revisión exhaustiva del sistema para determinar qué puntos que deben reordenarse. Se debe constatar que ningún trabajo se realice fuera de secuencia.

Nota. Se describen las fases de la metodología Kanban. Elaborado por: El autor.

1.16.3 Metodología ágil SCRUM

SCRUM se caracteriza por fomentar una estructura de desarrollo incremental, en la que el desarrollo de cada ciclo de productos se divide en "pequeños proyectos" en el que se deben cumplir 3 fases (análisis, desarrollo y pruebas de funcionamiento). La ejecución de la fase de desarrollo se centra en definir las interacciones de proceso o sprints, entregas periódicas de los productos intermedios para revisión y pruebas de funcionamiento de los usuarios finales. Este enfoque permite mejorar la gestión de los proyectos complejos que se caracterizan por que los clientes demandan de flexibilidad y velocidad en la entrega de los sistemas de información (Garrido Sotomayor, 2021).

Acorde a lo descrito en la Figura 25, en esta metodología la definición del alcance del producto se realiza a través de la identificación del Product Backlog, que es el instrumento en el que se recogen todas las actividades que el cliente desea que el sistema realice para satisfacer sus

necesidades, mismas que son desarrolladas en cada uno de los sprints que se definen. Las fases de la metodología ágil SCRUM son: 1) Planeación y estimación, 2) Diseño, 3) Desarrollo, 4) Pruebas de funcionamiento, 5) Despliegue, 6) Revisión; y, 7) Lanzamiento

Figura 25

Fases para la ejecución de los sprints



Nota. Se muestran las fases que se ejecutan por cada sprint. Fuente: (Zamora, 2021)

En esta metodología se identifican dos roles fundamentales dentro de su flujo organizativo:

- Product Owner: se encarga de tomar las decisiones sobre los cambios que se requieren teniendo la visión clara del producto a desarrollar.
- ScrumMaster: se encargada de recopilar y consolidar las necesidades del cliente y de vigilar que el modelo y la metodología funcionen correctamente, se encarga adicionalmente de solventar los inconvenientes presentados durante la ejecución de este.
- Equipo de desarrollo: se conforma por un grupo de menos de 10 personas encargada de escribir el código de los sistemas.
- Usuarios: persona o personas que serán los usuarios finales del sistema de información.

CAPÍTULO III

METODOLOGÍA

El desarrollo se lo realizará utilizando la metodología ágil SCRUM, considerando que acorde a lo descrito en el capítulo anterior tiene un enfoque dinámico para la implementación de sistemas informáticos en general, centrándose en iteraciones rápidas que buscan satisfacer las necesidades del cliente con entregas rápidas y continuas evitando resultados finales insatisfactorios.

Considerando que la finalidad de la presente investigación es la implementación de una aplicación móvil que mejore el proceso de evaluación docente al interior de la Universidad, se determina que esta investigación es del tipo aplicada tecnológica. La Figura 26 describe las fases de la metodología de trabajo a ser aplicada.

Figura 26

Fases de la metodología de trabajo



Nota. Se describe las fases identificadas para el desarrollo de la presente investigación.

Elaborado por: El autor.

1.17 Recopilación documental

Para la obtención de las fuentes de información documental se recurrirá a portales académicos como Redalyc, Google Académico y la red de repositorios de tesis de Universidades Latinoamericanas. La búsqueda de información se realizará mediante palabras claves y operadores lógicos en los casos que las palabras usadas eran sinónimos; para lo cual, los criterios de inclusión considerados son:

- Búsqueda de la información.
- Palabras clave: sistemas de información, SCRUM, aplicaciones móviles, metodologías ágiles, arquitectura de sistemas.
- Año de publicación: de preferencia se seleccionaron artículos y tesis elaboradas en los últimos 5 años; sin embargo, también se incluyeron referencias de años anteriores de autores relevantes.

Los criterios de exclusión aplicados fueron los siguientes:

- Información que no guarde relación directa con el tema y objetivos de la presente investigación.
- Información que provenga de fuentes poco confiables, estudios que no presenten rigurosidad científica, en el caso de artículos, aquellos que hayan sido publicados en revistas que no se encuentren indexadas.

1.17.1 Metodología SCRUM

La metodología seleccionada para el desarrollo e implementación del aplicativo móvil es SCRUM considerando que se fundamenta en aplicar un enfoque ágil basado en la flexibilidad, la colaboración y la entrega incremental de los productos. Esta metodología, se caracteriza por dividir a un proyecto en ciclos más pequeños llamados “sprints”, que en promedio tienden a durar hasta cuatro semanas, para cada sprint se realiza una reunión de planificación para definir las tareas a realizar y establecer el objetivo a cumplir.

Durante cada sprint, el equipo trabaja en esas tareas de manera colaborativa y autoorganizada llevando a cabo reuniones diarias de seguimiento, conocidas como “scrum diarios”, en las que los miembros del equipo comparten el progreso, identifican obstáculos y coordinan sus esfuerzos. Al finalizar el sprint, se realiza una revisión para mostrar los resultados y recopilar comentarios; así también, se lleva a cabo reuniones de evaluación que permitan reflexionar

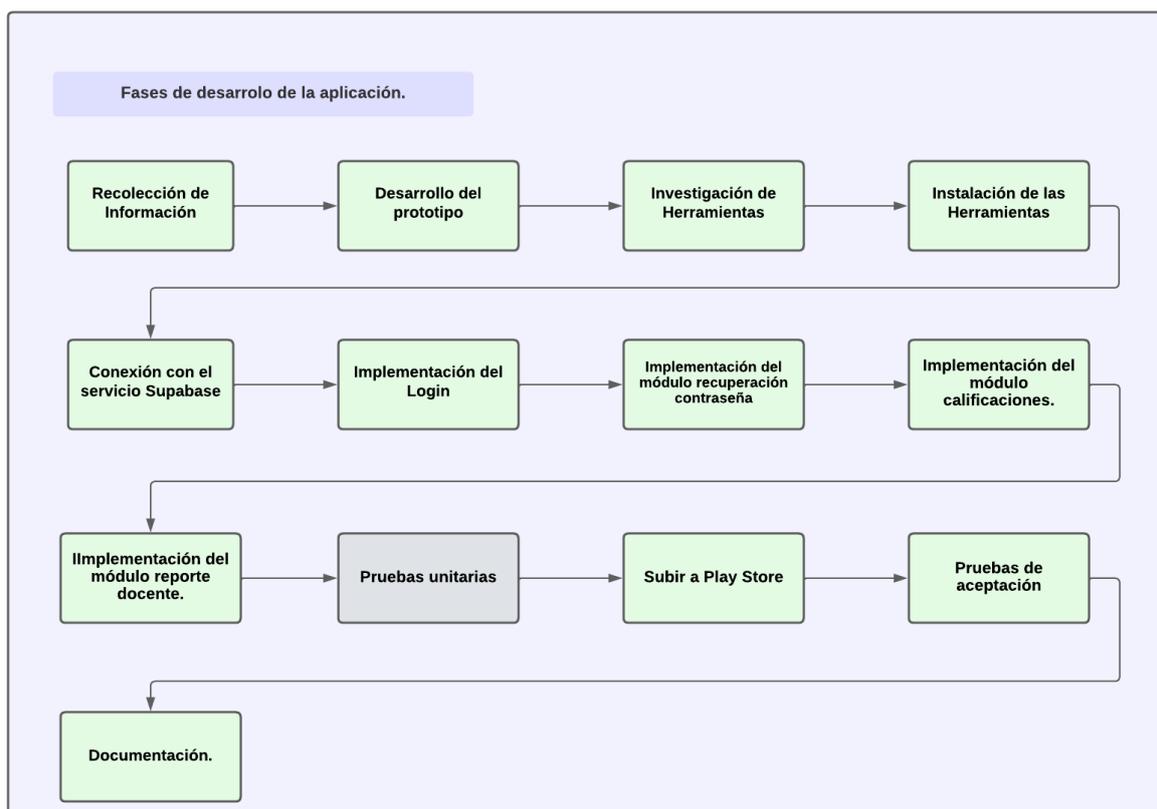
sobre los puntos fuertes y débiles encontrados durante la ejecución del proceso y de esta manera buscar incorporar acciones de mejora continua.

Adicionalmente, SCRUM se caracteriza por tener un enfoque centrado en la adaptabilidad, la transparencia y la comunicación fluida entre todos los miembros del equipo y con la contraparte del proyecto, todo esto con el objetivo principal de entregar valor de manera rápida y constante, manteniendo la capacidad de responder rápidamente a los cambios y las necesidades del cliente.

1.18 DIAGRAMA DE BLOQUES

Figura 27

Fases de evolución de la aplicación



Nota. Se describe el diagrama fases de evolución de la aplicación. Elaborado por: El autor.

1.19 SPRINT 0

A este sprint se lo conoce también como fase inicial del proyecto de desarrollo, se le dedica en promedio una semana y como máximo dos semanas, tiempo en el que se prepara al equipo en los ámbitos tecnológicos y metodológicos con el objetivo de garantizar que el proyecto tenga un buen comienzo.

1.19.1 Especificación de roles

Acorde a lo descrito en la Tabla 7, los roles se definieron considerando la formación y capacidades profesionales de cada persona: Holger Ortega cuenta con alta experiencia en el levantamiento de requerimientos funcionales con los usuarios finales (Product Owner), asimismo el rol de Scrum Master y de desarrollador estará a cargo de Bryan Morales considerando sus conocimientos de la metodología SCRUM y de su experiencia en el desarrollo de aplicaciones móviles.

Tabla 7

Definición de roles del proyecto

Rol	Persona	Descripción
Product Owner	Holger Ortega	Negocia decisiones sobre el producto con los usuarios.
Scrum Master	Bryan Morales	Guía el equipo en la metodología SCRUM
Equipo de Desarrollo	Bryan Morales	Encargados de diseñar y desarrollar la aplicación.

Nota. Se describen los roles definidos para la aplicación de SCRUM. Elaborado por: El autor.

1.19.2 Declaración del producto

Este aplicativo móvil está enfocado en satisfacer las necesidades de contar con un aplicativo móvil para la evaluación continua del desempeño docente por parte de los alumnos al finalizar cada clase. Esto facilitará el proceso actual que es realizado una vez cada semestre, de la cual no todos los alumnos participan.

1.19.3 Creación del área de trabajo

Para el desarrollo del proyecto es necesario definir las herramientas que se utilizará para el desarrollo de la app, garantizando contar con el acceso a todas a sus últimas versiones más estables; para lo cual, en la Tabla 8 se detallan las herramientas seleccionadas.

Tabla 8

Definición de herramientas para el desarrollo

Ítem	Descripción
Lenguaje de programación	Dart
Framework	Flutter
Base de datos	Supabase

Nota. Se describen las herramientas seleccionadas para el desarrollo de la aplicación móvil.

Elaborado por: El autor.

1.19.4 Product backlog o pila de producto

Hace referencia a la lista de funcionalidades priorizadas que deben ser parte del sistema de información, mismas que han sido identificadas en trabajos conjuntos con los usuarios finales. El resultado de esta actividad finalmente le servirá al equipo desarrollador como los requerimientos funcionales para la programación del aplicativo móvil. En la Tabla 9 se describen todas las funcionalidades de la aplicación móvil.

Tabla 9

Pila de productos

ID	COMO	QUIERO	PARA
H001	Administrador	Registrar información de los usuarios.	Tener control de los usuarios autorizados a utilizar el sistema.
H002	Administrador	Actualizar información de los usuarios.	Actualizar la información de los usuarios.
H003	Administrador	Asignar permisos para el uso del sistema	Tener el control de las acciones que los usuarios

ID	COMO	QUIERO	PARA
			pueden realizar por tipo de usuario.
H004	Administrador	Administrar todos los módulos del sistema	Monitorear y supervisar el funcionamiento de todos los módulos.
H005	Usuario	Utilizar la App en sistemas operativos Android	Permitir el funcionamiento de la App en todos los dispositivos móviles.
H006	Usuario	Ingresar al sistema con mis credenciales personales (usuario y contraseña).	Seguridad de mi actividad en el sistema.
H007	Usuario	Ingresar al sistema desde mi dispositivo móvil luego de terminar las clases	Controlar que la evaluación se haga posterior a terminada la clase.
H008	Usuario	Permitir evaluar únicamente al docente que terminó su clase.	Evitar errores al momento de realizar la evaluación.
H009	Usuario	Mantener un histórico de las evaluaciones realizadas.	Tener control de las actividades realizadas en la App.
H010	Administrador	Activar e inactivar usuarios.	Tener control de los usuarios que pueden acceder al sistema.
H011	Administrador	Mantener la confidencialidad de las evaluaciones.	Garantizar transparencia en el proceso.
H012	Docente	Generar reportes para la evaluación periódica.	Contar con información precisa para la evaluación docente.

Nota. Se describen las actividades identificadas que deben ser automatizadas a través del sistema de información. Elaborado por: El autor.

1.19.5 Refinamiento del product backlog

El objetivo de esta actividad es la organización de las actividades con un orden lógico y agrupándolas acorde al sprint en el que serán desarrolladas; para alcanzar este objetivo, se desarrollaron las siguientes actividades:

- Ordenar el product backlog en una secuencia lógica para su posterior desarrollo.
- Analizar el eliminar ítems.
- Dividir ítems en unos más pequeños.
- Unir ítems en unos más grandes

Tabla 10

Refinamiento de la pila de productos

ID	Orden	COMO	QUIERO	PARA
H001	1	Administrador	Registrar información de los usuarios.	Tener control de los usuarios autorizados a utilizar el sistema.
H002	1	Administrador	Actualizar información de los usuarios.	Actualizar la información de los usuarios.
H003	2	Administrador	Asignar permisos para el uso del sistema	Tener el control de las acciones que los usuarios pueden realizar por tipo de usuario.
H005	2	Usuario	Utilizar la App en sistemas operativos Android.	Permitir el funcionamiento de la App en todos los dispositivos móviles.
H006	2	Usuario	Ingresar al sistema con mis credenciales personales (usuario y contraseña).	Seguridad de mi actividad en el sistema.

ID	Orden	COMO	QUIERO	PARA
H008	3	Usuario	Permitir evaluar únicamente al docente que terminó su clase.	Evitar errores al momento de realizar la evaluación.
H009	3	Usuario	Mantener un histórico de las evaluaciones realizadas.	Tener control de las actividades realizadas en la App.
H012	3	Docente	Generar reportes para la evaluación periódica.	Contar con información precisa para la evaluación docente.
H010	4	Administrador	Activar e inactivar usuarios.	Tener control de los usuarios que pueden acceder al sistema.
H011	4	Administrador	Mantener la confidencialidad de las evaluaciones.	Garantizar transparencia en el proceso.
H004	5	Administrador	Administrar todos los módulos del sistema	Monitorear y supervisar el funcionamiento de todos los módulos.
H007	6	Usuario	Ingresar al sistema desde mi dispositivo móvil luego de terminar las clases	Controlar que la evaluación se haga posterior a terminada la clase.

Nota. Se describen las actividades organizadas de tal manera que respondan a una secuencia lógica de programación. Elaborado por: El autor.

1.19.6 Planificación de los sprints

El orden de los sprint para el desarrollo fue definido por el equipo SCRUM en conjunto, llegando a consensos para seleccionar y comprender el trabajo que será realizado. Así también, el desarrollador junto con el product Owner acordaron el tiempo en que se realizará cada sprint, considerando el rendimiento del equipo de desarrollo y el número de tareas necesarias para el

desarrollo de cada ítem del product backlog. Considerando la auto organización se establecieron los siguientes sprints.

La Tabla 11 describe los 5 ítems del Product Backlog que serán desarrollados en el primer sprint; para lo cual, se estima un tiempo de tres semanas.

Tabla 11

Ítems del producto backlog para el primer sprint

ID	COMO	QUIERO	PARA
H001	Administrador	Registrar información de los usuarios.	Tener control de los usuarios autorizados a utilizar el sistema.
H002	Administrador	Actualizar información de los usuarios.	Actualizar la información de los usuarios.
H003	Administrador	Asignar permisos para el uso del sistema	Tener el control de las acciones que los usuarios pueden realizar por tipo de usuario.
H005	Usuario	Utilizar la aplicación en sistemas operativos Android.	Permitir el funcionamiento de la aplicación en todos los dispositivos móviles.
H006	Usuario	Ingresar al sistema con mis credenciales personales (usuario y contraseña).	Seguridad de mi actividad en el sistema.

Nota. Se describen las actividades a ser ejecutadas en el primer sprint. Elaborado por: El autor.

La Tabla 12 describe los 3 ítems del product backlog que serán desarrollados en el segundo sprint; para lo cual, se estima un tiempo de ocho semanas.

Tabla 12*Ítems del producto backlog para el segundo sprint*

ID	COMO	QUIERO	PARA
H008	Usuario	Permitir evaluar únicamente al docente que terminó su clase.	Evitar errores al momento de realizar la evaluación.
H009	Usuario	Mantener un histórico de las evaluaciones realizadas.	Tener control de las actividades realizadas en la aplicación.
H012	Docente	Generar reportes para la evaluación periódica.	Contar con información precisa para la evaluación docente.

Nota. Se describen las actividades a ser ejecutadas en el segundo sprint. Elaborado por: El autor.

La Tabla 13 describe los 4 ítems del product backlog que serán desarrollados en el tercer sprint; para lo cual, se estima un tiempo de cuatro semanas.

Tabla 13*Ítems del producto backlog para el tercer sprint*

ID	COMO	QUIERO	PARA
H010	Administrador	Activar e inactivar usuarios.	Tener control de los usuarios que pueden acceder al sistema.
H011	Administrador	Mantener la confidencialidad de las evaluaciones.	Garantizar transparencia en el proceso.
H004	Administrador	Administrar todos los módulos del sistema	Monitorear y supervisar el funcionamiento de todos los módulos.
H007	Usuario	Ingresar al sistema desde mi dispositivo móvil luego de terminar las clases	Controlar que la evaluación se haga posterior a terminada la clase.

Nota. Se describen las actividades a ser ejecutadas en el tercer sprint. Elaborado por: El autor.

1.19.7 Definición de finalizado

Con el objetivo de evitar ambigüedades al momento de calificar como terminada o no una tarea, se definen los parámetros, contenidos en la Figura 28, que le permita al equipo de una manera sencilla determinar que las tareas han sido completadas satisfactoriamente.

Figura 28

Listado para definir la finalización de un sprint

Definición de finalizado	
<input type="checkbox"/>	Se reviso el diseño.
<input type="checkbox"/>	Código Completo
<input type="checkbox"/>	El programa realizo el Build sin ningún error.
<input type="checkbox"/>	Código inspeccionado
<input type="checkbox"/>	Código esta comentado.
<input type="checkbox"/>	Se actualizo la documentación.
<input type="checkbox"/>	Pruebas
<input type="checkbox"/>	Pruebas unitarias
<input type="checkbox"/>	Pruebas de integración
<input type="checkbox"/>	Aceptación del PO
<input type="checkbox"/>	Funcional en diferentes navegadores.

Nota. Se describen los parámetros identificados para la definición de la tarea terminada.

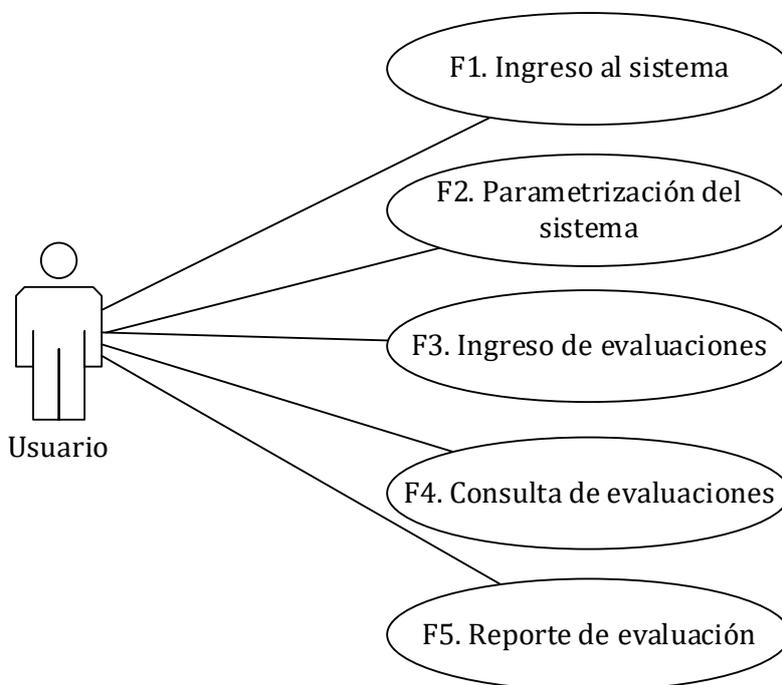
Elaborado por: El autor.

1.19.8 Diagrama de casos de uso general

Los casos de uso se refieren a representaciones gráficas en las que los usuarios pueden interactuar con las funcionalidades del sistema. Para la presente investigación Figura 29 contiene el diagrama general cuyo objetivo es brindar un contexto general de las funcionalidades que tendrá el sistema para posteriormente representar los respectivos diagramas de cada una de las funcionalidades.

Figura 29

Diagrama de casos de uso general

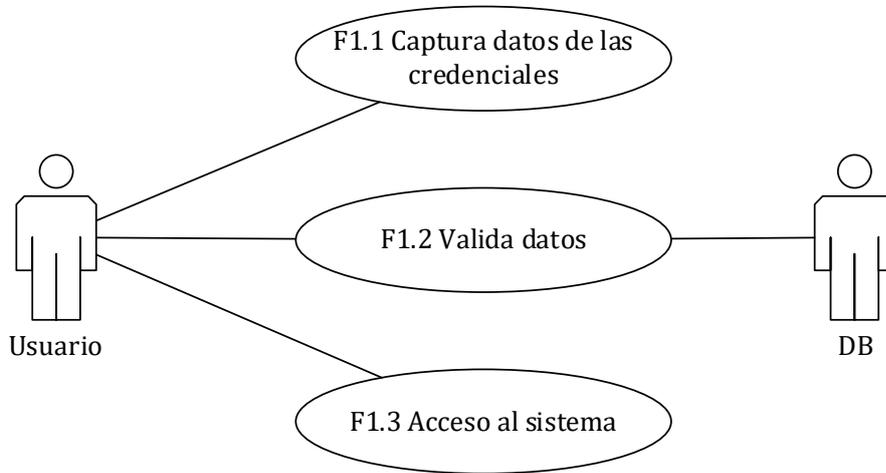


Nota. Se describe el diagrama de casos de uso general del sistema. Elaborado por: El autor.

1.19.9 Diagrama de casos de uso para el ingreso al sistema

Figura 30

Diagrama de casos de uso de ingreso al sistema



Nota. Se describen las actividades del diagrama para el ingreso al sistema. Elaborado por: El autor.

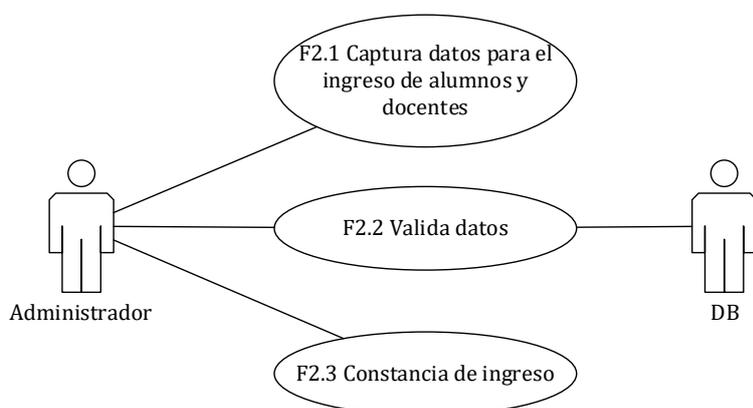
Caso de uso:	Ingreso al aplicativo móvil
Descripción	Este caso de uso describe la manera en que los usuarios pueden validar su identidad ante el sistema, para lo cual se muestra en pantalla los campos necesarios para que el usuario digite sus credenciales de acceso.
Actores:	Usuario
Actividades:	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación. 2. El sistema solicita el ingreso de credenciales. 3. El usuario ingresa sus credenciales (nombre de usuario y contraseña). 4. La aplicación verifica que la información ingresada coincida con la existente en la base de datos 5. La App da acceso.
Excepciones	<p>El usuario no se encuentra registrado</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un error indicando que el las credenciales suministradas no se encuentran registradas para interactuar en el aplicativo móvil, regresa al paso 3.

<p>La contraseña es incorrecta</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un error indicando que la contraseña es incorrecta, regresa al paso 3.
<p>Pre condición:</p> <p>El usuario debe estar registrado en el sistema.</p>
<p>Post condición:</p> <p>El sistema permite el acceso a la aplicación.</p>

1.19.10 Diagrama de casos de uso para la parametrización del sistema

Figura 31

Diagrama de casos de uso para la parametrización del sistema



Nota. Se describe el diagrama de casos de uso para la parametrización del sistema. Elaborado por: El autor.

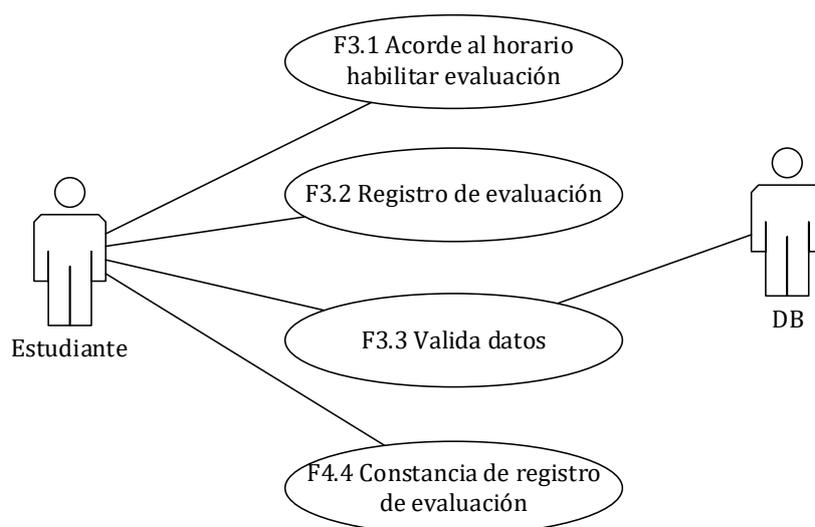
Caso de uso:	Parametrización del aplicativo móvil
Descripción	Se describe la manera en la que el administrador registrar la información de alumnos y docentes quienes posteriormente podrán operar en el sistema.
Actores:	Administrador
Actividades:	<ol style="list-style-type: none"> 1. El administrador ingresa a la página web. 2. El administrador ingresa sus credenciales de acceso. 3. La App muestra los perfiles habilitados para el usuario. 4. El usuario selecciona el perfil de Administrador.

<ol style="list-style-type: none"> 5. La App muestra las opciones para el ingreso de información de alumnos y docentes. 6. El usuario registra toda la información solicitada. 7. La App valida que todos los datos cumplan con las reglas del negocio establecidas. 8. La App muestra un mensaje que indica que la información ha sido ingresada correctamente.
<p>Excepciones</p> <p>Información incompleta</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un mensaje indicando la información que falta por completar, regresa al paso 6. <p>Información duplicada</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un error indicando que la información que se pretende ingresar ya ha sido registrada anteriormente, regresa al paso 6.
<p>Pre condición:</p> <p>El usuario debe estar registrado y debe tener asignado el perfil de administrador.</p>
<p>Post condición:</p> <p>El sistema permite la parametrización del sistema.</p>

1.19.11 Diagrama de casos de uso para el ingreso de evaluaciones

Figura 32

Diagrama de casos de uso para el ingreso de evaluaciones



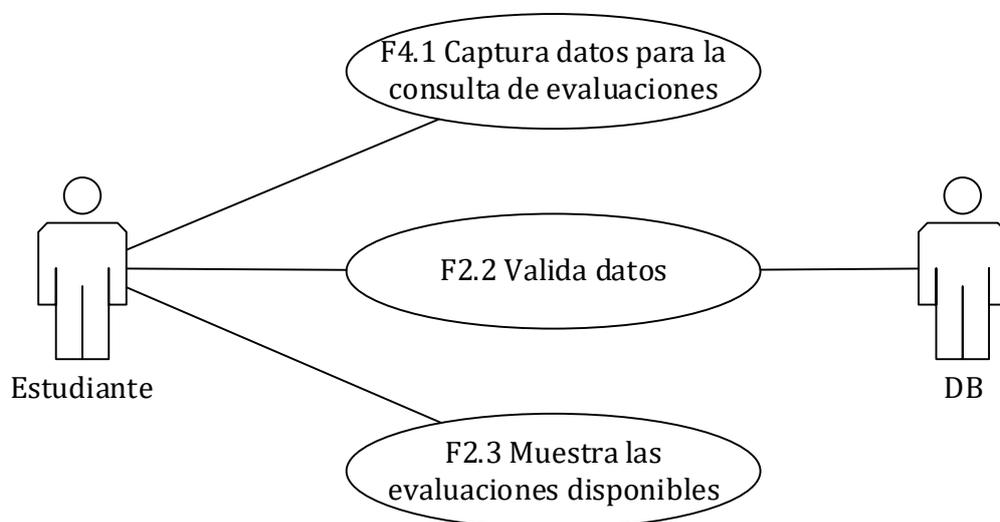
Nota. Se describe el diagrama de casos de uso para el registro de evaluaciones a los docentes por parte de los estudiantes. Elaborado por: El autor.

Caso de uso:	Ingreso de evaluaciones
Descripción	Este caso de uso permite al estudiante cada vez que termine una clase de manera voluntaria registrar la evaluación a su docente.
Actores:	Estudiante
Actividades:	<ol style="list-style-type: none"> 1. El usuario ingresa a la App. 2. El usuario ingresa sus credenciales de acceso. 3. La App muestra los perfiles habilitados para el usuario. 4. El usuario selecciona el perfil de Alumno. 5. La App muestra las opciones para la evaluación al docente de la materia que acabó de recibir clases. 6. El usuario selecciona el estado emocional que más se apegue a como se siente luego de haber recibido la clase. 7. La App valida que todos los datos cumplan con las reglas del negocio establecidas. 8. La App muestra un mensaje que indica que la información ha sido ingresada correctamente.
Excepciones	<p>Información incompleta</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un mensaje indicando la información que falta por completar, regresa al paso 6. <p>Información duplicada</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un error indicando que la evaluación ya ha sido registrada anteriormente, regresa al paso 4.
Pre condición:	El usuario debe estar registrado en el sistema y debe tener asignado el perfil de alumno.
Post condición:	El sistema permite el registro de evaluación a los docentes.

1.19.12 *Diagrama de casos de uso para la consulta de evaluaciones*

Figura 33

Diagrama de casos de uso para la consulta de evaluaciones



Nota. Se describe el diagrama de casos de uso para la consulta de evaluaciones históricas realizadas a los docentes por parte de los estudiantes. Elaborado por: El autor.

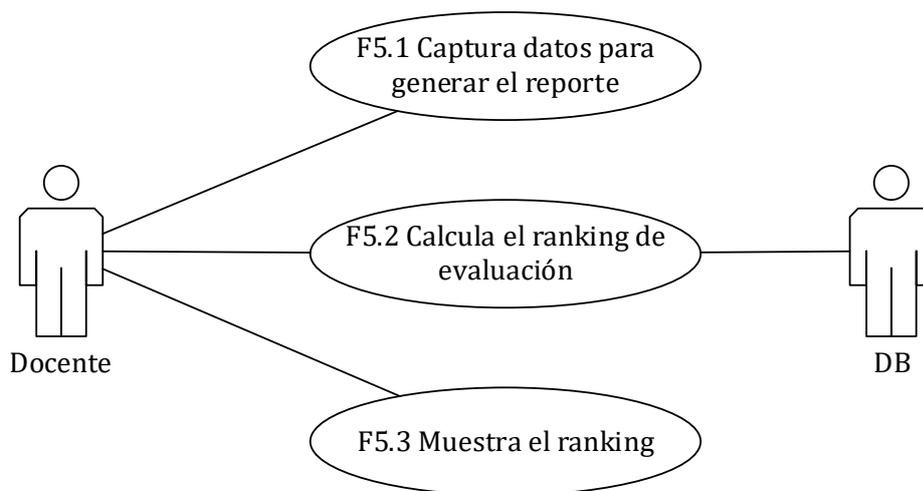
Caso de uso:	Consulta de evaluaciones
Descripción	Este caso de uso permite al estudiante ingresar a consultar las evaluaciones que ha realizado durante un período de tiempo.
Actores:	Estudiante
Actividades:	<ol style="list-style-type: none"> 1. El usuario ingresa a la App. 2. El usuario ingresa sus credenciales de acceso. 3. La App muestra los perfiles habilitados para el usuario. 4. El usuario selecciona el perfil de Alumno. 5. La App muestra las opciones para la consulta de evaluaciones a los docentes el alumno ha realizado. 6. El usuario registra los criterios de búsqueda de información por los que requiere consultar. 7. La App valida que todos los datos cumplan con las reglas del negocio establecidas.

8. La App muestra el listado de evaluaciones que cumplan con los criterios ingresados por el estudiante.
Excepciones
Información incompleta
1. Se muestra en pantalla un mensaje indicando la información que falta por completar, regresa al paso 6.
Pre condición:
El usuario debe estar registrado en el sistema y debe tener asignado el perfil de estudiante.
Post condición:
El sistema permite consultar las evaluaciones históricas registradas.

1.19.13 Diagrama de casos de uso para el reporte de evaluación

Figura 34

Diagrama de casos de uso para el reporte de evaluación



Nota. Se describe el diagrama de casos de uso para la generación del reporte de la evaluación docente (ranking). Elaborado por: El autor.

Caso de uso:	Revisión de evaluaciones
Descripción	Este caso de uso permite a los docentes consultar su reporte de evaluación por materia, por paralelo y a nivel global.
Actores:	Docente

<p>Actividades:</p> <ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación. 2. El usuario ingresa sus credenciales de acceso. 3. La App muestra los perfiles habilitados para el usuario. 4. El usuario selecciona el perfil de Docente. 5. La App muestra las opciones para la consulta del reporte de evaluaciones. 6. La App en base a la información ingresada por sus alumnos realiza los cálculos establecidos y determinar el puntaje de evaluación, generando un ranking. 7. La App muestra los resultados de la evaluación.
<p>Excepciones</p> <p>Docente sin evaluaciones de sus alumnos</p> <ol style="list-style-type: none"> 1. Se muestra en pantalla un mensaje indicando que hasta el momento sus alumnos no han registrado sus evaluaciones, regresa al paso 6.
<p>Pre condición:</p> <p>El usuario debe estar registrado en el sistema y debe tener asignado el perfil de docente.</p>
<p>Post condición:</p> <p>El sistema permite consultar el reporte de evaluaciones.</p>

1.20 IMPLEMENTACIÓN

En el presente apartado se detallará la implementación realizada los sprints identificados para el desarrollo del proyecto como se muestra en la Figura 41, mismos que en total fueron cuantificados para para 17 semanas, considerando que cada sprint tiene su propia planificación de actividades de story points.

1.20.1 Desarrollo font-end

En cuanto al desarrollo de la aplicación móvil se da a partir de las Historias de Usuario, está compuesta por módulos funcionales al final de cada sprint, cada codificación plasmada está desarrollada bajo el patrón de diseño MVC.

Estándares de codificación

- **Variables.** – las variables declaradas se encuentran relacionadas al valor y son escritas en letras minúsculas, con excepción de la primera.
- **Métodos.** – los métodos son denominados de acuerdo al proceso a ejecutar y son escritas en letras minúsculas. Sin embargo, si el método es extenso es recomendable aplicar cada primera letra en mayúscula.
- **Clases.** – la primera letra se escribe en mayúscula, si se compone de más palabras se recomienda aplicar cada primera letra en mayúscula.
- **Paquetes.** – generalmente los paquetes están escritos en minúsculas.

Para ello, se procede a preparar el ambiente de trabajo, instalando las siguientes librerías las cuales son esenciales para el funcionamiento del mismo. A continuación, se procede a explicar breves fragmentos de codificación desarrollados. Para del desarrollo del controlador de la vista, luego, se crea diferentes carpetas para aplicar el patrón de diseño MVC, en las siguientes líneas de código refleja CSS incorporados para la vista del aplicativo móvil, tales como: `go_router`, `hixe`, `json_path`, entre otros, ver figura 35.

Figura 35

Líneas de fragmento de código instalación librerías.

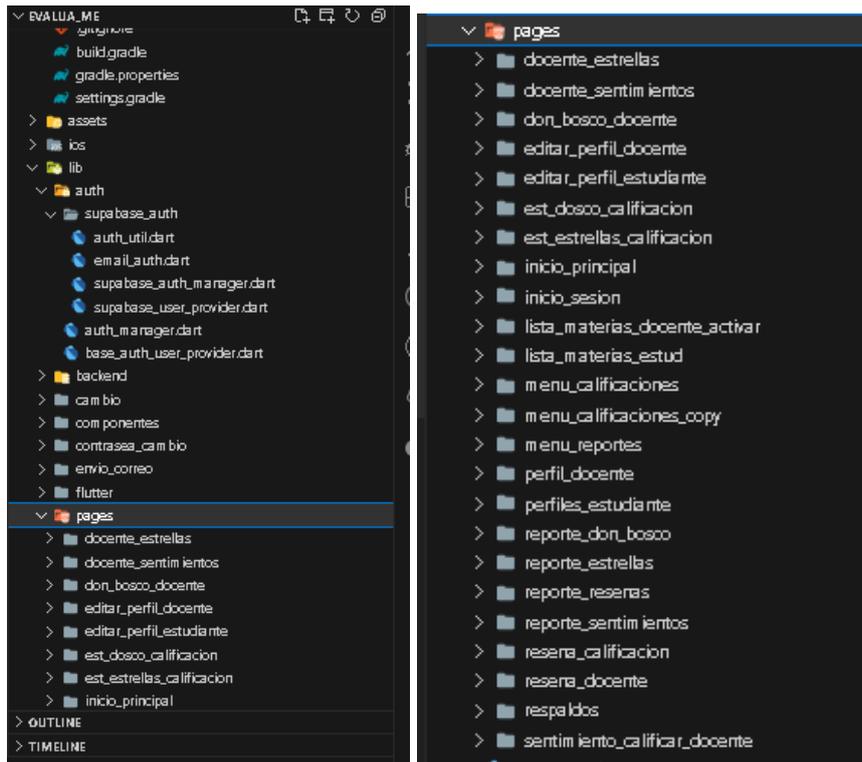
```
dependencies:
  go_router: 7.1.1
  google_fonts: 4.0.3
  hive: 2.2.3
  intl: 0.18.0
  json_path: 0.4.1
  mime_type: 1.0.0
  page_transition: 2.0.4
  path_provider: 2.0.14
  path_provider_android: 2.0.25
  path_provider_foundation: 2.2.2
  path_provider_platform_interface: 2.0.6
  pin_code_fields: 7.4.0
  provider: 6.0.4
  share_plus: 6.3.1
  shared_preferences: 2.0.15
  shared_preferences_android: 2.1.0
  shared_preferences_ios: 2.1.1
  shared_preferences_platform_interface: 2.2.0
  shared_preferences_web: 2.1.0
  sign_in_with_apple: 4.3.0
  sign_in_with_apple_platform_interface: 1.0.0
  sign_in_with_apple_web: 1.0.1
  smooth_page_indicator: 1.0.1
  sqflite: 2.2.6
  supabase: 1.6.3
  supabase_flutter: 1.7.0
  timeago: 3.2.2
  url_launcher: 6.1.10
  url_launcher_android: 6.0.27
  url_launcher_ios: 6.1.4
  url_launcher_platform_interface: 2.1.2
  webview_flutter: 4.0.5
  webview_flutter_android: 3.3.1
  webview_flutter_platform_interface: 2.0.2
  webview_flutter_wkwebview: 3.1.1
```

Una vez instalado las librerías mencionas, se presenta la siguiente estructura y organización

MVC del proyecto, ver siguiente figura 36, 37:

Figura 36

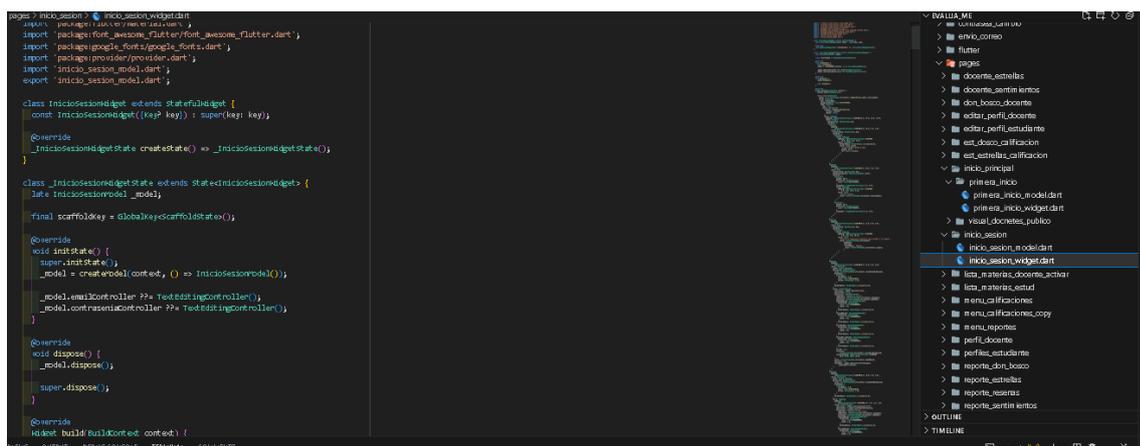
Organización lógica MVC proyecto



Una vez, levantado el proyecto se procede a desarrollar el FRONT-END del aplicativo móvil, para lo cual, se utilizan lenguajes óptimos para el desarrollo intuitivo, se importan paquetes, se declara métodos, ver figura 37.

Figura 37

Organización lógica MVC proyecto



```

pages > inicio_sesion > inicio_sesion_model.dart
import '/backend/api_requests/api_calls.dart';
import '/componentes/contras_cambio/contras_cambio_widget.dart';
import '/componentes/email_confirma/email_confirma_widget.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';

class InicioSesionModel extends FlutterFlowModel {
  final umfocusNode = FocusNode();
  TextEditingController? emailController;
  String? Function(BuildContext, String)? emailControllerValidator;
  TextEditingController? contraseniaController;
  late bool contraseniaVisibilidad;
  String? Function(BuildContext, String)? contraseniaControllerValidator;
  ApiCallResponse? apiResult4ah;
  ApiCallResponse? apiStudent;

  void initState(BuildContext context) {
    contraseniaVisibilidad = false;
  }

  void dispose() {
    umfocusNode.dispose();
    emailController?.dispose();
    contraseniaController?.dispose();
  }

  Future email(BuildContext context) async {}
}

```

Posteriormente, se procede a desarrollar el requerimiento funcional de evaluación al docente, para lo cual se importan paquetes, métodos y funciones.

Figura 38

Fragmentos de líneas de código calificación con estrella

```

pages > est_estrellas_calificacion > est_estrellas_calificacion_model.dart
import '/backend/api_requests/api_calls.dart';
import '/backend/supabase/supabase.dart';
import 'package:flutter/material.dart';
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';

class EstEstrellasCalificacionModel extends FlutterFlowModel {
  final umfocusNode = FocusNode();
  double? ratingBarValue;
  ApiCallResponse? apiResultClass;

  void initState(BuildContext context) {}

  void dispose() {
    umfocusNode.dispose();
  }
}

```

```

pages > est_estrellas_calificacion > est_estrellas_calificacion_widget.dart
import '/backend/api_requests/api_calls.dart';
import '/backend/supabase/supabase.dart';
import 'package:flutter/material.dart';
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';
import 'est_estrellas_calificacion_model.dart';
export 'est_estrellas_calificacion_model.dart';

class EstEstrellasCalificacionWidget extends StatefulWidget {
  const EstEstrellasCalificacionWidget({
    Key? key,
    this.idSubject,
    this.idUser,
    this.email,
  }) : super(key: key);

  final int? idSubject;
  final String? idUser;
  final String? email;

  @override
  _EstEstrellasCalificacionWidgetState createState() =>
    _EstEstrellasCalificacionWidgetState();
}

class _EstEstrellasCalificacionWidgetState
  extends State<EstEstrellasCalificacionWidget> {
  late EstEstrellasCalificacionModel _model;

  final scaffoldKey = GlobalKey<ScaffoldState>();

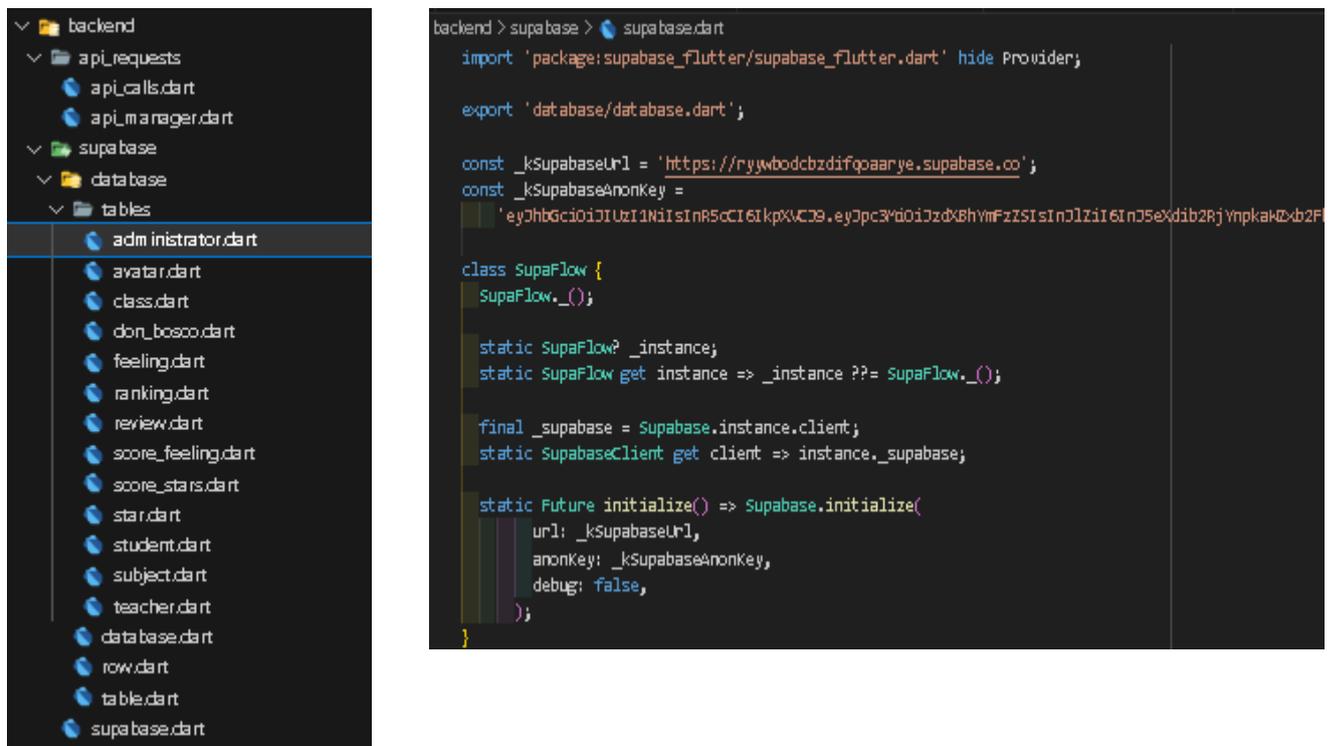
  @override
  void initState() {
    super.initState();
    _model = createModel(context, () => EstEstrellasCalificacionModel());
  }
}

```

Seguidamente En cuanto al desarrollo del controlador se encuentra constituido por aquellas tablas especificadas en el diagrama de clase, como: adm inistrator.dart, avatar.dart, class.dart, don_bosco.dart, reviw.dart, feeling.dart, student.dart, teacher.dart, etc.. Cada archivo de tipo dart utilizado está compuesto por los siguientes fragmentos de líneas de código, los cuales permiten realizar la gestión de información, es decir aquí ya posee la lógica de negocio

Figura 39

Fragmentos de línea de código para el desarrollo del back-end



Finalmente, la base de datos que pertenece al modelo se procede a aplicar un llamado a la base de datos supabase. A continuación, ver las siguientes líneas de código para establecer conexión con la base de datos.

Figura 40

Diagrama de casos de uso para el reporte de evaluación

```
import 'package:supabase_flutter/supabase_flutter.dart' hide Provider;
export 'database/database.dart';

const _kSupabaseUrl = 'https://ryywbodcbzdfqoaanye.supabase.co';
const _kSupabaseAnonKey =
  'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJ1ZiI6Ij5eXddib2RjYnpkaWZxb2Fhcn11Iiwicm9sZSI6ImFub24iLCJ0eXAiOiJKV1QiLCJkaXYiOiJkZXBlbWVudC99';

class SupaFlow {
  SupaFlow._();

  static SupaFlow? _instance;
  static SupaFlow get instance => _instance ??= SupaFlow._();

  final _supabase = Supabase.instance.client;
  static SupabaseClient get client => instance._supabase;

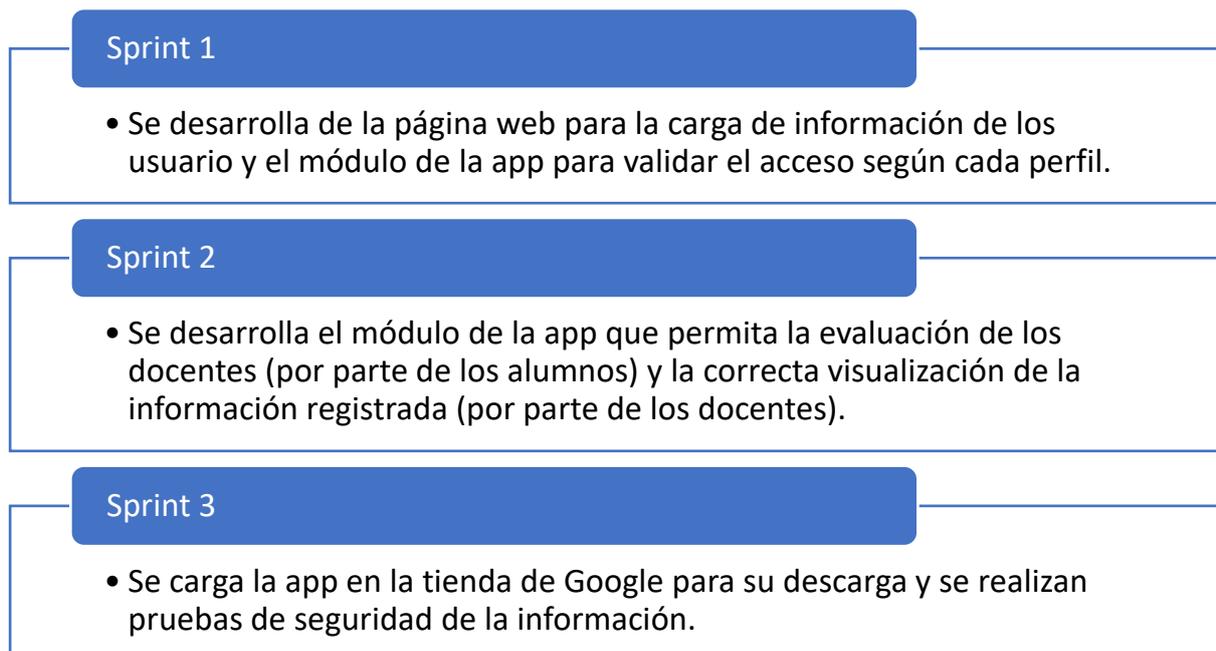
  static Future initialize() => Supabase.initialize(
    url: _kSupabaseUrl,
    anonKey: _kSupabaseAnonKey,
    debug: false,
  );
}
```

```
sqlite: 2.2.6
supabase: 1.6.3
supabase_flutter: 1.7.0
```

A continuación, se procede a describir cada uno de los sprint que se compone para el desarrollo del aplicativo móvil.

Figura 41

Descripción de los sprints



Nota. Se describen brevemente lo que se desarrollará en cada sprint. Elaborado por: El autor.

1.20.2 Sprint 1

El sprint 1 fue planificado para cinco semanas (25 días laborables) en el cual acorde a lo descrito en la Tabla 13 se desarrollaron las siguientes funcionalidades para la App.

Tabla 14

Actividades identificadas para ejecutar el sprint 1

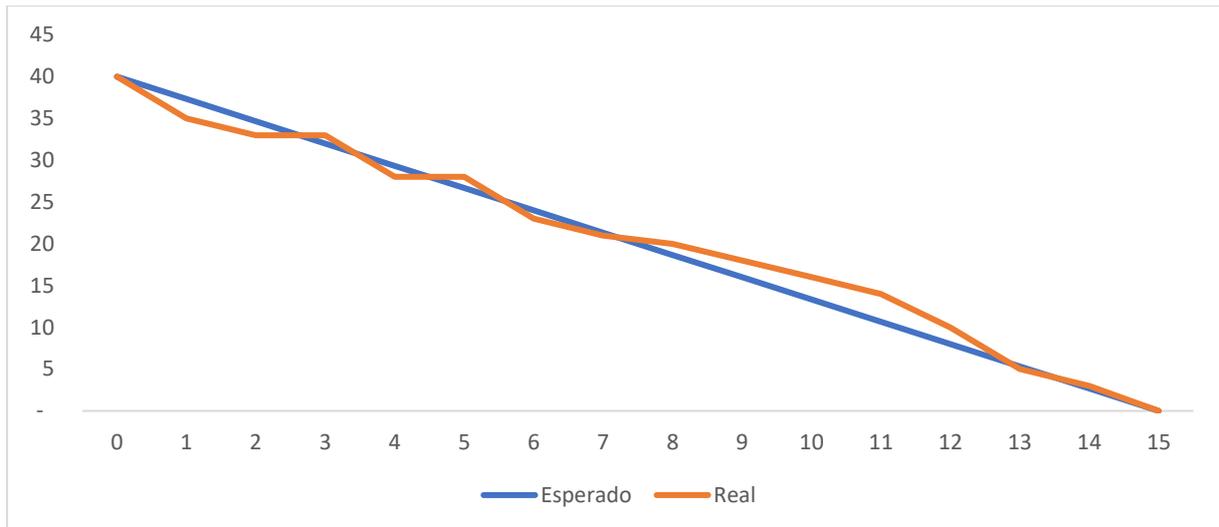
Actividades	Story Points
Diseño de la interfaz principal (menú de navegación, logo, etc.).	5
Diseño de la base de datos	5
MÓDULO LOGIN: Implementación de la interfaz para el ingreso de usuarios.	5
MÓDULO PARAMETRIZACIÓN: Implementación de la interfaz principales.	5
MÓDULO PARAMETRIZACIÓN: Implementación de la interfaz para el registro de roles de usuarios.	7
MÓDULO PARAMETRIZACIÓN: Ingreso de información de usuario y asignación de roles.	8
MÓDULO LOGIN: Asignación de permisos acorde a los roles establecidos.	5

Nota. Se describen las actividades y el número de story point a desarrollar para ejecutar el primer sprint. Elaborado por: El autor.

Para el sprint 1 se identificaron 40 story point, cuya ejecución se realizó en el tiempo previsto superando los inconvenientes y retrasos propios de un proyecto de este tipo.

Figura 42

BurnDown Chart del sprint 1



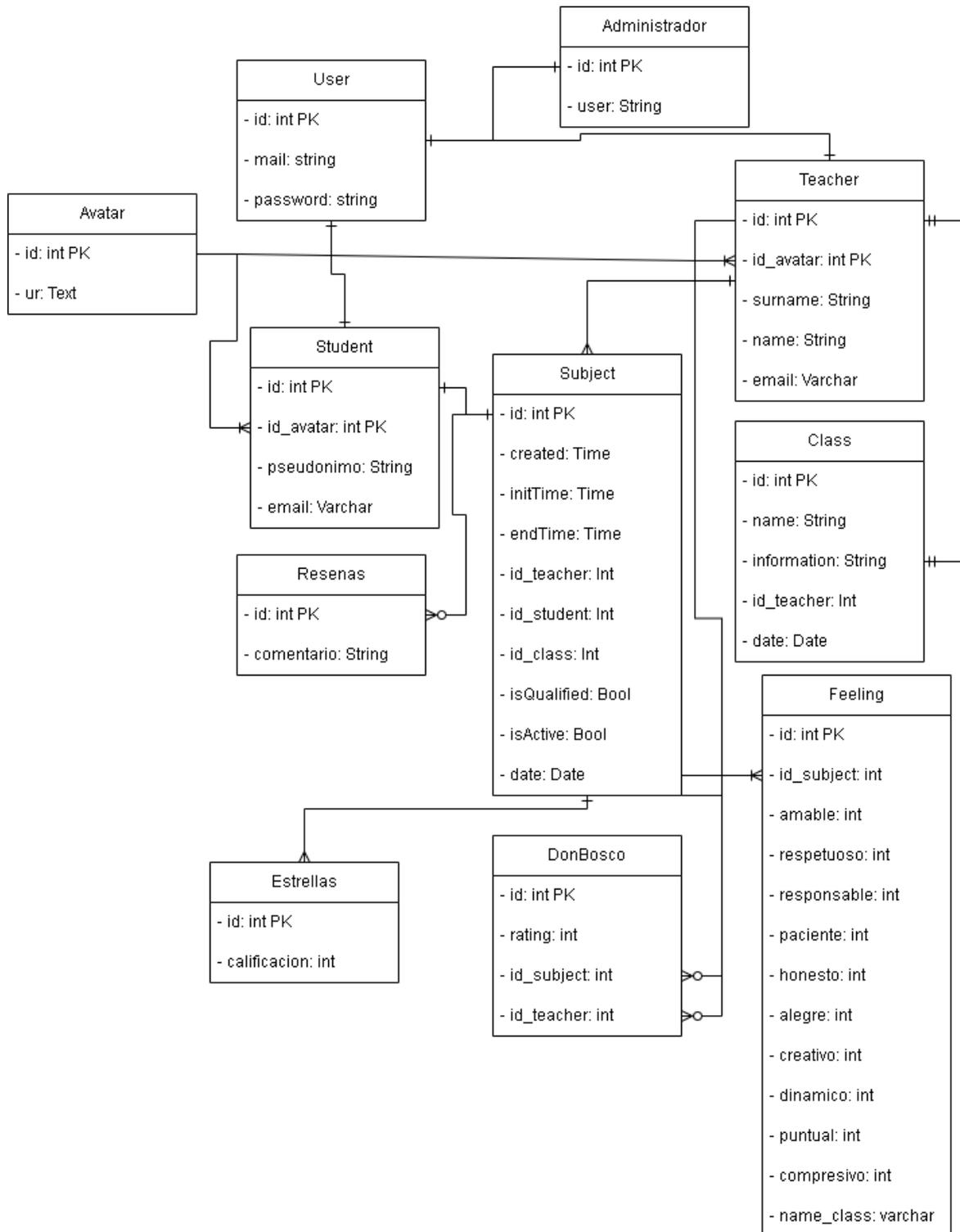
Nota. Se describe la manera en que se cumplieron lo story points del primer sprint en los 15 días de ejecución. Elaborado por: El autor.

Para garantizar el correcto funcionamiento del aplicativo móvil garantizando la integridad de la información, se diseñó la base de datos utilizando el modelo entidad-relación (MER) cuyo diagrama se presenta en la Figura 43.

Figura 43

Modelo entidad relación

Name	Description	Rows (Estimated)	Size (Estimated)	Realtime Enabled
Administrator	Mobile app manager	1	32 kB	6 columns
Avatar	Avatar image data	7	32 kB	3 columns
Class	Data for class	8	88 kB	6 columns
DonBosco	Data for ranking Don bosco	1	24 kB	6 columns
Feeling	Data for feeling	1	24 kB	16 columns
Ranking	Data for ranking teacher	2	48 kB	4 columns
Review	Data for review	1	32 kB	6 columns
ScoreFeeling	Data for ScoreFeeling	8	8192 bytes	16 columns
ScoreStars	Data for ScoreStars	5	48 kB	6 columns
Star	Star rating for subject	1	24 kB	7 columns
Student	Data for student	80	104 kB	6 columns
Subject	Data for subject	181	112 kB	10 columns
Teacher	Data for teacher	5	48 kB	7 columns



Nota. Muestra el MER de la base de datos diseñada. Elaborado por: El autor.

Considerando el alto volumen de información para el registro de alumnos y docentes se desarrolló una página web a través de la cual se permitirá la carga de un archivo de Excel con

toda la información necesaria de los usuarios que estarán habilitados para utilizar el aplicativo móvil.

En el aplicativo móvil acorde a la Figura 44 se presenta la pantalla que se le desplegará al usuario al momento de ingresar a la aplicación, en la cual deberá seleccionar un avatar, ingresar un pseudónimo y la contraseña que permita el acceso a la aplicación móvil.

Figura 44

Ingreso al sistema de información del lado del alumno

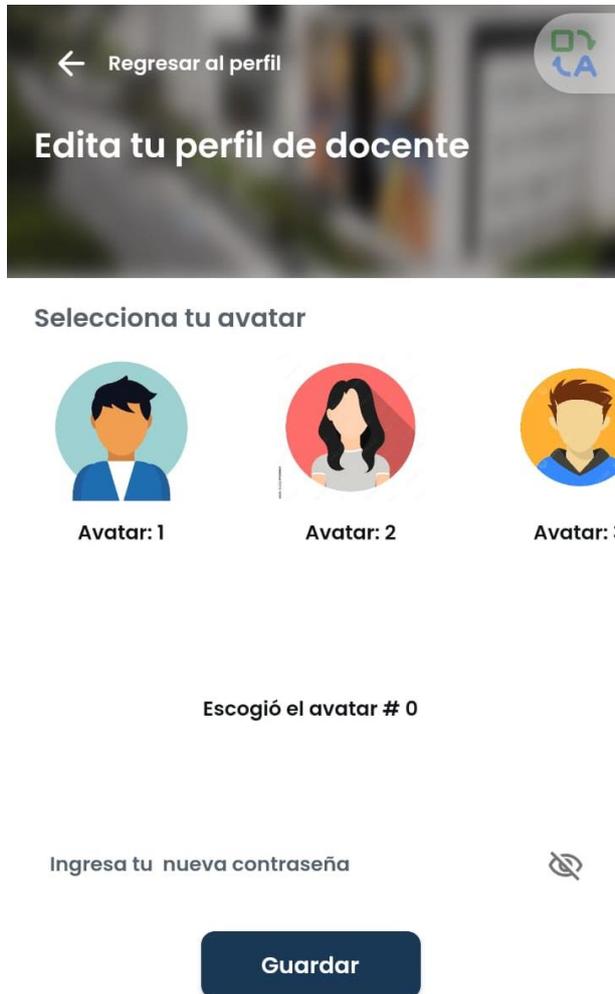


Nota. Se describe la pantalla para ingresar a la App del lado del alumno. Elaborado por: El autor.

Para el caso de los docentes, acorde a la Figura 45 se debe seleccionar el avatar e ingresar la contraseña que permita el acceso a la aplicación móvil.

Figura 45

Ingreso al sistema de información del lado del docente



Nota. Se describe la pantalla para ingresar a la App del lado del docente. Elaborado por: El autor.

1.20.3 Sprint 2

El sprint 2 fue planificado para ocho semanas (40 días laborables) en el cual acorde lo descrito en la Tabla 15 se desarrollaron las siguientes funcionalidades para la App.

Tabla 15

Actividades identificadas para ejecutar el sprint 2

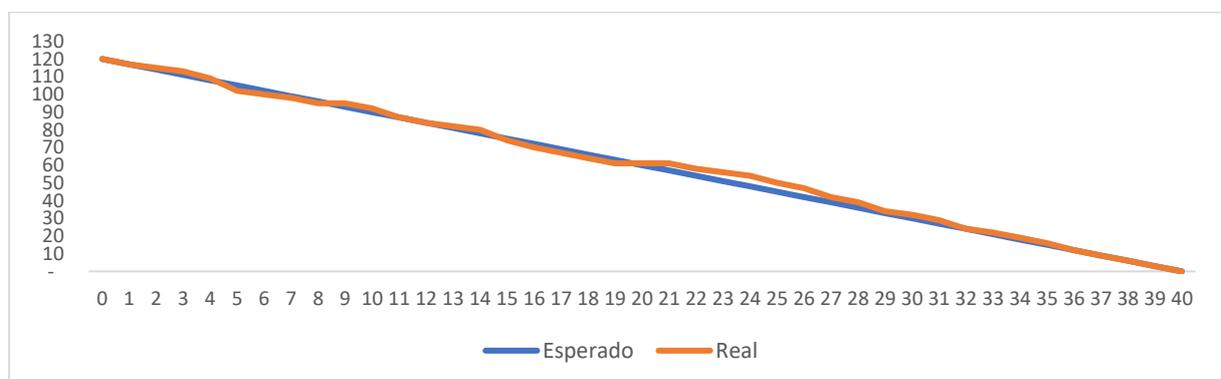
Actividades	Story Points
Diseño de la interfaz (menú de navegación, logo, etc.).	5
Identificación que los adjetivos que describen al buen docente	5
MÓDULO EVALUACIÓN: parametrización del número de estrellas para evaluar.	5
MÓDULO EVALUACIÓN: asignación de alumnos a cada docente para su evaluación.	10
MÓDULO EVALUACIÓN: registro de la evaluación.	25
MÓDULO EVALUACIÓN: categoría estrella.	25
MÓDULO EVALUACIÓN: categoría insignias.	25
MÓDULO EVALUACIÓN: categoría reseñas.	25

Nota. Se describen las actividades y el número de story point a desarrollar para ejecutar el segundo sprint. Elaborado por: El autor.

Para el sprint 2 se identificaron 120 story point, cuya ejecución se realizó en el tiempo previsto superando los inconvenientes y retrasos propios de un proyecto de este tipo.

Figura 46

BurnDown Chart del sprint 2



Nota. Se describe la manera en que se cumplieron lo story points del segundo sprint en los 40 días de ejecución. Elaborado por: El autor.

Para identificar los adjetivos que permitan calificar a los docentes, en el mes de octubre de 2020 se realizó una encuesta a 30 alumnos y ex alumnos de Ingeniería de Ciencias de la Computación a través del denominado “Formulario del Buen Docente” mismo que estuvo compuesto por 7 preguntas. El llenado de la encuesta fue mediante Google Forms, la Tabla 16 muestra el resultado de los 10 adjetivos más frecuentes con los que los alumnos calificarían a sus docentes.

Tabla 16

Resultado de la encuesta del buen docente

Nro.	Adjetivo	Total
1	Amable	12
2	Respetuoso	12
3	Responsable	10
4	Paciente	8
5	Honesto	7
6	Alegre	5
7	Creativo	5
8	Dinámico	4
9	Puntual	4
10	Comprensivo	3

Nota. Se describen los 10 adjetivos resultado de la encuesta. Elaborado por: El autor.

Acorde a lo descrito en la Figura 47, al alumno se le enlistarán todas las materias que tienen pendientes por calificar con la descripción del horario y fecha de la misma.

Figura 47

Listado de materias a evaluar



Nota. Se describe la manera en que se muestran las materias que se encuentran pendientes de evaluar. Elaborado por: El autor.

La evaluación a los docentes debe ser realizada a través de 4 categorías que se describen en la Figura 48, para lo cual se debe tener en cuenta las siguientes consideraciones de cada categoría:

- Estrellas: esta categoría permite calificar la calidad de la clase asistida con un mecanismo de entre 1 (la más baja) y 5 (la más alta) estrellas.
- Ranking Don Bosco: esta categoría, permite calificar la pasión que muestra el docente al impartir sus clases en el aula, cuenta con un mecanismo de 1 (la más baja) a 4 (la más alta).
- Insignias: esta categoría, permite que los estudiantes describan como si sintieron durante la clase.
- Reseña: permite enviar un comentario corto sobre la clase.

Figura 48

Listado de categorías que forman parte de la evaluación

← Regresar a materias

Categorías para calificar

4 Categorías

Selecciona

- 1 Estrellas**
Califica la calidad de la clase asistida.
- 2 Ranking Don Bosco**
Califica la pasión de enseñanza del docente.
- 3 Insignias**
Describe como te sentiste en esta clase.
- 4 Reseña**
Comentarios sobre la clase y/o del docente

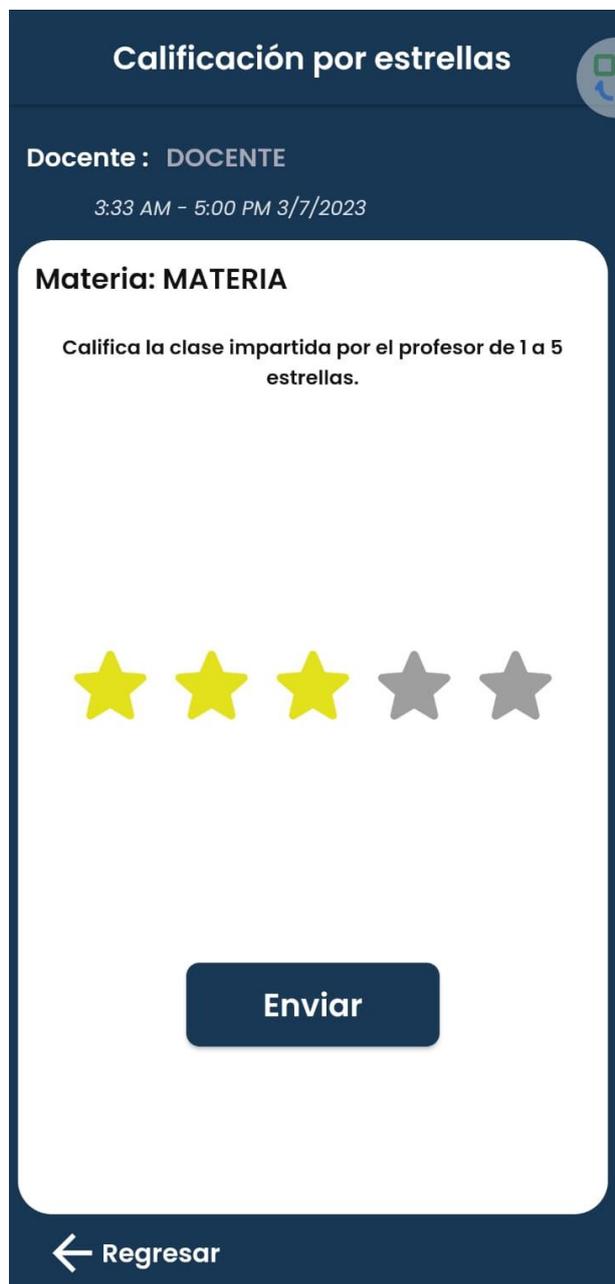
Enviar

Nota. Se describe las categorías que forman parte de la evaluación. Elaborado por: El autor.

Las Figura 49, Figura 50, Figura 51 y Figura 52 muestran la interfaz que cada alumno utilizará para la respectiva evaluación.

Figura 49

Categoría de estrellas



The image shows a mobile application interface for a star rating evaluation. The title is "Calificación por estrellas" (Star Rating). Below the title, it displays "Docente: DOCENTE" (Teacher: DOCENTE) and the time "3:33 AM - 5:00 PM 3/7/2023". The subject is "Materia: MATERIA" (Subject: MATERIA). The instruction is "Califica la clase impartida por el profesor de 1 a 5 estrellas." (Rate the class taught by the professor from 1 to 5 stars). There are five stars in a row; the first three are yellow and the last two are gray. At the bottom, there is a dark blue button labeled "Enviar" (Send) and a back arrow labeled "Regresar" (Return).

Nota. Se describe la manera de evaluar en la categoría estrellas. Elaborado por: El autor.

Figura 50

Categoría Ranking Don Bosco

Ranking Don Bosco

Docente : **DOCENTE PRIMERO**

Materia:: MATERIA 7/7/2023 12:32 PM



Pasión por la enseñanza



Usted está concediendo:
2.0 boscoins a su docente **DOCENTE PRIMERO**

Enviar

← Regresar

Nota. Se describe la manera de evaluar en la categoría Ranking Don Bosco. Elaborado por:

El autor.

Figura 51

Categoría Insignias



Selecciona el adjetivo para la materia: MATERIA
7/7/2023 12:33 PM

 Amable 0	 Respetuoso 0
 Responsable 0	 Paciente 0
 Honesto 0	 Alegre 0

Enviar

Nota. Se describe la manera de evaluar en la categoría insignias. Elaborado por: El autor.

Figura 52

Categoría Reseña

← **Reseña**

Materia: MATERIA
Docente: DOCENTE

11/7/2023 12:16

Deja tu comentario...

Comentar

Nota. Se describe la manera en que se muestran los resultados de la categoría Ranking Don Bosco. Elaborado por: El autor.

Así también el aplicativo móvil permite a los docentes revisar las evaluaciones recibidas por parte de sus alumnos; para lo cual, se debe considerar que se le mostrará una información

resumen y no el detalle de la evaluación de cada alumno. La Figura 53 describe la manera en la que el docente podrá acceder a cada categoría.

Figura 53

Listado de categorías para revisión de la evaluación



Nota. Se describe las categorías a través de las cuales el docente podrá revisar su evaluación.

Elaborado por: El autor.

La Figura 54, Figura 55, Figura 56 y Figura 57 muestran la interfaz que cada docente utilizará para revisar los resultados de su evaluación por cada materia.

Figura 54

Categoría de estrellas



The screenshot shows a web interface with a header 'Estrellas' and a sub-header 'DOCENTE PRIMERO'. Below this is a table with three columns: 'Materia', 'Promedio', and '# Calificaciones'. The table contains one row with the values 'MATERIA', '4.0', and '1'. A 'Regresar' button with a left-pointing arrow is located at the bottom left of the interface.

DOCENTE PRIMERO		
Materia	Promedio	# Calificaciones
MATERIA	4.0	1

← Regresar

Nota. Se describe la manera en que se muestran los resultados de la categoría estrella.

Elaborado por: El autor.

Figura 55

Categoría insignias

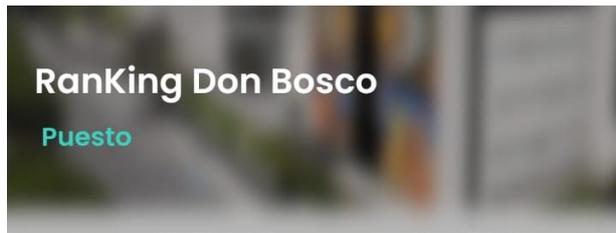


Nota. Se describe la manera en que se muestran los resultados de la categoría insignia.

Elaborado por: El autor.

Figura 56

Categoría Ranking Don Bosco



Gana el que tenga más boscomonedas

Pasión por la enseñanza

Puesto	Nombre	Boscoins
1	DOCENTE PRIMERO	3

Nota. Se describe la manera en que se muestran los resultados de la categoría Ranking Don Bosco. Elaborado por: El autor.

Figura 57

Categoría Reseña



Nota. Se describe la manera en que se muestran las reseñas ingresadas por los alumnos. Elaborado por: El autor.

1.20.4 Sprint 3

El sprint 3 fue planificado para cuatro semanas (20 días laborables) en el cual se desarrollaron las siguientes funcionalidades para la App.

Tabla 17

Actividades identificadas para ejecutar el sprint 3

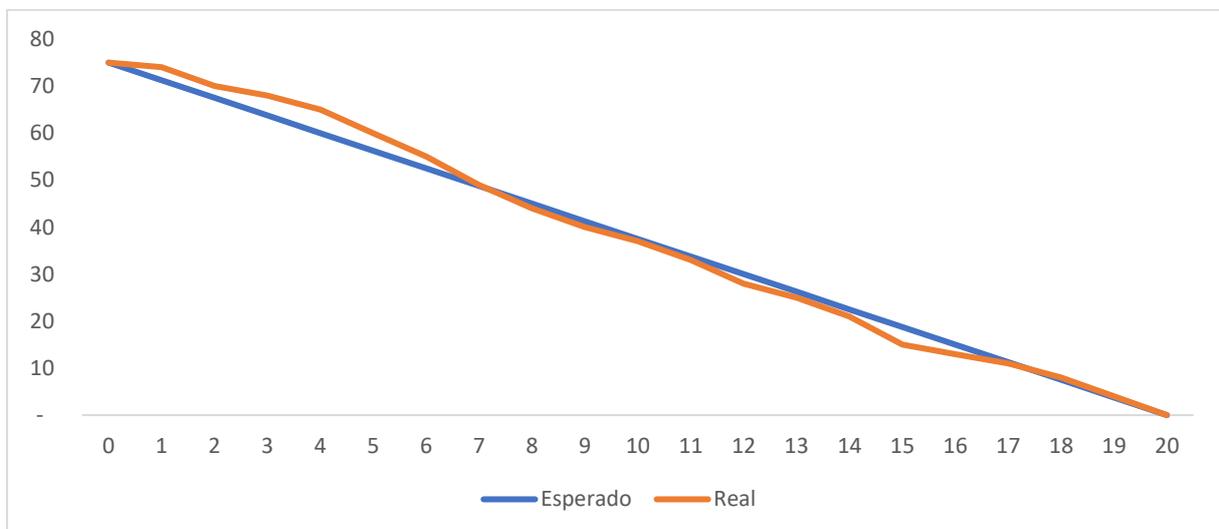
Actividades	Story Points
Diseño de la interfaz (menú de navegación, logo, etc.).	5
CARGA EN PLAY STORE: permitir la instalación de la App para validar su correcto funcionamiento.	5
MÓDULO EVALUACIÓN: garantizar que no se desplieguen detalles individuales de las evaluaciones.	5
SEGURIDAD DEL SISTEMA: garantizar que el sistema restrinja accesos no autorizados.	30

Nota. Se describen las actividades y el número de story point a desarrollar para ejecutar el segundo sprint. Elaborado por: El autor.

Para el sprint 3 se identificaron 45 story points, cuya ejecución se realizó en el tiempo previsto superando los inconvenientes y retrasos propios de un proyecto de este tipo.

Figura 58

BurnDown Chart del sprint 3



Nota. Se describe la manera en que se cumplieron lo story points del tercer sprint en los 20 días de ejecución. Elaborado por: El autor.

Como parte del proceso de validación de la funcionalidad del aplicativo móvil dentro de la universidad se realizó una campaña para su descarga, instalación y pruebas de funcionamiento mediante el código QR descrito en la Figura 59 mismo que le direcciona hacia la tienda de Google como lo muestra la Figura 60.

Figura 59

Post para motivar a la descarga de la App



Nota. Se describe el post utilizado para la campaña de pruebas de funcionamiento del aplicativo móvil. Elaborado por: El autor.

Figura 60

Descarga de la App de Play Store



Nota. Se describe la manera en que se muestra la App en la tienda de Google para su instalación. Elaborado por: El autor.

CAPÍTULO IV

RESULTADOS

1.21 PRUEBAS DE FUNCIONAMIENTO

Esta fase permite comprobar el correcto funcionamiento de los módulos, reglas del negocio definidas, tiempos de respuesta, robustez de la base de datos y finalmente que el aplicativo móvil en su integralidad satisfaga con todas las necesidades del cliente que fueron identificadas en la fase de análisis. En esta fase a medida que se prueba el correcto funcionamiento del sistema con información de prueba, si se identifican errores se realizarán las respectivas correcciones y ajustes tanto en el código de programación o en el modelamiento de la base de datos.

Estas pruebas son una parte fundamental dentro del ciclo de vida del sistema ya que permiten constantemente probar su correcto funcionamiento. En este caso, las pruebas fueron realizadas con los denominados datos de prueba, mismos que para esta fase representan a un conjunto de datos seleccionados previamente garantizando que cuenten con:

- Datos típicos.
- Datos que permitan poner a prueba los límites del sistema.
- Datos que permitan medir el cumplimiento de todas las reglas del negocio.
- Datos con condiciones particulares que puedan provocar error en el funcionamiento.

Las pruebas realizadas permitieron corregir errores por omisión en la incorporación de reglas del negocio en la programación, fueron realizadas varias veces hasta comprobar su correcto funcionamiento y fueron de dos tipos:

- Unitarias: Se probó por separado el correcto funcionamiento de las validaciones de cada uno de los campos de ingreso de información.
- De integración: Una vez que las pruebas unitarias terminaron y permitieron corregir errores, se prueba el funcionamiento todos los módulos en su conjunto para comprobar su integralidad.

Para las pruebas de funcionamiento, se utilizó la denominada prueba de la caja negra misma que es bastante común por su simpleza y eficacia, ya que una vez definidos los datos de entrada se realizan pruebas comprobando que los datos de salida sean los mismos, lo que permite verificar el pleno funcionamiento de cada uno de los módulos.

1.21.1 Pruebas para la creación de usuarios

En la Tabla 18 se detallan todas las pruebas realizadas para verificar el óptimo funcionamiento del módulo para la creación de usuarios y los resultados obtenidos.

Tabla 18

Pruebas de funcionamiento para la creación de usuarios

Nro.	Caso de prueba	Resultado	Conclusión
1	Ingreso de la información de 15 usuarios.	Exitoso	Se comprobó el ingreso de los 15 registros en la base de datos.

Nro.	Caso de prueba	Resultado	Conclusión
2	Ingreso de la información de 5 usuarios, todos los registros con uno o más campos vacíos.	Exitoso	Se validó que la información se encuentre completa y se comprobó que en la base de datos no se hayan ingresado dichos registros.
3	Ingreso de la información de 5 usuarios, todos los registros en el campo de correo electrónico se puso texto.	Exitoso	Se validó que la información se encuentre completa y se comprobó que en la base de datos no se hayan ingresado dichos registros.

1.21.2 Pruebas del Login

En la Tabla 19 se detallan todas las pruebas realizadas para verificar el óptimo funcionamiento del módulo de login y los resultados obtenidos.

Tabla 19

Pruebas de funcionamiento para el login

Nro.	Caso de prueba	Resultado	Conclusión
1	Intento de ingreso al sistema con 5 credenciales verdaderas	Exitoso	Se comprobó que los usuarios con credenciales correctas ingresen al sistema.
2	Intento de ingreso al sistema con 5 credenciales incorrectas	Exitoso	Se validó que la información se encuentre completa y se comprobó que los usuarios con

Nro.	Caso de prueba	Resultado	Conclusión
			credenciales incorrectas no ingresen al sistema.
3	Intento de ingreso al sistema ingresando el nombre de usuario y dejando vacío el campo de la clave	Exitoso	Se validó que la información se encuentre completa y se comprobó que los usuarios con credenciales incorrectas no ingresen al sistema.

1.21.3 Pruebas de ingreso de información en los módulos de la evaluación

En la Tabla 20 se detallan todas las pruebas realizadas para verificar el óptimo funcionamiento del módulo para para el ingreso de evaluaciones en sus cuatro módulos; así también, se describen los resultados obtenidos.

Tabla 20

Resultados de las pruebas de ingreso de información en los módulos de la evaluación

Nro.	Caso de prueba	Resultado	Conclusión
1	Ingreso de 15 evaluaciones en el módulo de estrellas	Exitoso	Se comprobó el ingreso de los 15 registros en la base de datos.
2	Ingreso de 15 evaluaciones en el módulo ranking Don Bosco	Exitoso	Se comprobó el ingreso de los 15 registros en la base de datos.
3	Ingreso de 15 evaluaciones en el módulo insignias	Exitoso	Se comprobó el ingreso de los 15 registros en la base de datos.
4	Ingreso de 15 comentarios en el módulo reseñas	Exitoso	Se comprobó el ingreso de los 15 registros en la base de datos.

1.21.4 Pruebas de revisión de la información de las evaluaciones del lado del docente

En la Tabla 21 se detallan todas las pruebas realizadas para verificar el óptimo funcionamiento del módulo para revisión de los docentes de las evaluaciones recibidas; así también, se describen los resultados obtenidos.

Tabla 21

Resultados de las pruebas de visualización de los docentes de las evaluaciones recibidas

Nro.	Caso de prueba	Resultado	Conclusión
1	Comprobación que las evaluaciones ingresadas a través del módulo de estrellas se muestren únicamente a los docentes evaluados	Exitoso	Con las credenciales de los docentes se verificó que los resultados se muestren únicamente a los docentes evaluados.
2	Comprobación que las evaluaciones ingresadas a través del módulo de ranking Don Bosco se muestren únicamente a los docentes evaluados	Exitoso	Con las credenciales de los docentes se verificó que los resultados se muestren únicamente a los docentes evaluados.
3	Comprobación que las evaluaciones ingresadas a través del módulo de insignias se muestren únicamente a los docentes evaluados	Exitoso	Con las credenciales de los docentes se verificó que los resultados se muestren únicamente a los docentes evaluados.
4	Comprobación que los comentarios ingresados a través del módulo de reseñas se muestren únicamente a los docentes evaluados	Exitoso	Con las credenciales de los docentes se verificó que los resultados se muestren únicamente a los docentes evaluados.

Nro.	Caso de prueba	Resultado	Conclusión
5	Comprobación que el cálculo del resultado de las evaluaciones ingresadas a través del módulo de estrellas sea correcto.	Exitoso	Con los datos ingresados a la base de datos se realizaron cálculos manuales para comprobar que el valor desplegado en la aplicación es el correcto.
6	Comprobación que las evaluaciones ingresadas a través del módulo de ranking Don Bosco se muestren únicamente a los docentes evaluados	Exitoso	Con los datos ingresados a la base de datos se realizaron cálculos manuales para comprobar que el valor desplegado en la aplicación es el correcto.

1.22 PRUEBAS DE USABILIDAD.

Como parte del proceso de validación de la aplicación por los usuarios finales se conformó un grupo de 30 estudiantes que descargaron y utilizaron la aplicación móvil a quienes se les pidió llenar una encuesta conformada por 5 preguntas. Los resultados de la pregunta ¿Encuentras la interfaz de la aplicación intuitiva y fácil de usar?, acorde a lo descrito en la Figura 61, el 93% indicaron que las interfaces de la aplicación móvil son intuitivas y fáciles de usar.

La realización de las pruebas de usabilidad de la aplicación móvil utilizando el aplicativo web Maze.

Figura 61

Resultados de la pregunta relacionada las interfaces del aplicativo móvil

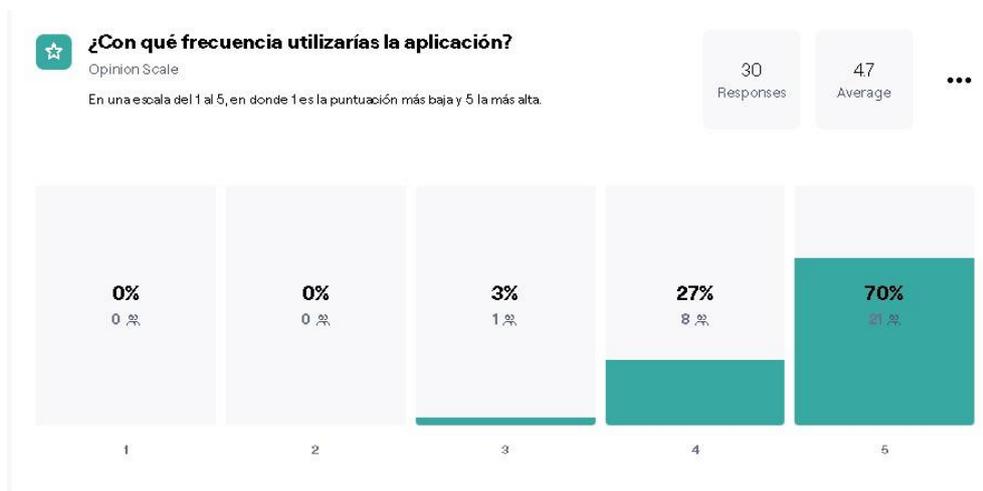


Nota. Se describen los resultados sobre las interfaces de la app. Elaborado por: El autor.

Los resultados de la pregunta “¿Con qué frecuencia usarías la aplicación?”, acorde a lo descrito en la Figura 62, el 97% de los participantes evaluaron a la aplicación móvil con las dos calificaciones más altas (4 y 5)

Figura 62

Resultados de la pregunta relacionada con la frecuencia de uso de la aplicación móvil



Nota. Se describen los resultados sobre la frecuencia de uso. Elaborado por: El autor.

Figura 63

Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil



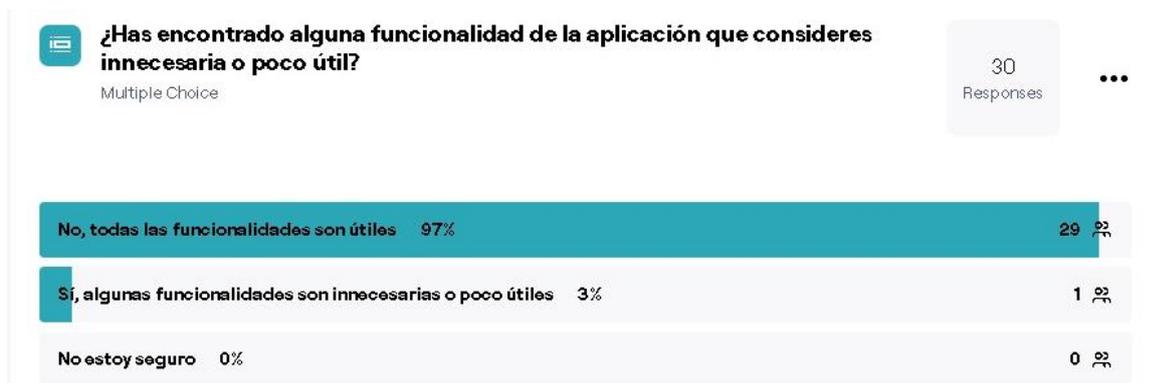
Nota. Se describen los resultados sobre la facilidad de navegación. Elaborado por: El autor.

Los resultados de la pregunta “¿Qué tan fácil te resulta navegar por la aplicación?”, acorde a lo descrito en la Figura 63, el 84% de los participantes evaluaron a la navegación en la aplicación móvil como muy fácil o fácil.

Los resultados de la pregunta “¿Qué tan fácil te resulta navegar por la aplicación?”, acorde a lo descrito en la Figura 64 el 84% de los participantes evaluaron a la navegación en la aplicación móvil como muy fácil o fácil.

Figura 64

Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil



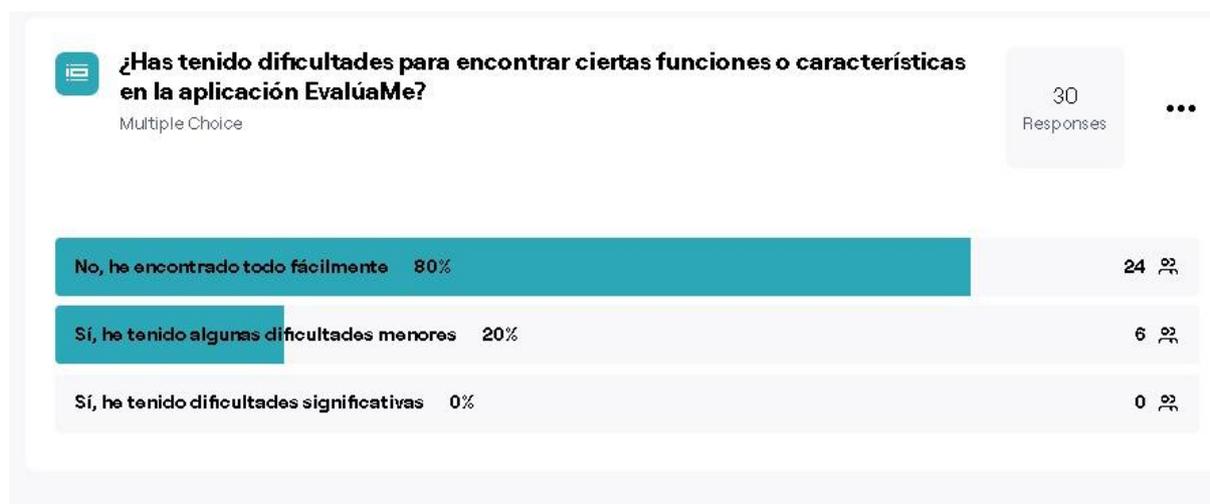
Nota. Se describen los resultados sobre la funcionalidad de la app. Elaborado por: El autor.

Los resultados de la pregunta “¿Has tenido dificultades para encontrar ciertas funciones o características en la aplicación EvalúaMe?”, acorde a lo descrito en la Figura 65, el 80% de los

participantes indicaron que todo lo han encontrado muy fácilmente y un 20% que han tenido dificultades menores.

Figura 65

Resultados de la pregunta relacionada con la facilidad de navegación en la aplicación móvil



Nota. Se describen los resultados de la pregunta 5. Elaborado por: El autor.

Finalmente, se hicieron pruebas de usabilidad de la aplicación móvil, acorde a lo descrito en la Figura 66 la totalidad de usuarios que probaron la pantalla para editar el perfil de usuario no tuvieron inconvenientes y pasaron exitosamente a la siguiente pantalla. En el caso de la pantalla para seleccionar el tipo de evaluación a realizar el 96% de los usuarios no tuvieron inconvenientes y pasaron exitosamente a la siguiente pantalla (ver Figura 67). En el caso de la pantalla para revisar los resultados de la evaluación del componente de estrellas, el 85% de los docentes que la probaron no tuvieron inconvenientes y pasaron exitosamente a la siguiente pantalla (ver Figura 68). Con todos los resultados mostrados se evidencia el correcto funcionamiento del aplicativo móvil y su facilidad de uso en las etapas de prueba.

Figura 66

Resultados del uso de la pantalla de perfil de usuario



Nota. Se describen los resultados sobre los resultados del uso de la interfaz de usuario para editar el perfil de usuario. Elaborado por: El autor.

Figura 67

Resultados del uso de la pantalla de los componentes de la evaluación



Nota. Se describen los resultados sobre los resultados del uso de la pantalla de los módulos de la evaluación. Elaborado por: El autor.

Figura 68

Resultados del uso de la pantalla para revisar los resultados de la calificación por estrellas



Nota. Se describen los resultados sobre los resultados del uso de la pantalla de los módulos de la evaluación. Elaborado por: El autor.

CONCLUSIONES

- La revisión bibliográfica realizada durante el tiempo de la investigación permitió reforzar y actualizar los conocimientos adquiridos durante la vida universitaria, garantizando una mejor aplicación de los mismos durante el desarrollo de la aplicación móvil.
- La metodología SCRUM para el desarrollo de la aplicación móvil permitió optimizar el tiempo de planificación, ejecución y comprobación durante el desarrollo de los diferentes sprints permitiendo solucionar los inconvenientes, ajustar los cronogramas y realizar cambios en el funcionamiento de la app de forma ágil.
- Uno de los ejes fundamentales para el óptimo funcionamiento del aplicativo móvil se centró en la identificación de aquellos módulos que centrados en las emociones generen mayor interés para que cada vez más alumnos de manera voluntaria realicen la evaluación de sus docentes.
- El desarrollo de la aplicación móvil fue pensado en todo momento para dar soluciones a los inconvenientes detectados en la evaluación a los docentes, lo que incluyó también diseñar un entorno amigable a través de la incorporación de las mejores prácticas de navegabilidad en la aplicación y el uso de los distintos widgets disponibles para Flutter.
- Para el desarrollo del aplicativo móvil se utilizaron las herramientas de software libre (Dart, Flutter, Supabase) mismas que se encuentran entre las más utilizadas por los desarrolladores de apps lo que permitió que tanto a nivel de base de datos como de front-end la aplicación posea las características necesarias para su óptimo funcionamiento en un ambiente real.
- Finalmente, durante la fase de pruebas de funcionamiento, tanto con registros ficticios como con información de histórica, se verificó el funcionamiento.

RECOMENDACIONES

- Al nivel de producción se debería implementar los aspectos de seguridad como confiabilidad, disponibilidad e integridad de la aplicación como indican las normas ISO 27001.
- El desarrollo del aplicativo móvil es una herramienta necesaria que brinda respuestas oportunas y confiables para mejorar la gestión del proceso de evaluación docente, por lo que, se recomienda el uso del aplicativo.
- El mayor beneficio del uso del aplicativo móvil estará dado por el uso que se le dé a la información que diariamente se generará, por lo que se recomienda que la información periódicamente sea analizada por docentes y directivos con el objetivo de realizar ajustes de ser necesarios o promover el intercambio de buenas prácticas del proceso de enseñanza aprendizaje ejecutados en el aula.
- El uso constante del aplicativo móvil demandará la implementación de mejoras en su funcionamiento, por lo que, se recomienda contar con el personal idóneo que permita garantizar su óptimo funcionamiento a través del tiempo.
- Finalmente, considerando los beneficios que genera aplicativo móvil para su óptimo funcionamiento se recomienda actualizar las diferentes normativas internas que regulen el proceso de evaluación docente considerando que el mismo será permanente.

REFERENCIAS

- Acosta, E., Álvarez, J. A., & Gordillo, A. (2019). Arquitecturas en n-Capas: Un Sistema Adaptivo. *Polibits*(34), 34-37. Obtenido de <https://www.redalyc.org/articulo.oa?id=402640447007>
- Apache Cassandra. (2022). *Fundamentos de Casandra*. Obtenido de Sitio web: https://cassandra.apache.org/_/cassandra-basics.html
- Apple. (2022). *Swift. Un poderoso lenguaje al alcance de todos para crear increíbles apps*. Obtenido de Sitio Web Apple: <https://www.apple.com/la/swift/>
- Asociación Iberoamericana de Educación Superior a Distancia. (2017). Recensiones/Reseñas. *RIED. Revista Iberoamericana de Educación a Distancia*, 20(2), 339-344. Obtenido de <https://www.redalyc.org/pdf/3314/331453132016.pdf>
- Castro, A., González, J., & Callejas, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21(33), 21-32. Retrieved from <https://www.redalyc.org/articulo.oa?id=413940772003>
- DART. (2022). *Resumen de DART*. Obtenido de Sitio Web: https://dart-dev.translate.google.com/overview?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc
- Delgado, E. (2019). Metodologías de desarrollo de software. ¿Cuál es el camino? *Revista de Arquitectura e Ingeniería*, 2(3). Obtenido de <https://www.redalyc.org/articulo.oa?id=193915935003>
- Filgueira, J. (2014). *Arquitectura de sistemas de información -Un enfoque real basado en experiencia-*. Galicia: Autoediciones TAGUS.
- Finanzas. (2021). *El 71% de las organizaciones a nivel mundial ya usa Metodologías Ágiles*. Obtenido de Sitio Web Finanzas: https://www.finanzas.com/empresas/el-71-de-las-organizaciones-a-nivel-mundial-ya-usa-metodologias-agiles_13757964_102.html

- Firestore. (2022). *Almacena y sincroniza datos en tiempo real*. Obtenido de Sitio Web Firestore: <https://firebase.google.com/products/realtime-database?hl=es-419#:~:text=Firestore%20Realtime%20Database%20es%20una,de%20app%20a%20e-scala%20global>.
- Flutter España. (2022). *Bienvenido a la página de la comunidad FlutterES*. Obtenido de Sitio Web Flutter España: <https://esflutter.dev/>
- Garcés, L., & Egas, L. (2018). 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>
- Garrido Sotomayor, S. (09 de diciembre de 2021). *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. Obtenido de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- Gestión Educativa. (2016). *Evaluación de la práctica docente por parte del alumnado: Cómo hacerla*. Obtenido de Sitio Web GestionEducativa: <https://gestioneducativa.educaweb.com/evaluacion-practica-docente-alumnado-como/>
- Jimenez, V., Tello, W., & Rios, J. (2014, Diciembre). Lenguajes de Patrones de Arquitectura de Software: Una Aproximación Al Estado del Arte. *Scientia Et Technica*, 19(4), 371-376. Retrieved from <https://www.redalyc.org/pdf/849/84933912003.pdf>
- Kotlin. (2022 a). *A modern programming language*. Obtenido de Sitio Web Kotlin: <https://kotlinlang.org/>
- Kotlin. (2022 b). *Sobre Kotlin*. Obtenido de Sitio Web Kotlin: <https://kotlin.es/sobre-kotlin/#:~:text=Kotlin%20es%20un%20lenguaje%20de,%C3%BAltimas%20versiones%20tambi%C3%A9n%20ejecutables%20nativos>.

- KYOCERA. (15 de Enero de 2020). *Los 6 principales tipos de sistemas de información*.
Obtenido de <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/the-cloud/los-6-principales-tipos-sistemas-informacion.html>
- Leyva, K., Alarcón, L., & Ortégón, L. (2019). Exploración del diseño y arquitectura web. Aplicación a páginas electrónicas del sector bancario desde la perspectiva del usuario. *Revista Escuela de Administración de Negocios*(80), 41-57. Obtenido de <https://www.redalyc.org/pdf/206/20645903004.pdf>
- Malave, K., & Beauperthuy, J. (2011). "Android" el sistema operativo de Google para dispositivos móviles. *Negotium*, 19(7), 79-96. Obtenido de <https://www.redalyc.org/articulo.oa?id=78219156004>
- MariaDB Foundation. (2023). *MariaDB Server: The open source relational database*. Obtenido de Sitio web: <https://mariadb.org/>
- MongoDB. (2021). *Reinventando la gestión de datos*. Obtenido de <https://www.mongodb.com/es>
- Montoya, M., & Espinoza, G. (05 de Mayo de 2017). *La responsabilidad social universitaria en Ecuador*. Obtenido de <https://revistas.uasb.edu.ec/index.php/eg/article/view/568/3108>
- Morales , J. (2001). *La evaluación en el área de educación*. Obtenido de <https://www.tdx.cat/bitstream/handle/10803/5036/jjma08de16.pdf.PDF>
- Moreno, T. (2018). La Evaluación Docente en la Universidad: Visiones de los Alumnos. *REICE. Revista Iberoamericana sobre Calidad*, 87-99. Obtenido de <https://www.redalyc.org/journal/551/55160077005/55160077005.pdf>
- Mozilla. (2022). *JavaScript*. Obtenido de Sitio Web Mozilla: <https://developer.mozilla.org/es/docs/Web/JavaScript>

- ORACLE. (2022). *Temas de base de datos*. Obtenido de <https://www.oracle.com/mx/database/what-is-database/>
- Ordóñez, D. (2021). Reduciendo la brecha de seguridad del IoT con una arquitectura de microservicios basada en TLS y OAuth2. *Ingenius*(25). doi:<https://doi.org/10.17163/ings.n25.2021.09>
- Parra, E. (2021). Metodología de desarrollo de software para objetos virtuales de aprendizaje - MESOVA. *Revista Virtual Universidad Católica del Norte*(34), 113-137. Obtenido de <https://www.redalyc.org/articulo.oa?id=194222473006>
- Payares, G. (2022). *¿Cuáles son las aplicaciones más usadas en todo el mundo?* Obtenido de Sitio Web Ecommercenews: <https://www.ecommercenews.pe/marketing-digital/2022/cuales-son-las-aplicaciones-mas-usadas-en-todo-el-mundo.html>
- PostgreSQL. (2023). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. Obtenido de Sitio web: <https://www.postgresql.org/>
- Revista Líderes. (2014). *En 20 años, la telefonía móvil superó las expectativas*. Obtenido de Sitio Wweb: <https://www.revistalideres.ec/lideres/20-anos-telefonía-movil-supero.html>
- Rodríguez, R., Vera, P., Martínez, M., & Dogliotti, M. (2020). Análisis de estándares para la web móvil. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 9(2), 1-20. Obtenido de <https://www.redalyc.org/journal/5122/512267931003/512267931003.pdf>
- Ros, I. (2015). *Un paseo por la evolución de los teléfonos móviles*. Obtenido de Sitio Web MuyComputer: <https://www.muycomputer.com/2015/07/28/evolucion-de-los-telefonos-moviles/>
- Soares, S. (2010). *The IBM Data Governance Unified Process*. MC Press.
- Souza. (2022). *UNESCO*. Obtenido de Docentes: <https://www.unesco.org/es>

- Supabase. (2023). *Construir en un fin de semana de semana Escala a millones*. Obtenido de Sitio web: <https://supabase.com/>
- Tokio School. (05 de noviembre de 2021). *Lenguaje de programación swift: características*. Obtenido de Sitio Web Tokio School: <https://www.tokioschool.com/noticias/lenguaje-programacion-swift-caracteristicas/>
- Universidad Politécnica Salesiana. (2022). *Organigramas UPS*. Obtenido de Sitio Web Universidad Politécnica Salesiana: <https://www.ups.edu.ec/organigrama>
- Urriola, K. (2013). *Sistema de evaluación del desempeño profesional docente aplicado en Chile*. Obtenido de Sitio web: Universitat de Barcelona: https://www.tesisenred.net/bitstream/handle/10803/131128/01.KMUL_1de2.pdf?sequence
nc
- White, T. (2009). *Hadoop: The Definitive Guide*. USA: O'Reilly.
- Zamora, S. (2021). *Kanban, ¿Qué es?* Obtenido de Sitio Web Scrum México: <https://scrum.mx/informate/kanban/que-es>
- Zubikarai, S. (2021). *Ventajas y Desventajas de JavaScript*. Obtenido de Sitio Web Freecodecamp: <https://www.freecodecamp.org/espanol/news/ventajas-y-desventajas-de-javascript/>