



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA GESTIÓN DE
PORTAFOLIOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA
CIVIL USANDO LA METODOLOGÍA SWIRL

CABRERA MIRANDA SILVIA EUGENIA
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA
GESTIÓN DE PORTAFOLIOS ACADÉMICOS DE LA FACULTAD
DE INGENIERÍA CIVIL USANDO LA METODOLOGÍA SWIRL

CABRERA MIRANDA SILVIA EUGENIA
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA GESTIÓN DE
PORTAFOLIOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA CIVIL
USANDO LA METODOLOGÍA SWIRL

CABRERA MIRANDA SILVIA EUGENIA
INGENIERA DE SISTEMAS

HONORES TAPIA JOOFRE ANTONIO

MACHALA, 27 DE ABRIL DE 2021

MACHALA
2021

Trabajo de titulación de Silvia Cabrera

INFORME DE ORIGINALIDAD

9%

INDICE DE SIMILITUD

6%

FUENTES DE INTERNET

0%

PUBLICACIONES

5%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Técnica de Machala Trabajo del estudiante	2%
2	Submitted to UNIV DE LAS AMERICAS Trabajo del estudiante	<1%
3	Submitted to INACAP Trabajo del estudiante	<1%
4	qdoc.tips Fuente de Internet	<1%
5	Submitted to Universidad Estatal a Distancia Trabajo del estudiante	<1%
6	Submitted to Universidad Sergio Arboleda Trabajo del estudiante	<1%
7	futur.upc.edu Fuente de Internet	<1%
8	Submitted to Universidad Alas Peruanas Trabajo del estudiante	<1%
9	Submitted to Universidad Peruana de Las Americas	<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, CABRERA MIRANDA SILVIA EUGENIA, en calidad de autora del siguiente trabajo escrito titulado IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA GESTIÓN DE PORTAFOLIOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA CIVIL USANDO LA METODOLOGÍA SWIRL, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

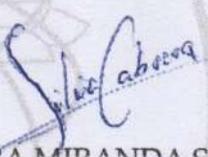
La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 27 de abril de 2021


CABRERA MIRANDA SILVIA EUGENIA
0706385200

DEDICATORIA

Dedico el presente trabajo investigativo a mis padres, que con sus enseñanzas han logrado orientarme correctamente e inspirarme a seguir mis sueños y cumplir esta meta en mi vida.

A mi pareja Jorge González por su amor y comprensión, alentandome en momentos difíciles con su apoyo y compañía incondicional.

Srta. Cabrera Miranda Silvia Eugenia

AGRADECIMIENTO

Por su apoyo y comprensión a lo largo de estos años de estudio, quiero brindar mi más grande agradecimiento a mis padres, ustedes siempre creyeron en mí, dándome ánimos y los recursos necesarios para seguir adelante con mi sueño.

Por su ayuda y cariño, quiero agradecer a mi pareja Jorge González, estuviste a mi lado en los momentos que más necesitaba un apoyo, alentándome a seguir cada día, muchas gracias amor.

A mi tutor Ing. Joofre Antonio Honores Tapia que con su mano amiga, supo guiarme y encaminarme en la realización de este trabajo de titulación.

Srta. Cabrera Miranda Silvia Eugenia

RESUMEN

Una de las tareas que debe llevar a cabo todo estudiante universitario es la elaboración del portafolio académico. Su importancia radica en que, al ser un método de aprendizaje, enseñanza y evaluación, permite al final de cada curso observar cuáles fueron las debilidades y fortalezas durante ese Período, dando como resultado un mejoramiento continuo del mismo.

Al tratarse de una compilación de todos los trabajos y un seguimiento diario de todas las clases impartidas por cada asignatura de un período académico, su proceso de elaboración se puede volver algo tedioso para el estudiante, además que existe un formato de presentación que se debe seguir y que no muchos alumnos cumplen en su totalidad al no respetar los lineamientos establecidos para su correcta presentación.

El uso de aplicaciones móviles se encuentra en gran ascenso, debido a que cada vez son más los usuarios que optan por utilizar aplicaciones en sus celulares con el fin de realizar sus actividades o simplemente distraerse de su vida cotidiana, volviéndose actualmente algo indispensable.

Gracias a esto, el desarrollo de aplicaciones como negocio ha incrementado en su producción, buscando satisfacer las necesidades de este público para así ayudar a mejorar su productividad y experiencia de usuario, en especial a aquellos que hacen uso de aplicaciones móviles con fines laborales o educativos.

Es por eso que, como resolución a esta problemática, se desarrolló una aplicación móvil con el fin de mejorar el proceso de elaboración del portafolio académico de los estudiantes, dando una solución rápida a los problemas de formatos y la pesada tarea de organización de documentos.

La aplicación móvil fue desarrollada principalmente bajo el framework multiplataforma Flutter, también se hizo uso de otras herramientas como, jQuery, NodeJS, Flask (back-end), la metodología SWIRL como métrica de desarrollo y finalmente como fuentes de bases de datos a PostgreSQL y MongoDB. Estas bases de datos se encuentran alojadas en un servidor propio bajo una dirección pública para tener acceso a esta información mediante el uso de APIs elaboradas para este proyecto.

Dentro de la aplicación móvil, si el usuario es un estudiante, podrá loguearse y matricularse en la o las asignaturas que curse durante ese Período académico, luego podrá acceder al formato del portafolio creado automáticamente para esa asignatura, donde los diarios metacognitivos se encontrarán organizados y listos para que el alumno escriba la información correspondiente en ellos.

El uso de notificaciones (mediante Firebase) se hará presente para recordar al estudiante que debe realizar el diario correspondiente a ese día. Una vez culminado el portafolio, este puede ser descargado en formato .rar para que este pueda ser enviado y utilizado.

En cuanto a si el usuario es un docente, podrá asignar una clave de matriculación a la asignatura impartida por él para que los alumnos puedan matricularse. El docente podrá descargar los portafolios culminados de todos los estudiantes matriculados a su asignatura.

La evaluación del prototipo desarrollado se basó en los estándares de calidad de la norma ISO/IEC9126 que trata acerca de la funcionalidad, eficiencia, compatibilidad de los sistemas y la evaluación de accesibilidad bajo el estándar WCAG 2.1.

Palabras clave: Aplicación móvil, Flutter, desarrollo multiplataforma, portafolio académico, SWIRL.

ABSTRACT

One of the tasks that every university student must carry out is the preparation of the academic portfolio. Its radical importance is that, being a method of learning, teaching and evaluation, it allows at the end of each course to observe results were the weaknesses and strengths during that Period, resulting in a continuous improvement of the same.

As it is a compilation of all the works and a daily monitoring of all the classes taught for each subject of an academic period, its preparation process can become somewhat tedious for the student, in addition to the fact that there is a presentation format that must be follow and that not many students fully comply by not respecting the guidelines established for their correct presentation.

The use of mobile applications is on the rise, because more and more users choose to use applications on their cell phones in order to carry out their activities or simply distract themselves from their daily life, making them now indispensable.

Thanks to this, the development of applications as a business has increased in its production, seeking to satisfy the needs of this audience in order to help improve their productivity and user experience, especially those who make use of mobile applications for work or educational purposes. .

That is why, as a resolution to this problem, a mobile application was developed in order to improve the process of preparing the students' academic portfolio, giving a quick solution to formatting problems and the heavy task of organizing documents.

The mobile application was developed mainly under the Flutter multiplatform framework, other tools were also used such as jQuery, NodeJS, Flask (back-end), the SWIRL methodology as a development metric and finally as database sources to PostgreSQL and MongoDB. These databases are hosted on their own server under a public address to access this information through the use of APIs developed for this project.

Within the mobile application, if the user is a student, they will be able to login and enroll in the subject (s) that they take during that academic Period, then they will be able to access the portfolio format automatically created for that subject, where the

metacognitive journals will be organized and ready for the student to write the corresponding information on them.

The use of notifications (through Firebase) will be present to remind the student that they must carry out the journal corresponding to that day. Once the portfolio is completed, it can be downloaded in .rar format so that it can be sent and used.

Regarding whether the user is a teacher, he can assign a registration key to the subject taught by him so that students can enroll. The teacher will be able to download the completed portfolios of all the students enrolled in their subject.

The evaluation of the developed prototype was based on the quality standards of the ISO / IEC9126 standard that deals with the functionality, efficiency, compatibility of the systems and the evaluation of accessibility under the WCAG 2.1 standard.

Keywords: Mobile application, Flutter, cross-platform development, academic portfolio, SWIRL.

INTRODUCCIÓN

El uso de aplicaciones móviles con el paso de los años ha ido en incremento, tanto así que cada vez son más los usuarios que prefieren utilizar aplicaciones móviles para realizar sus diferentes actividades cotidianas. Al ser accesibles y al alcance de todas aquellas personas que posean un smartphone, son una alternativa viable para la realización de tareas tanto de ocio, laborales o educativas.

A nivel educativo superior la implementación de aplicaciones móviles ha mejorado los aspectos de aprendizaje de los estudiantes, ya que permite que estos tengan acceso rápido a la información que requieren permitiendo que aprendan desde cualquier lugar y realicen sus actividades educativas de una forma más didáctica e interactiva. [1] [2]

Como tarea de cada uno de los estudiantes universitarios, se encuentra la realización de un portafolio académico que cuenta con el registro de todas sus actividades, trabajos y seguimiento diario de sus clases por asignatura, a lo largo de un período.

Dicho portafolio cuenta con un formato que muchas veces no es respetado por los estudiantes y la elaboración del mismo puede llegar a ser una tarea tediosa, es por esto que se propone desarrollar una aplicación móvil para la gestión de portafolios académicos de la Facultad de Ingeniería Civil de la Universidad Técnica de Machala, con el fin de mejorar este proceso.

Para lograr el funcionamiento de dicha aplicación se necesita de un back-end robusto que permita acceder a la información de los estudiantes, docentes, carreras, asignaturas de la facultad mediante la creación de APIs que serán consumidas por la aplicación.

Se utilizará NodeJS y Flask para el back-end y la aplicación será desarrollada en el framework multiplataforma Flutter, que se trata de tecnología nueva que permite el desarrollo de aplicaciones para diferentes sistemas operativos a partir de una única base de código. [3]

El presente documento se encuentra estructurado de la siguiente forma:

Capítulo 1: se detalla la problemática, la justificación de la presente investigación y los requerimientos del prototipo a desarrollar.

Capítulo 2: se presenta la definición del prototipo, su fundamentación teórica, además de definir la metodología, los objetivos y el diseño del prototipo.

Capítulo 3: se verifica la funcionalidad del prototipo mediante el plan de evaluación y los resultados de dicha evaluación.

CONTENIDO

DEDICATORIA.....	1
AGRADECIMIENTO.....	2
RESUMEN.....	3
ABSTRACT.....	5
INTRODUCCIÓN.....	7
1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.....	14
1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.....	14
1.2. ESTABLECIMIENTO DE REQUERIMIENTOS.....	15
1.3. JUSTIFICACIÓN DE REQUERIMIENTOS A SATISFACER.....	16
2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO.....	17
2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.....	17
2.1.1. Arquitectura de software.....	17
2.1.1.1. Modelado de tres capas.....	18
2.1.1.1.1. Capa de presentación.....	18
2.1.1.1.1.1. Cliente.....	19
2.1.1.1.2. Capa lógica.....	19
2.1.1.1.2.1. Servidores web.....	19
2.1.1.1.3. Capa de datos.....	19
2.1.1.1.3.1. Bases de datos.....	19
2.1.1.1.3.2. Bases de datos relacionales.....	20
2.1.1.1.3.3. Bases de datos no relacionales.....	20
2.1.2. Herramientas CASE.....	20
2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.....	20
2.2.1. Flutter.....	21
2.2.2. jQuery.....	22
2.2.3. NodeJS.....	22

2.2.4.	Flask.....	22
2.2.5.	Firestore.....	22
2.2.6.	PostgreSQL.....	22
2.2.7.	MongoDB.....	23
2.2.8.	Enterprise Architect.....	23
2.2.9.	Toad Data Modeler.....	23
2.2.10.	Balsamiq Mockups.....	23
2.2.11.	StarUML.....	24
2.2.12.	Bizagi.....	24
2.2.13.	Metodología de desarrollo de software.....	24
2.2.13.1.	Metodologías tradicionales.....	24
2.2.13.2.	Metodologías ágiles.....	25
2.2.13.3.	Metodologías híbridas.....	25
2.2.13.4.	Metodología SWIRL.....	26
2.3.	OBJETIVOS DEL PROTOTIPO.....	26
2.3.1.	Objetivo general.....	26
2.3.2.	Objetivos generales.....	27
2.4.2.1.2.	Factibilidad técnica.....	29
2.4.3.	Fase de planificación.....	30
2.4.3.1.	Gestión de cronograma.....	31
2.4.3.2.	Historias de usuario.....	32
2.4.4.	Fase de modelado.....	35
2.4.4.1.	Diseño de base de datos relacional.....	36
2.4.4.2.	Diseño de base de datos no relacional.....	37
2.4.4.3.	Diseño navegacional.....	38
2.4.4.3.1.	Diseño navegacional del alumno.....	38
2.4.4.3.2.	Diseño navegacional del docente.....	39
2.4.4.4.	Diseño de interfaces.....	40

2.4.4.5. Diseño de diagramas UML.....	43
2.4.4.5.1. Diagramas de caso de uso.....	43
2.4.4.5.2. Diagramas de actividades.....	46
2.4.4.5.3. Diagramas de secuencias.....	47
2.5. EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO.....	48
2.5.1. Módulo inicio de sesión.....	48
2.5.2. Módulo portafolios.....	49
2.5.3. Módulo de navegación.....	51
2.5.4. Módulo matriculación.....	53
2.5.5. Módulo asignación de clave.....	56
2.5.6. Módulo estructura de portafolio.....	57
2.5.7. Módulo diario metacognitivo.....	58
3. CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO.....	59
3.1. PLAN DE EVALUACIÓN.....	59
3.2. RESULTADO DE LA EVALUACIÓN.....	60
3.3. EVALUACIÓN CON HERRAMIENTAS.....	63
3.4. CONCLUSIONES.....	66
3.5. RECOMENDACIONES.....	67
BIBLIOGRAFÍA.....	68
ANEXOS.....	72
Anexo 1: Evaluación de análisis dinámico con la herramienta MobSF.....	72
Anexo 2: Resultado de evaluación de análisis dinámico con la herramienta MobSF.....	73
Anexo 3: Modelo entrevista a alumnos de la carrera de Ingeniería de Sistemas para recolección de requerimientos.....	74
Anexo 4: Modelo entrevista a docentes y coordinador de la carrera de Ingeniería de Sistemas para recolección de requerimientos.....	74
Anexo 5: Análisis de riesgos.....	75

ÍNDICE DE TABLAS

Tabla 1: Especificación de requerimientos funcionales.....	28
Tabla 2: Especificación de requerimientos no funcionales.....	28
Tabla 3: Factibilidad operativa.....	29
Tabla 4: Factibilidad técnica.....	30
Tabla 5: Historia de usuario Inicio de sesión en la aplicación.....	32
Tabla 6: Historia de usuario Asignación clave de ingreso.....	32
Tabla 7: Historia de usuario Matriculación alumno.....	33
Tabla 8: Historia de usuario Inicio de asignaturas alumno.....	33
Tabla 9: Historia de usuario Notificaciones.....	34
Tabla 10: Historias de usuario Estructura del portafolio.....	34
Tabla 11: Historias de usuario Descarga del portafolio.....	35
Tabla 12: Historias de usuario Elaboración de diarios.....	35
Tabla 13: Resultado de evaluación de calidad.....	60
Tabla 14: Resumen resultado de evaluación de calidad.....	62

ÍNDICE DE FIGURAS

Figura 1: Arquitectura cliente-servidor del proyecto.....	17
Figura 2: Modelo de tres capas.....	18
Figura 3: Fundamentación teórica del prototipo.....	21
Figura 4: Fases del ciclo de vida de la metodología SWIRL.....	26
Figura 5: Cronograma de actividades.....	31
Figura 6: Base de datos relacional.....	36
Figura 7: Base de datos no relacional.....	37
Figura 8: Diseño navegacional del alumno.....	38
Figura 9: Diseño navegacional del docente.....	39
Figura 10 Diseño: Inicio de sesión.....	40
Figura 11 Diseño: Inicio.....	40
Figura 12 Diseño: Estructura portafolio.....	41
Figura 13 Diseño: Elaboración de diario.....	41
Figura 14 Diseño: Matriculación - carreras.....	42
Figura 15 Diseño: Matriculación - asignaturas.....	42

Figura 16 Diseño: Matriculación - insertar clave.....	43
Figura 17: Caso de uso general del sistema.....	43
Figura 18: Caso de uso módulo inicio de sesión.....	44
Figura 19: Caso de uso matriculación.....	44
Figura 20: Casos de uso portafolios.....	45
Figura 21: Caso de uso procesos de docente.....	45
Figura 22: Diagrama de actividades portafolio.....	46
Figura 23: Diagrama de actividades matriculación.....	46
Figura 24: Diagrama de actividades asignación de claves.....	47
Figura 25: Diagrama de actividades asignación de claves.....	47
Figura 26: Diagrama de secuencia Login.....	47
Figura 27: Diagrama de secuencia portafolios.....	48
Figura 28: Ensamblaje Módulo inicio de sesión.....	49
Figura 29: Ensamblaje Módulo portafolios del alumno.....	50
Figura 30. Ensamblaje Módulo portafolios del docente.....	51
Figura 31: Ensamblaje Módulo de navegación del alumno.....	52
Figura 32: Ensamblaje Módulo de navegación del docente.....	53
Figura 33: Módulo matriculación.....	54
Figura 34: Módulo matriculación colocar clave de acceso.....	55
Figura 35: Módulo matriculación matrícula exitosa.....	56
Figura 36: Módulo asignación de clave.....	57
Figura 37: Módulo estructura del portafolio.....	58
Figura 38: Módulo diario metacognitivo.....	59
Figura 39: Resultado de evaluación de calidad.....	63
Figura 40: Resultado evaluación Axe Edit TextName.....	63
Figura 41: Resultado evaluación Axe No violations found.....	64
Figura 42: Evaluación análisis dinámico calificación.....	71
Figura 43: Evaluación análisis dinámico opciones.....	71
Figura 44: Evaluación análisis dinámico en ejecución.....	71
Figura 45: Evaluación análisis dinámico reporte.....	72

1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.

1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.

La Universidad Técnica de Machala es una institución de educación superior ubicada en la ciudad de Machala que cuenta actualmente con cinco facultades, entre estas la Facultad de Ingeniería Civil (FIC), la cual posee cuatro carreras ofertadas al público: Ingeniería Civil, Ingeniería de Sistemas, Tecnologías de la Información e Ingeniería Ambiental.

La realización de un portafolio académico es trabajo de cada estudiante de las diferentes carreras que componen la Universidad Técnica de Machala, donde se recopilan todas sus actividades y seguimiento de sus clases a lo largo de un Período académico. Todo esto con el fin de que sirva como retroalimentación de aprendizaje para el maestro como para el alumno.

El formato que posee este portafolio muchas veces los estudiantes lo pasan por alto, dando como resultado una desorganización del contenido.

Actualmente existen muchas herramientas tecnológicas que pueden facilitarnos algunas tareas de la vida cotidiana, como lo pueden las páginas web y las aplicaciones móviles. Estas últimas han ido en aumento en los últimos años ya que son accesibles para casi la mayoría de personas y es mucho más práctico poder realizar tareas desde un dispositivo móvil que tenemos a la mano.

Es por esto que la aplicación móvil desarrollada bajo el framework de Flutter, permitirá a los estudiantes realizar la elaboración de su portafolio de una manera más fácil y organizada. La redacción de diarios de clase será mucho más interactiva y la estructura del portafolio estará correctamente elaborada como su formato lo estipula, liberando al estudiante de aquella carga y facilitando el trabajo del docente en su retroalimentación.

1.2. ESTABLECIMIENTO DE REQUERIMIENTOS

Con respecto a la realización de un portafolio académico, se requiere que el formato y organización del mismo sea el estipulado por la universidad, por lo tanto, al desarrollar la aplicación móvil se requieren establecer distintos parámetros para su aplicación. Es por esto que se dialogó con el coordinador, docentes y estudiantes de la carrera de Ingeniería de Sistemas (Anexo 3 y 4), obteniendo los siguientes requerimientos:

- La aplicación permitirá el ingreso únicamente de estudiantes y docentes de las distintas asignaturas impartidas en la facultad.
- Los estudiantes podrán matricularse en las asignaturas que cursan mediante una clave asignada por el docente.
- Una vez matriculado el alumno en la asignatura, se crea automáticamente la estructura del portafolio para dicho estudiante.
- Los diarios metacognitivos se crean automáticamente según las horas de clase establecidas por el docente al momento de asignar una clave a la asignatura.
- El estudiante mediante la aplicación podrá escribir en los diarios metacognitivos el contenido del diario en sí, sin preocuparse del formato del mismo.
- Al finalizar el portafolio, el alumno y el docente tendrán la opción de descargarlo en formato .rar con todas sus carpetas establecidas.
- Los alumnos recibirán diariamente una notificación como recordatorio de realizar el diario metacognitivo de ese día para estar al día con su elaboración.

1.3. JUSTIFICACIÓN DE REQUERIMIENTOS A SATISFACER.

La razón fundamental de la realización de esta aplicación es mejorar el proceso de elaboración del portafolio académico, permitiendo que los estudiantes posean una herramienta que los ayude durante su desarrollo, al alcance de su mano desde un dispositivo móvil ya sea Android o iOS.

Dentro del proceso de aprendizaje de un estudiante universitario, la elaboración del portafolio académico permite observar cómo ha sido su trayectoria de estudio y el cómo ha ido evolucionando en los diferentes retos que le ha presentado la asignatura, como si se tratase del diario del protagonista de alguna historia. [4]

De ninguna manera se debe tratar al portafolio académico únicamente como un reciclaje de todas las tareas y documentos utilizados a lo largo del Período académico, sino más bien enfocarlo a qué podemos hacer con esa información recopilada, lo que se puede aprender de los procesos y resultados que se obtuvieron durante un curso. [5] [6]

Los sistemas operativos van mejorando en su funcionalidad debido al entorno en el que deben ejecutarse. Uno de los aspectos importantes en el desarrollo de aplicaciones móviles es la compatibilidad [7], es por esto que se decidió desarrollar una aplicación óptima para la elaboración del portafolio académico, utilizando tecnología nueva como lo son los frameworks multiplataforma.

La aplicación móvil a desarrollar permitirá a los estudiantes y docentes de la Facultad de Ingeniería Civil, tener acceso a un portafolio con formato y organización correcta para su retroalimentación del aprendizaje del alumno.

La elaboración de los diarios metacognitivos será mucho más sencilla y práctica para los estudiantes que únicamente tendrán que preocuparse de la información que va dentro de estos, reforzando así sus conocimientos y mejorando el proceso enseñanza-aprendizaje.

2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO

A continuación, se detalla la estructura general del sistema a desarrollar, ya que este posee una estructura específica para este tipo de proyecto de aplicación móvil.

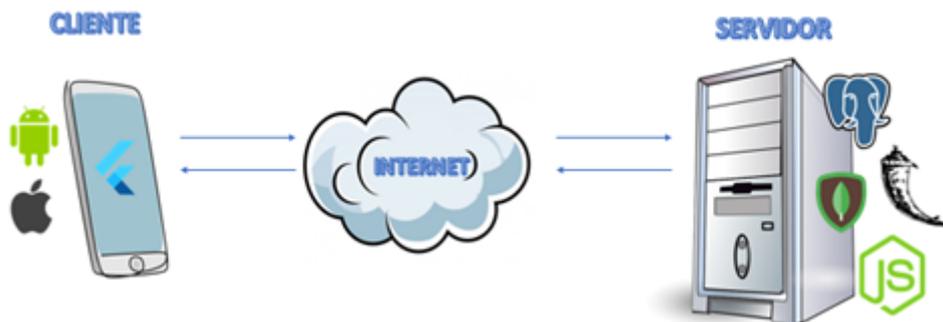
2.1.1. Arquitectura de software

Un modelo de arquitectura de software corresponde a la estructura general del sistema en sí, compuesta por distintos elementos donde cada uno de ellos pueden llegar a afectar los requerimientos tanto funcionales como no funcionales del sistema. [8] [9]

Es por esta razón que los modelos de arquitectura de software están encargados de soportar múltiples tareas de desempeño, tales como el rendimiento, el tiempo de respuesta de peticiones, operaciones del sistema. [10]

La arquitectura de software que se ha escogido para este proyecto es la de arquitectura cliente-servidor, donde la computadora o dispositivo final, desde donde esté el usuario, realiza una solicitud a los servidores que contienen la información que el cliente requiere, aceptan la petición y la responden, finalmente el usuario la receipta en pantalla y puede interpretarla. [11]

Figura 1: Arquitectura cliente-servidor del proyecto



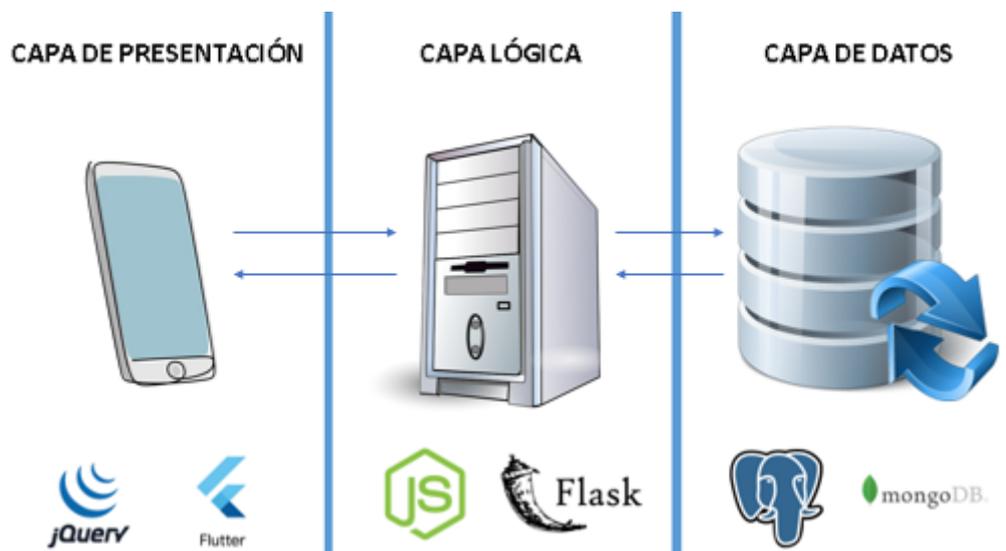
Fuente: Elaboración propia

2.1.1.1. Modelado de tres capas

La arquitectura de tres capas es muy utilizada en distintos tipos de sistemas, puesto que esta permite de manera muy eficaz que los sistemas realicen las tareas que deben llevar a cabo de forma independiente, sin verse afectadas otras capas, ayudando así a su flexibilidad y que su rendimiento sea el más óptimo. [12]

El modelado de tres capas se compone de las siguientes: presentación, lógica y datos. En la siguiente figura se muestra el modelado de tres capas del presente proyecto.

Figura 2: Modelo de tres capas



Fuente: Elaboración propia

2.1.1.1.1 Capa de presentación

La capa de presentación corresponde del lado del usuario que va a hacer uso del sistema. En este se capturan las peticiones, las cuales serán de obtener y enviar datos que posteriormente serán enviados a la siguiente capa, es decir a la capa lógica. [12]

2.1.1.1.1. Cliente

En la capa cliente, tal como lo dice su nombre, el usuario será el protagonista principal ya que va a interactuar de manera directa con el sistema de manera visual. Este se encargará de obtener y enviar datos mediante los métodos GET y POST respectivamente. Ambos métodos tienen contacto tanto con el servidor como con las bases de datos. [13]

2.1.1.1.2. Capa lógica

Dentro de la capa lógica se encuentran los llamados servidores, en caso de este proyecto, dos servidores web bajo el sistema operativo Centos8: uno con el framework de NodeJS para realizar las APIs de consulta y modificación de la base de datos PostgreSQL y un framework en Flask para la modificación de portafolios académicos por cada estudiante en MongoDB.

2.1.1.1.2.1. Servidores web

Los servidores web responden las solicitudes que provienen de un navegador o de una API, proporcionando la información requerida mediante el protocolo HTTP. Estos tienen conexión directa con la capa de presentación, ya que esta es la que se encarga de realizar las peticiones o solicitudes. Mediante esta conexión es que se efectúan los almacenamientos o consulta de información. [14]

2.1.1.1.3. Capa de datos

Finalmente se encuentra la capa de datos, que es donde se gestionan las diferentes bases de datos que posea un sistema y mantiene una conexión directa con la capa lógica descrita anteriormente.

2.1.1.1.3.1. Bases de datos

Se establecieron dos bases de datos para trabajar en el presente proyecto: PostgreSQL (Base de datos relacional) y MongoDB (Base de datos no relacional).

Donde la relacional tendrá la información de los docentes, alumnos, asignaturas y carreras, y la no relacional guardará la estructura del portafolio académico creado para cada asignatura por alumno.

2.1.1.1.3.2. Bases de datos relacionales

Las bases de datos relaciones permiten elaborar un modelo relacional con la inserción de atributos, índices, claves que servirán para el ingreso, edición y eliminación de datos correspondientes a una tabla o modelo. [15] La base de datos relacional escogida por su reputación y mejor adaptación para este proyecto, es PostgreSQL. [16]

2.1.1.1.3.3. Bases de datos no relacionales

Son aquellas bases de datos donde su información no se encuentra unida entre sí mediante tablas, como es el caso de las bases de datos relacionales. Este tipo de bases de datos escala horizontalmente y evita los problemas de unión de datos que pueden darse. [17] [18] La base de datos no relacional escogida para este proyecto de investigación fue MongoDB

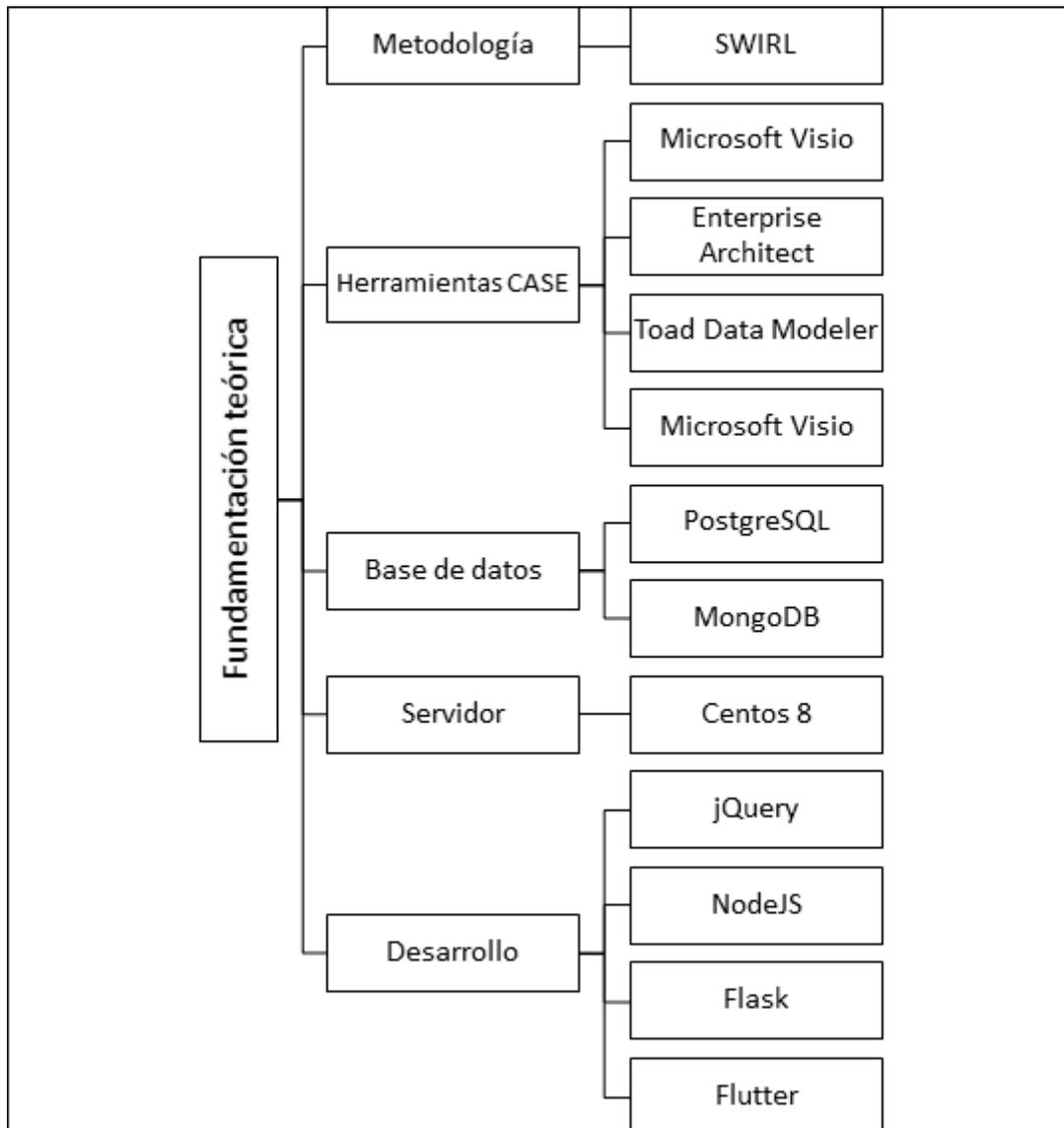
2.1.2. Herramientas CASE

Estas herramientas ayudan directamente al proceso de desarrollo de software, ya que mejoran la productividad y la calidad de los sistemas al acelerar el proceso de elaboración, liberando a los usuarios de ese esfuerzo manual y el consumo de tiempo que antes se requería para estos procesos. [19] [20]

2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO

A continuación, se presenta la fundamentación teórica del prototipo del presente documento donde se detallan las herramientas de desarrollo, base de datos, CASE y la metodología y servidor usados durante el proyecto.

Figura 3: Fundamentación teórica del prototipo



Fuente: Elaboración propia

2.2.1. Flutter

El framework multiplataforma Flutter desarrollado por Google, está enfocado en el desarrollo de aplicaciones interactivas para distintos sistemas operativos desde un único código. Los widgets que se utilizan en Flutter comparten las mismas características de un desarrollo nativo, tales como el desplazamiento de los componentes, la navegación y entre otros, utilizando tecnología nueva que permite diseñar interfaces, crear funciones y corregir errores fácilmente. [3]

2.2.2. jQuery

jQuery fue utilizado en el presente proyecto ya que, al ser una biblioteca de JavaScript rápida, hace que su funcionamiento en cuanto a manipulación de documentos HTML, eventos, entre otros, sean mucho más práctico de usar mediante una API. [21]

2.2.3. NodeJS

NodeJS cuenta con una variedad de librerías disponibles para desarrollar funciones que pueden ser accedidas mediante un navegador ejecutándose en un solo proceso haciéndolo mucho más eficaz. [22] [23]

2.2.4. Flask

Framework Flask proporciona herramientas y funciones para el usuario totalmente funcionales que ayudan en la creación de aplicaciones web en Python. En lo que comprende a la creación de APIs su aplicación es muy sencilla, permitiendo desarrollar estas de manera rápida y simple. [24]

2.2.5. Firebase

Se conoce a Firebase como una plataforma para el desarrollo de aplicaciones móviles o web. Entre los grupos más interesantes de funciones que posee se encuentra la función de desarrollo, donde se encuentran todos los módulos necesarios para desarrollar una aplicación, entre estos se encuentra realtime database o bases de datos en tiempo real. [25]

Otra de las funciones de Firebase se encuentra cloud messaging, el envío de notificaciones y mensajes, la cual se implementará en la aplicación móvil.

2.2.6. PostgreSQL

PostgreSQL conocido gestor de bases de datos relacional, es un software libre que posee integridad referencial, es decir, que las claves externas deben existir en una fila válida de ambas tablas que se están referenciando, dándole así su característica de relacional, ya que permite que las tablas se conecten sus datos entre sí. [26]

2.2.7. MongoDB

MongoDB es un gestor de bases de datos no relacional que usa como base el tipo de documentos llamado JSON. La estructura de los datos encontrados en archivos JSON es muy sencilla, donde la información se encuentra en llaves, identificando cada valor por medio de una clave, generalmente separado por comas. Estos documentos son agrupados en colecciones y estos pueden tener esquemas independientes. [27]

2.2.8. Enterprise Architect

Enterprise Architect es una herramienta diseñada para la elaboración de diagramas en Lenguaje Unificado de Modelado (UML), lo que ayuda en la etapa de diseño desde la recolección de requisitos del sistema. [26]

Apoya en el desarrollo de sistemas de gran calibre dando un mantenimiento de los procesos y mejoramiento de los mismos. [28]

2.2.9. Toad Data Modeler

Toad Data Modeler es una herramienta CASE que permite la creación y diseño de bases de datos relacionales que al final genera un código SQL que puede ser colocado en un gestor de base de datos relacional para su creación. [29]

Esto agiliza el proceso de creación de tablas, primary keys, foreign keys y demás que son característicos de una base de datos relacional.

2.2.10. Balsamiq Mockups

Balsamiq Mockups es una herramienta de diseño de interfaces de usuario para cualquier tipo de sistema. Cuenta con una interfaz sencilla y amigable con el usuario además de tener distintos diseños y elementos que se pueden acoplar a la interfaz que el usuario tenga en mente para diseñar. Permite realizar borradores de interfaces para ayudar a crear mejor la idea de una interfaz, simulando estar en el entorno en el que se desea aplicar el sistema. [30] [31]

2.2.11. StarUML

StarUML es una herramienta basada en un UML para el desarrollo de modelado de software. Este nos permite realizar diagramas de caso de uso, diagramas de clases, diagramas de actividades, diagramas de secuencia, entre otros, de una forma rápida y sencilla. Su interfaz es amigable e intuitiva para el usuario, lo que la convierte en una herramienta ideal para proyectos de software. [32]

2.2.12. Bizagi

Bizagi es un modelador de procesos de negocios caracterizado por ser potente, amigable con el usuario y gratuito. En esta herramienta se pueden desarrollar diagramas de flujo de trabajo, diagramas de actividades con la particularidad de tener un servicio en la nube donde se pueden publicar los diagramas elaborados y acceder a ellos cuando se requiera. [33]

2.2.13. Metodología de desarrollo de software

Se conoce como metodología de desarrollo de software a los lineamientos y prácticas existentes para la correcta elaboración de un sistema.

Mediante estas directrices el proceso de elaboración de software se vuelve mucho más organizado y ágil, satisfaciendo en un mayor porcentaje las necesidades o requerimientos del usuario. [34]

Dentro de la ingeniería de software existen las fases de producción del sistema, las cuales se crean a partir de la necesidad de satisfacer las necesidades del usuario estipuladas en los requerimientos del sistema. Esto se puede lograr al seguir las actividades que sugiere una metodología para una correcta elaboración de software. [35]

2.2.13.1. Metodologías tradicionales

Las metodologías tradicionales son aquellas que se basan en un sistema de trabajo riguroso y planificado. Establece que toda actividad dentro del desarrollo de software debe ser propuesta con anterioridad y detallada. [36]

Su característica radica en que se centran en el control de todos los procesos, mediante la definición de roles, herramientas, actividades, todo lo que haga falta para un proceso de desarrollo de software y su correcta documentación. [37] [38]

Las metodologías tradicionales no se adaptan a los cambios que puedan suceder durante el desarrollo, razón por la cual se han dejado de implementar en la mayoría de proyectos de software. [39]

2.2.13.2. Metodologías ágiles

Las metodologías ágiles son aquellas donde las actividades de desarrollo se adaptan a los requerimientos del sistema a desarrollar, consiguiendo así flexibilidad y acelerar el tiempo de elaboración. [40]

Dentro de sus características, aparte de la flexibilidad, se encuentran: iterativo, traen consigo menos tiempo y costo de elaboración. En definitiva, se concentra mucho más en la codificación del proceso de desarrollo que en la documentación, a diferencia de las metodologías tradicionales. [41]

Otra diferencia que posee con las metodologías tradicionales es que las ágiles están abiertas a que ocurran posibles errores, ya que durante el proceso de elaboración permite que estos se vayan encontrando y solucionando a medida que avanza el proyecto. [42]

2.2.13.3. Metodologías híbridas

Por otro lado, las metodologías híbridas, toman todas las características positivas de las metodologías ágiles, sus prácticas, actividades, métodos, siendo así una combinación de buenas prácticas de cada una de ellas. [43]

En la ingeniería de software se ha tomado la iniciativa de formar metodologías híbridas como innovación para mejorar los procesos de desarrollo, creando un método fuerte y flexible que combina, hasta algunas veces, más de dos metodologías ágiles. [43] [44]

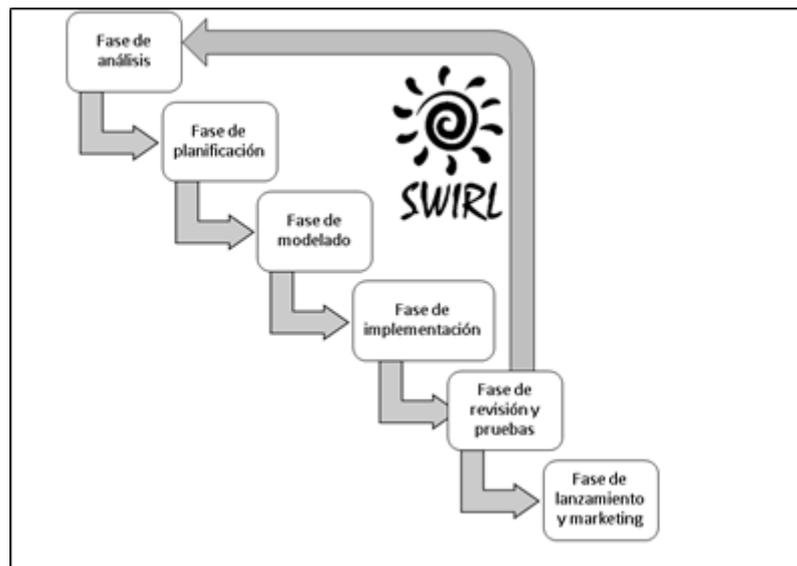
2.2.13.4. Metodología SWIRL

La metodología escogida para utilizar en el presente proyecto es SWIRL, debido a su marco de desarrollo que brinda una guía completa para el desarrollo tanto de aplicaciones web como móviles.

Dentro de esta metodología se toma en cuenta a todos los interesados del proyecto además de proveer métodos y actividades para un mejor proceso de desarrollo de software. [45] [46]

SWIRL posee siete fases para el ciclo de vida de un proyecto de software, las cuales se detallan en la siguiente figura:

Figura 4: Fases del ciclo de vida de la metodología SWIRL



Fuente: Elaboración propia

2.3. OBJETIVOS DEL PROTOTIPO

2.3.1. Objetivo general

Desarrollar una aplicación móvil para la gestión de portafolios de la Facultad de Ingeniería Civil utilizando la metodología SWIRL, lenguaje de programación Dart, framework multiplataforma Flutter, framework web NodeJS y Flask y bases de datos PostgreSQL y MongoDB, para la automatización del proceso de elaboración del portafolio de la asignatura del estudiante.

2.3.2. Objetivos generales

- Buscar en fuentes bibliográficas sobre el desarrollo de aplicaciones móviles multiplataforma.
- Aplicar las fases de la metodología SWIRL para elaborar un marco de trabajo en base a estas.
- Definir los requerimientos funcionales y no funcionales en historias de usuarios mediante entrevistas a las partes interesadas.
- Diseñar las diferentes vistas de la aplicación móvil mediante el uso de la herramienta CASE: Balsamiq Mockups.
- Diseñar e implementar la aplicación móvil a partir de los requerimientos del usuario.
- Llevar a cabo pruebas de funcionalidad y accesibilidad de la aplicación móvil.

2.4. DISEÑO DEL PROTOTIPO

La metodología SWIRL brinda las directrices a seguir para llevar a cabo una correcta documentación del desarrollo de software. El presente proyecto se encuentra diseñado bajo las siguientes fases:

2.4.1. Fase de análisis

Dentro de esta fase se llevan a cabo las actividades iniciales para la definición del sistema, considerándose primordial en el desarrollo de un software.

2.4.2. Alcance del proyecto

La aplicación móvil del presente trabajo tiene como propósito mejorar el proceso de elaboración del portafolio académico de los estudiantes de la Facultad de Ingeniería Civil.

A continuación, se presentan los requerimientos funcionales y no funcionales que tendrá el sistema a desarrollar:

Tabla 1: Especificación de requerimientos funcionales

ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES
RF-01: Permite el ingreso al sistema mediante un Login con cédula y contraseña.
RF-02: Permite al docente asignar una clave de acceso a la asignatura que este imparte durante un Período académico
RF-03: Permite al alumno matricularse a una asignatura mediante la clave de acceso asignada por el docente.
RF-04: El alumno podrá ver aquellas asignaturas en las que se encuentra matriculado.
RF-05: Los estudiantes recibirán una notificación diaria recordándoles que deben realizar el diario de ese día.
RF-06: Crea automáticamente al matricularse a una asignatura, la estructura del portafolio de la misma.
RF-07: Permite el ingreso de información para los diarios metacognitivos mediante un formulario.
RF-08: Permite descargar el portafolio de la asignatura.

Fuente: Elaboración propia

Tabla 2: Especificación de requerimientos no funcionales

ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES
RNF-01: Las interfaces tienen que ser intuitivas y amigables con el usuario
RNF-02: Mantener seguro datos sensibles de los usuarios
RNF-03: Pestañas y vistas responsivas y compatibles con cualquier tipo de pantalla
RNF-04: Tiempos de carga y obtención de información rápidos

Fuente: Elaboración propia

2.4.2.1. Estudio de factibilidad

El presente estudio de factibilidad determinará qué tan factible es la implementación de la aplicación móvil para la gestión de portafolios académicos.

2.4.2.1.1. Factibilidad operativa

Tabla 3: Factibilidad operativa

Factibilidad operativa				
No.	Actividad	Priorización		
		Alta	Media	Leve
1	Se cuenta con los elementos informáticos necesarios para la implementación de la aplicación móvil	X		
2	Se posee los conocimientos técnicos y de desarrollo para la elaboración de aplicaciones móviles	X		
3	Necesidad de capacitación para explicar el funcionamiento del sistema		X	

Fuente: Elaboración propia

Mediante la presente información, se puede deducir que es factible la realización de la propuesta ya que el entorno para la implementación es óptimo.

2.4.2.1.2. Factibilidad técnica

Al tratarse de un sistema para la gestión de portafolios académicos de los estudiantes de las carreras de la Facultad de Ingeniería Civil, se necesita de un back-end robusto capaz de manejar un volumen de información alto.

En el portafolio del estudiante se encuentran los documentos y diarios de clase del alumno, es decir que cada estudiante deberá tener una cierta cantidad de archivos dependiendo de su asignatura.

La cantidad de diarios máxima que un alumno puede tener es de treinta y dos y contando los documentos tales como: tareas intraclases y extra clase, syllabus de la asignatura, proyectos, evaluaciones, etc., se puede deducir que aproximadamente un estudiante subiría una cantidad de doscientos archivos al portafolio.

Si establecemos 5 mb como mínimo de peso para los archivos, tendríamos que cada estudiante ocuparía 1 GB de información por portafolio académico.

A continuación, se muestran los recursos de hardware y software planteados para el correcto funcionamiento del sistema.

Tabla 4: Factibilidad técnica

Factibilidad técnica			
Recursos de hardware		Recursos de software	
Cantidad	Descripción	Cantidad	Descripción
1	Procesador i7 de 8va generación o superior	3	Centos8 o Debian
2	Procesador i5 de 8va generación o superior	2	Bases de datos: PostgreSQL y MongoDB
3	Memoria RAM DE 16GB	1	Dominio
2	Memoria RAM de 8GB	1	IP pública
3	SSD de 480GB	3	Frameworks: ReactJS, NodeJS, Flask
1	HDD de 4TB		

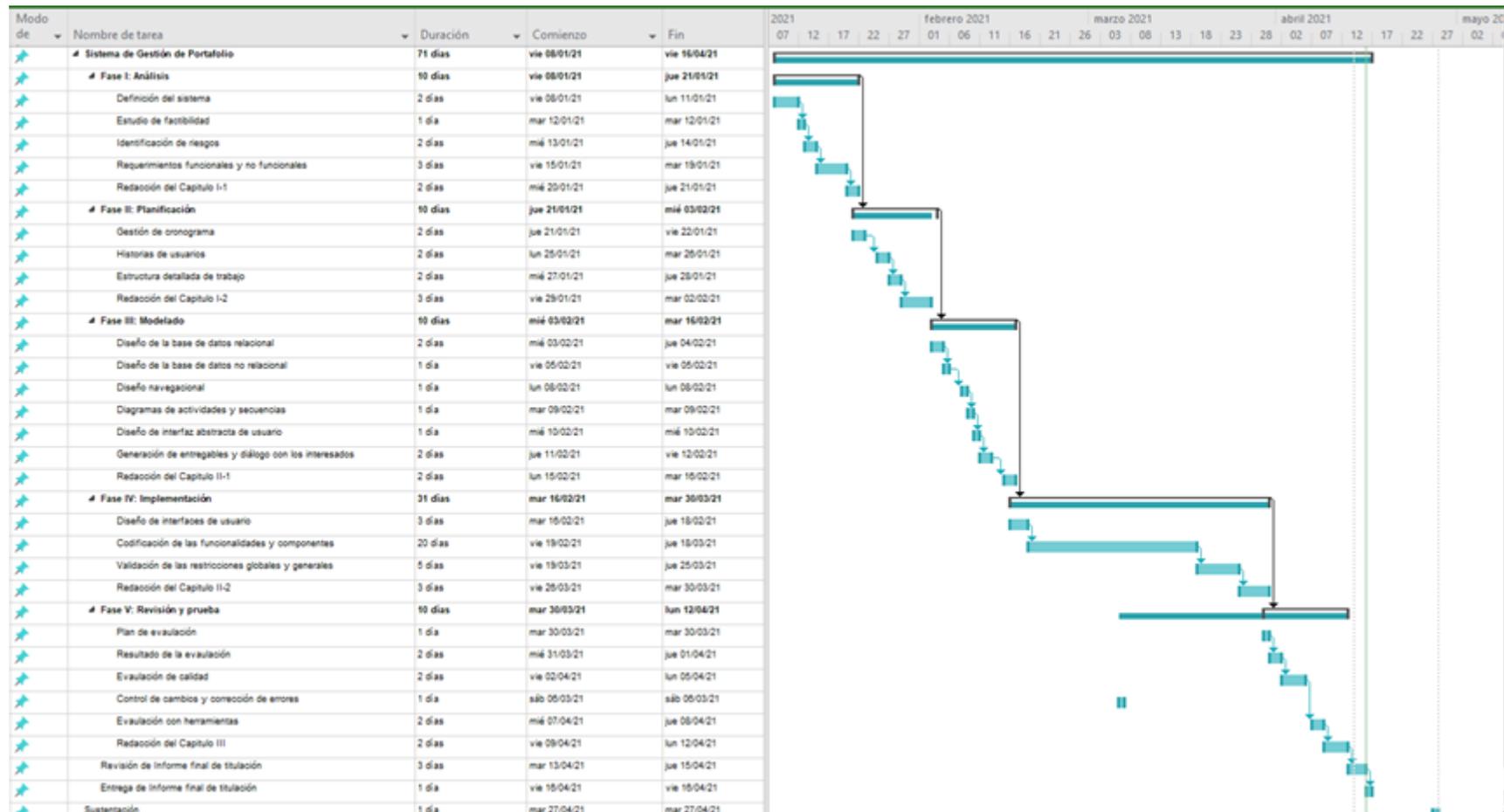
Fuente: Elaboración propia

2.4.3. Fase de planificación

En la fase de planificación se lleva a cabo la realización del cronograma de actividades y el desarrollo de las historias de usuario del sistema.

2.4.3.1. Gestión de cronograma

Figura 5: Cronograma de actividades



Fuente: Elaboración propia

2.4.3.2. Historias de usuario

Tabla 5: Historia de usuario Inicio de sesión en la aplicación

Historias de usuario	
Número: 1	Usuario: Alumnos y docentes
Nombre historia: Inicio de sesión en la aplicación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-01	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: La aplicación móvil debe poseer un inicio de sesión, tanto para alumnos y docentes, donde se coloque la cédula del usuario y su contraseña.	
Observaciones: Mostrar un mensaje de error cuando los datos ingresados por el usuario no se encuentren en la base de datos relacional, es decir datos erróneos.	

Fuente: Elaboración propia

La primera vez que el alumno acceda al sistema, se le presentará una pestaña para cambiar su contraseña actual, la cual hasta el momento de ingreso es su número de cédula.

Tabla 6: Historia de usuario Asignación clave de ingreso

Historias de usuario	
Número: 2	Usuario: Docentes
Nombre historia: Asignación clave de ingreso	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-02	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: El docente ingresará al menú "Asignación de claves" donde encontrará las asignaturas que el imparte, donde tendrá que seleccionar una de ellas e ingresar una clave de acceso que compartirá con sus estudiantes en su hora de clase.	
Observaciones: El docente podrá colocar una clave de acceso solo a las materias que el imparte.	

Fuente: Elaboración propia

Tabla 7: Historia de usuario Matriculación alumno

Historias de usuario	
Número: 3	Usuario: Alumnos
Nombre historia: Matriculación alumno	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-03	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: El estudiante ingresará al menú "Matriculación", donde encontrará las asignaturas correspondientes a cada carrera, donde este tendrá que seleccionar la que cursa e ingresar una clave de acceso antes asignada por el docente.	
Observaciones: Mostrar un mensaje de error cuando los datos ingresados sean erróneos.	

Fuente: Elaboración propia

Una vez matriculado el estudiante, se redirigirá a la pestaña de inicio donde podrá ver las asignaturas matriculadas incluyendo la recientemente realizada.

Tabla 8: Historia de usuario Inicio de asignaturas alumno

Historias de usuario	
Número: 4	Usuario: Alumnos
Nombre historia: Inicio de asignaturas alumno	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-04	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: Los estudiantes podrán ver el inicio de la aplicación los portafolios de las asignaturas en las que ya se ha matriculado con anterioridad.	
Observaciones: En caso de no tener ningún portafolio que mostrar, un mensaje saldrá en la ventana sugiriendo al estudiante que debería matricularse en alguna asignatura en el menú "Matriculación".	

Fuente: Elaboración propia

Tabla 9: Historia de usuario Notificaciones

Historias de usuario	
Número: 5	Usuario: Alumnos
Nombre historia: Notificaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-05	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: Los estudiantes recibirán diariamente una notificación recordándoles que deben realizar el diario de ese día.	
Observaciones: Las notificaciones no se guardarán dentro de la aplicación en ningún submenú.	

Fuente: Elaboración propia

Las notificaciones serán enviadas una vez la estructura del portafolio sea creada, con el fin de recordar al estudiante que debe elaborar los diarios metacognitivos de ese día.

Tabla 10: Historias de usuario Estructura del portafolio

Historias de usuario	
Número: 6	Usuario: Alumnos
Nombre historia: Estructura del portafolio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-06	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: Al dar clic en el portafolio de una asignatura matriculada, el alumno podrá ver la estructura del portafolio creada automáticamente con sus directorios y formato.	
Observaciones: Para poder ver la estructura del portafolio de una asignatura, el alumno debe estar matriculado a esta antes.	

Fuente: Elaboración propia

La estructura con la que se creará el portafolio es la estipulada por la Universidad Técnica de Machala a cada una de las facultades de la institución.

Tabla 11: Historias de usuario Descarga del portafolio

Historias de usuario	
Número: 8	Usuario: Alumnos y docentes
Nombre historia: Descarga del portafolio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-08	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: Tanto alumnos como docentes podrán descargar el portafolio de la asignatura en cualquier momento del curso.	
Observaciones: El portafolio académico se descargará en formato .rar con sus directorios y formato estipulado.	

Fuente: Elaboración propia

Tabla 12: Historias de usuario Elaboración de diarios

Historias de usuario	
Número: 7	Usuario: Alumnos
Nombre historia: Elaboración de diarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-07	Iteración asignada: 1
Programador responsable: Silvia Cabrera	
Descripción: Al encontrarse dentro de la estructura de un portafolio, el alumno podrá escoger un diario metacognitivo dándole clic. Este abrirá una nueva pestaña donde se podrá colocar los campos tales como: resumen, reflexión, objetivos, del día de ese diario escogido.	
Observaciones: Para la realización del diario de una asignatura, el alumno debe estar matriculado a esta antes.	

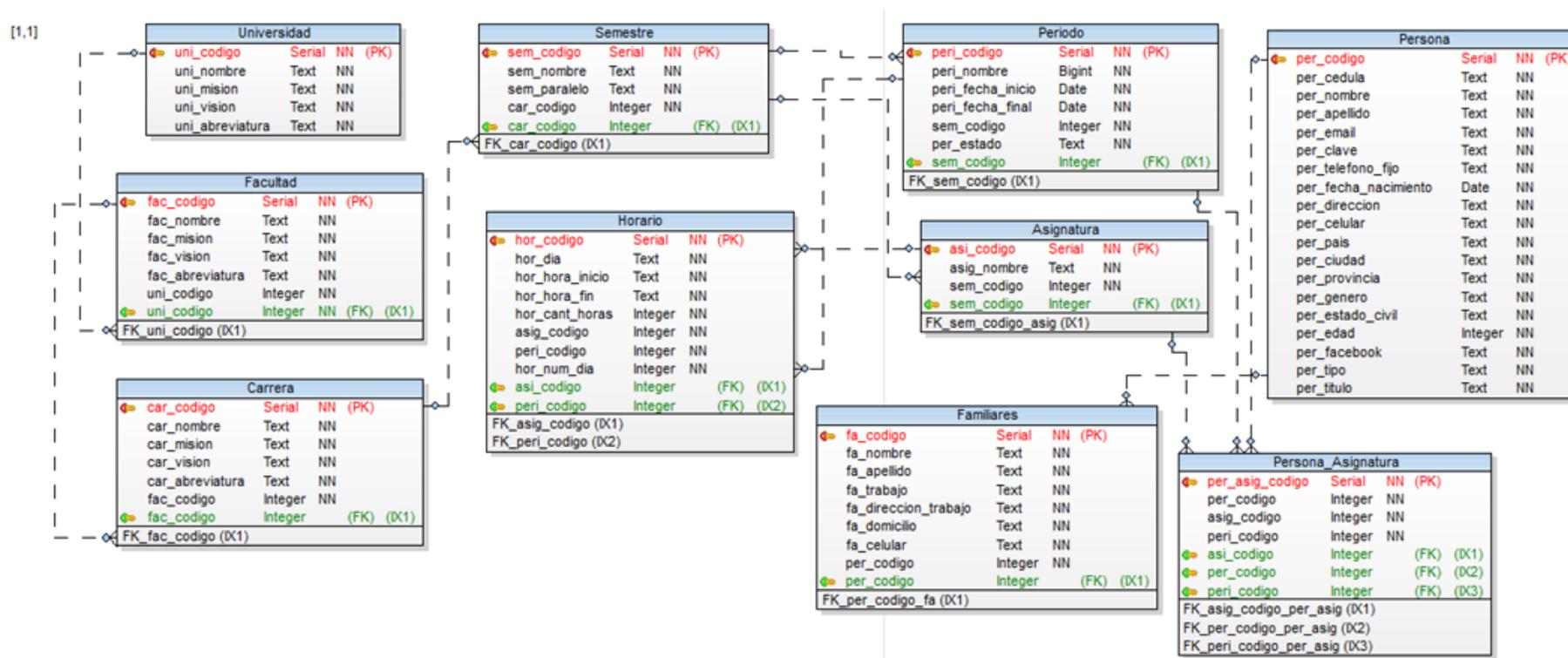
Fuente: Elaboración propia

2.4.4. Fase de modelado

Dentro de esta fase se encuentra el diseño de las bases de datos del proyecto, el diseño navegacional del sistema, el diseño de interfaces y diagramas UML del mismo.

2.4.4.1. Diseño de base de datos relacional

Figura 6: Base de datos relacional



Fuente: Elaboración propia

2.4.4.2. Diseño de base de datos no relacional

Figura 7: Base de datos no relacional

```
1
2
3 "sisport_utach": {
4   "fic.ls": [
5     {
6       "_id": {
7         "cod_asignatura": "1",
8         "periodo": "2021-01",
9         "nombre_asignatura": "PROGRAMACIÓN V",
10        "identificador": "1.5.8.12345",
11        "syllabus": "syllabus.pdf",
12        "docente": "Ing. Joofre Honores Tapia",
13        "clave": "prog202101",
14        "portafolios": [
15          {
16            "datos_informativos": {
17              "codigo_estudiante": "1055",
18              "autoretrato": "/foto_perfil.jpg",
19              "biografia": "Mi meta es ser Ingeniero de Sistemas"
20            },
21            "elementos_curriculares": {
22              "expectativas": "Aprender python de forma profesional",
23              "apuntes": [
24                {
25                  "unidad": "UNIDAD I - INSTRUCCIONES DE CONTROL DEL PROGRAMA",
26                  "tema": "Generalidades de Python",
27                  "fecha": "Viernes, 08 de Enero del 2021",
28                  "objetivo": "Analizar conceptos básicos de la programación python",
29                  "contenidos": "Enquadre de la asignatura, historia, tipos de datos",
30                  "resumen": "Python es un lenguaje potente, diseñado para proyectos multipropósitos",
31                  "reflexión": {
32                    "pregunta1": "Comprender en detalle la sintaxis del lenguaje de programación Python",
33                    "pregunta2": "Analizar la lógica de programación del lenguaje Python",
34                    "pregunta3": "Porque Python es un lenguaje de programación multiparadigma",
35                    "pregunta4": "Aprendí sobre las generalidades de Python"
36                  },
37                  "imagenes": "imagen1.jpg, imagen2.jpg"
38                }
39              ],
40              "evaluaciones": [
41                {
42                  "ruta": "/evaluación1.docx"
43                }
44              ],
45              "investigaciones": [
46                {
47                  "ruta": "/investigación1.docx"
48                }
49              ],
50              "actividades": [
51                {
52                  "ruta": "/actividad1.xls"
53                }
54              ],
55              "proyectos": [
56                {
57                  "ruta": "/proyecto.pdf"
58                }
59              ],
60              "caso_estudio": [
61                {
62                  "ruta": "/caso-de-estudio.docx"
63                }
64              ],
65              "planteamientos": [
66                {
67                  "ruta": "/planteamiento1.docx"
68                }
69              ],
70              "asistencia": "/asistencia.docx",
71              "observaciones": [
72                {
73                  "ruta": "/observación1.doc"
74                }
75              ],
76              "intraclases": [
77                {
78                  "ruta": "/intraclase1.doc"
79                }
80              ],
81              "autonomos": [
82                {
83                  "ruta": "/autonomo1.doc"
84                }
85              ],
86              "refuerzo": [
87                {
88                  "ruta": "/refuerzo1.doc"
89                }
90              ],
91              "informe_final": "/informefinal.docx"
92            }
93          }
94        ]
95      }
96    }
97  }
98 }
```

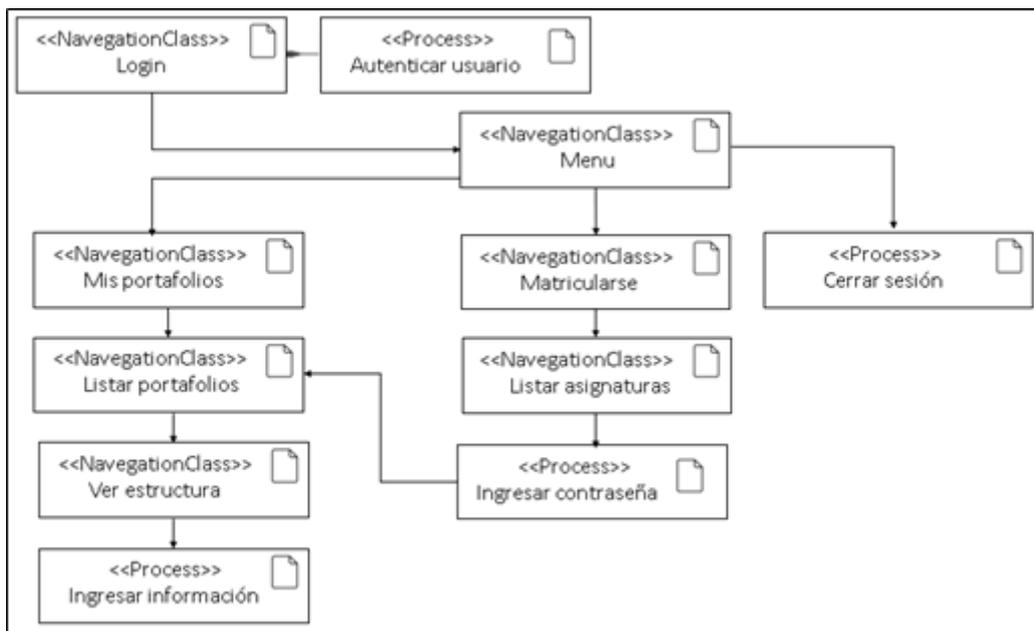
Fuente: Elaboración propia

2.4.3.3. Diseño navegacional

El sistema posee dos tipos de usuarios donde existe un diseño navegacional para cada uno de estos, ya que las ventanas y procesos son diferentes respectivamente.

2.4.3.3.1. Diseño navegacional del alumno

Figura 8: Diseño navegacional del alumno



Fuente: Elaboración propia

El alumno ingresará al sistema mediante un Login con su número de cédula y clave, luego de que el sistema autentique la sesión del alumno, este podrá navegar en las pestañas de Mis portafolios, Matricularse y Cerrar Sesión.

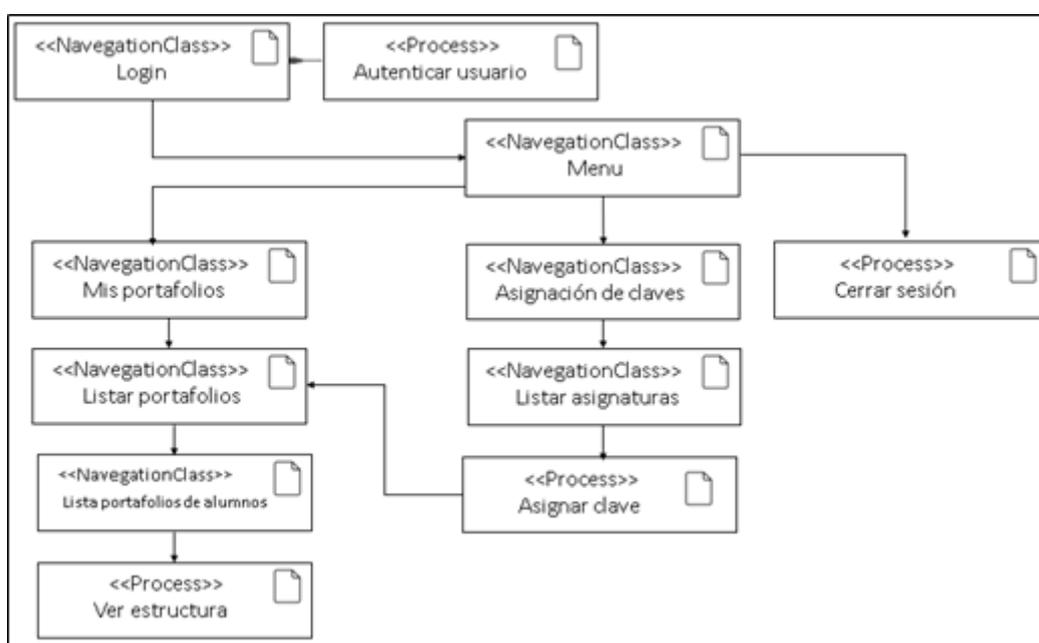
Al dar clic en el submenú Mis portafolios, se desplegará la lista de los portafolios a los que ese alumno esté matriculado. Al dar clic en alguno de estos portafolios, se podrá ver la vista de estructura de aquel portafolio y una vez dentro podrá ingresar información para guardar en el portafolio académico.

En el submenú Matricularse se mostrará una lista de las asignaturas filtradas por carreras de la facultad, donde al dar clic en una de ellas, podrá matricularse ingresando una clave de acceso previamente asignada por el docente.

Finalmente, al dar clic en el submenú Cerrar Sesión, la sesión del usuario será eliminada dirigiéndose a la vista de Login.

2.4.4.3.2. Diseño navegacional del docente

Figura 9: Diseño navegacional del docente



Fuente: Elaboración propia

El docente ingresará al sistema mediante un ingreso de sesión mediante su número de cédula y contraseña. Una vez dentro podrá navegar en la aplicación mediante los submenús de Mis portafolios, Asignación de claves y Cerrar Sesión.

Al dar clic en Mis portafolios, se mostrará una lista de las asignaturas que el imparte donde, al darle clic a cualquiera de ellas, podrá ver un listado de todos los estudiantes matriculados en tal asignatura. A su vez al dar clic en algún estudiante, podrá visualizar la estructura del portafolio de dicho alumno.

En el submenú Asignación de claves, el docente podrá seleccionar, de una lista de asignaturas filtradas por carrera, la que el imparte asignándole una clave que deberá compartir con los alumnos.

Cerrar sesión es el submenú que se encarga de borrar la sesión del usuario y volver a la pestaña del Login.

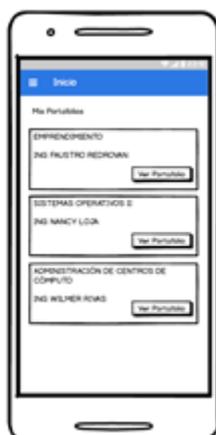
2.4.4.4. Diseño de interfaces

Figura 10 Diseño: Inicio de sesión



Fuente: Elaboración propia

Figura 11 Diseño: Inicio



Fuente: Elaboración propia

Figura 12 Diseño: Estructura portafolio



Fuente: Elaboración propia

Figura 13 Diseño: Elaboración de diario



Fuente: Elaboración propia

Figura 14 Diseño: Matriculación - carreras



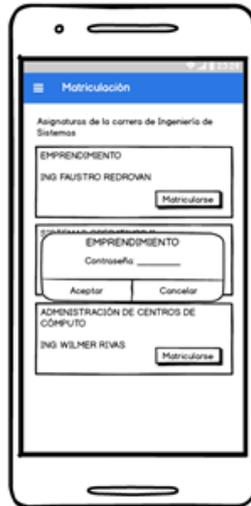
Fuente: Elaboración propia

Figura 15 Diseño: Matriculación - asignaturas



Fuente: Elaboración propia

Figura 16 Diseño: Matriculación - insertar clave



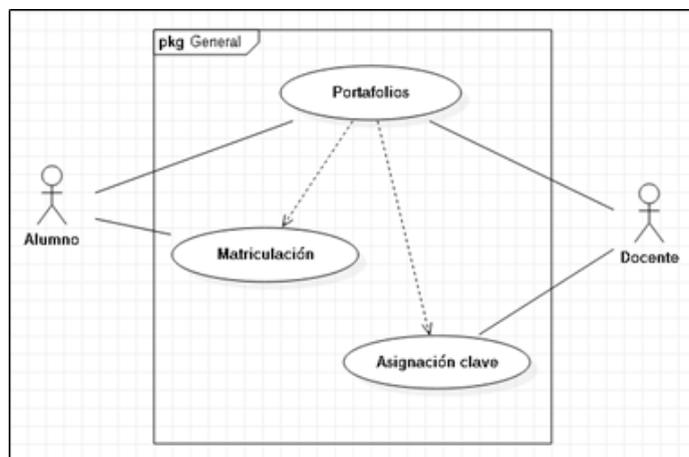
Fuente: Elaboración propia

2.4.4.5. Diseño de diagramas UML

Los diagramas UML (Lenguaje de modelado unificado) permiten documentar y diseñar un sistema de software. Estos son muy útiles en la fase de desarrollo ya que permiten tener una idea clara del funcionamiento que va a tener la aplicación.

2.4.4.5.1. Diagramas de caso de uso

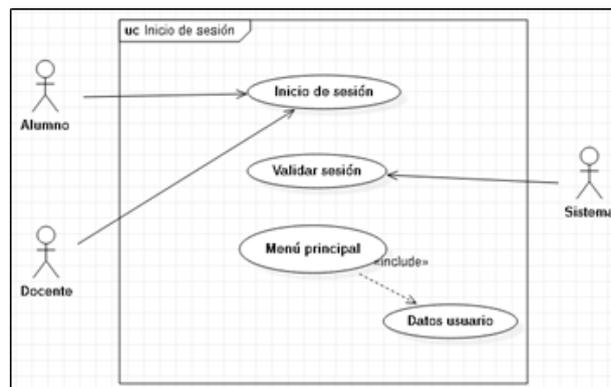
Figura 17: Caso de uso general del sistema



Fuente: Elaboración propia

El caso de uso general del sistema muestra los procesos que llevan a cabo los usuarios alumno y docente dentro de la aplicación. El alumno realiza el proceso de matriculación y posteriormente puede ver sus portafolios, por otro lado, el tipo de usuario docente realiza la asignación de clave de acceso para luego poder ver los portafolios de todos los estudiantes que se han matriculado a una asignatura de ese docente.

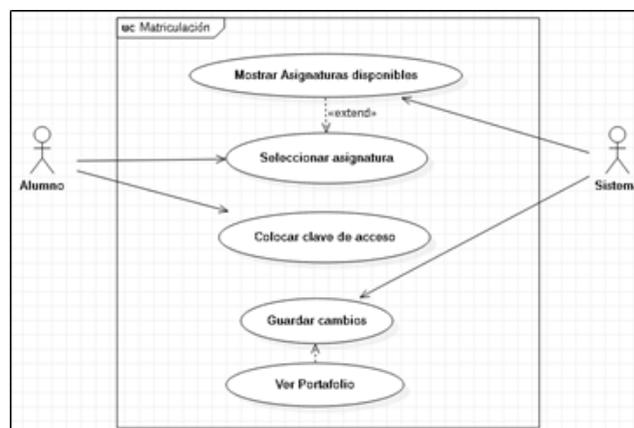
Figura 18: Caso de uso módulo inicio de sesión



Fuente: Elaboración propia

El módulo inicio de sesión es llevado a cabo tanto por docentes como por alumnos, donde esperan una respuesta del sistema que está realizando la validación de la sesión. Una vez esta petición se termine, el usuario podrá acceder a su menú principal correspondiente y visualizar sus datos.

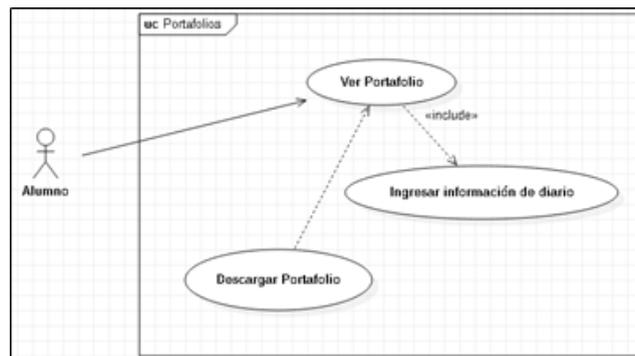
Figura 19: Caso de uso matriculación



Fuente: Elaboración propia

El alumno podrá seleccionar la asignatura que desea matricularse de una lista de asignaturas disponibles. Una vez escogida la asignatura, este coloca la clave de acceso y el sistema se encarga de guardar los cambios de esta acción, permitiendo a continuación al alumno ver la estructura del portafolio de la asignatura recién matriculado.

Figura 20: Casos de uso portafolios

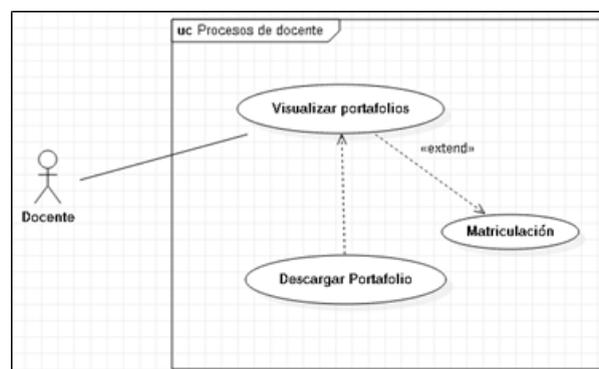


Fuente: Elaboración propia

El alumno al dar clic en Ver portafolio, además de ver la estructura del portafolio, podrá ingresar información de los diarios de clase y descargar el portafolio completo con su formato.

El docente únicamente podrá visualizar la estructura del portafolio de la asignatura de un estudiante y descargar el portafolio del mismo, como se muestra en la Figura 20.

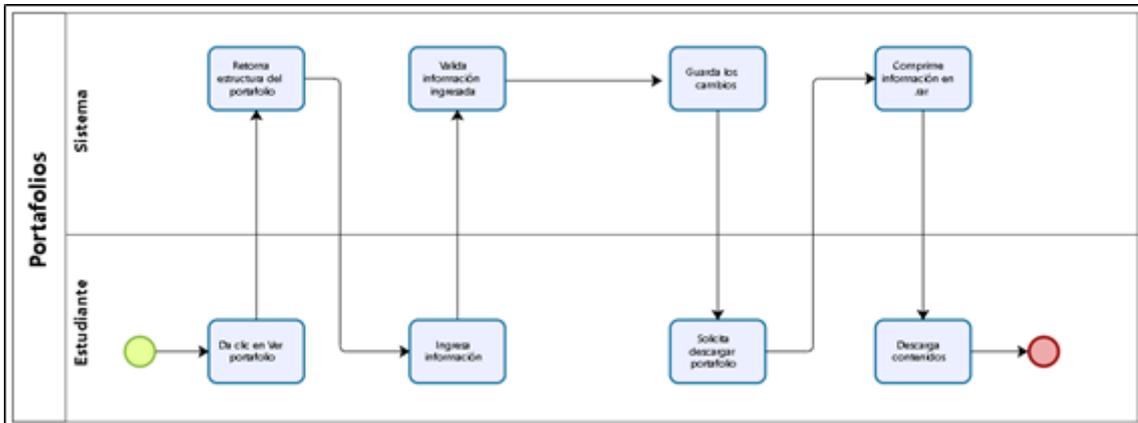
Figura 21: Caso de uso procesos de docente



Fuente: Elaboración propia

2.4.4.5.2. Diagramas de actividades

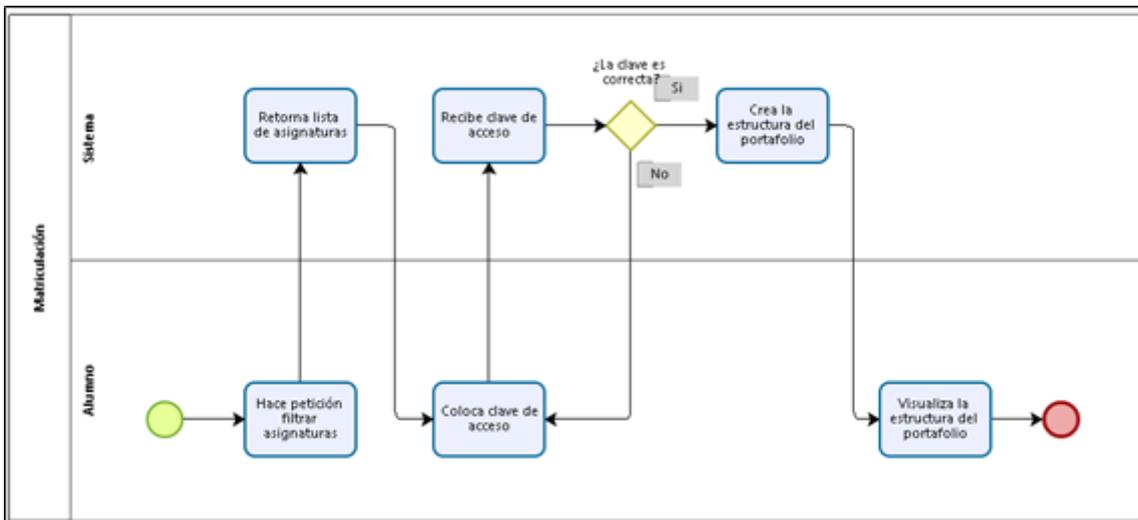
Figura 22: Diagrama de actividades portafolio



Fuente: Elaboración propia

El alumno al dar clic en Ver portafolio, además de ver la estructura del portafolio, podrá ingresar información de los diarios de clase y descargar el portafolio completo con su formato y estructura.

Figura 23: Diagrama de actividades matriculación



Fuente: Elaboración propia

El alumno al dar clic en Matricularse, escoge una asignatura y le coloca la clave de acceso, si la contraseña es la misma que la almacenada por el docente, se crea la estructura del portafolio y el alumno puede verlo.

Figura 24: Diagrama de actividades asignación de claves

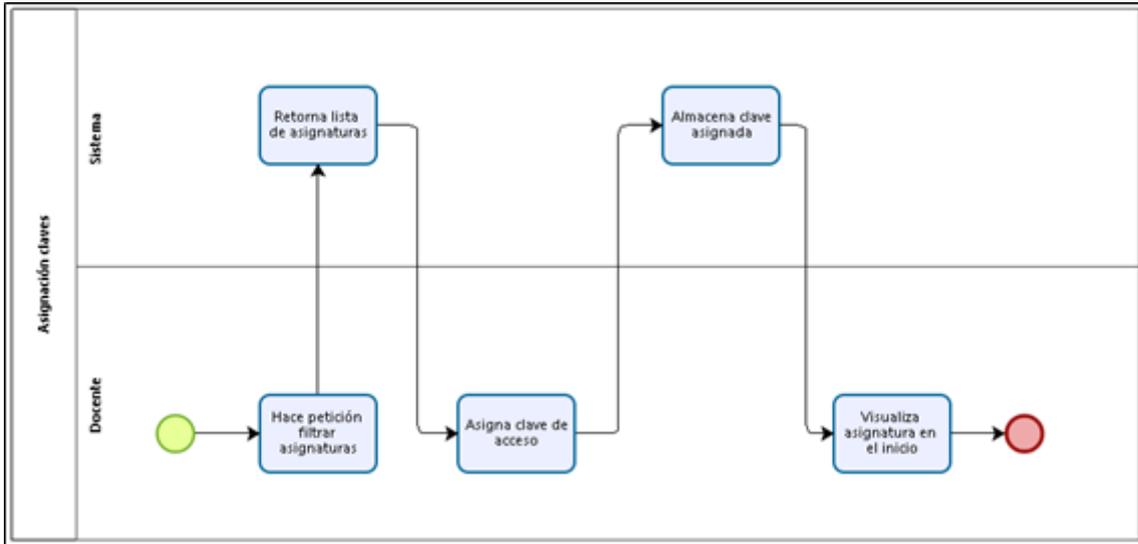
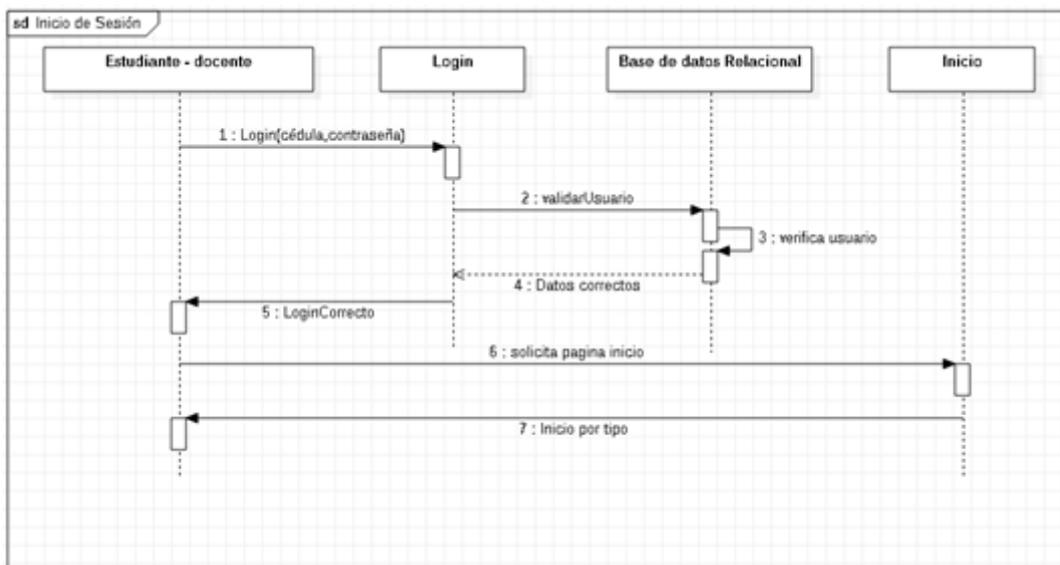


Figura 25: Diagrama de actividades asignación de claves

El docente da clic en el menú Asignación de claves, donde escogerá la asignatura y le asignará una contraseña de acceso, el sistema la almacenará y le aparecerá aquella asignatura en su pestaña de portafolios.

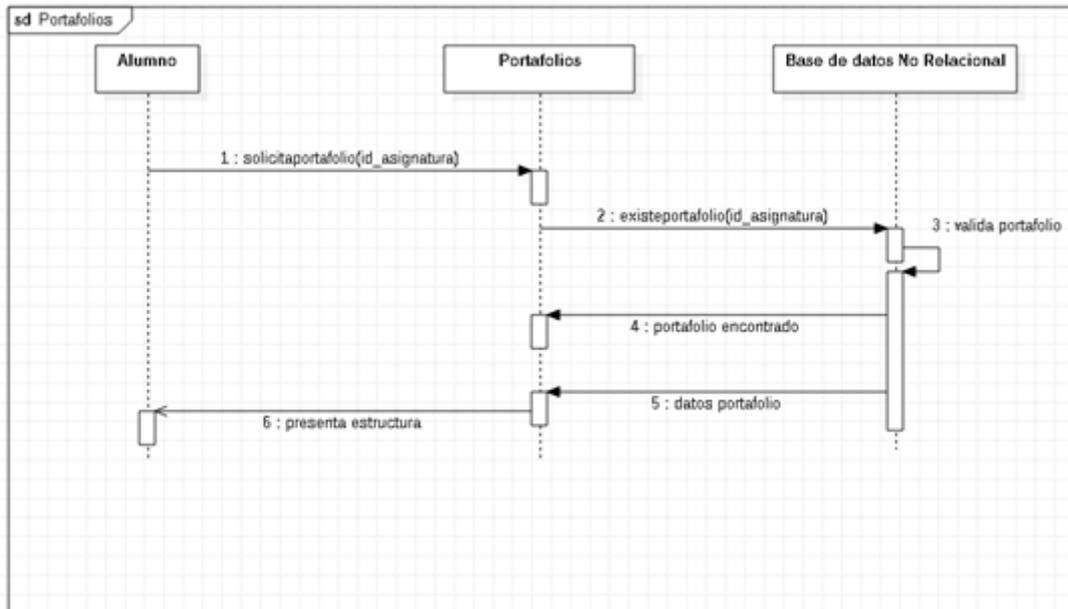
2.4.4.5.3. Diagramas de secuencias

Figura 26: Diagrama de secuencia Login



Fuente: Elaboración propia

Figura 27: Diagrama de secuencia portafolios



Fuente: Elaboración propia

2.5. EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO

El presente trabajo de titulación se basa en el desarrollo de una aplicación móvil para la gestión de portafolios académicos de la Facultad de Ingeniería Civil. Se utilizó la metodología SWIRL con el fin de llevar a cabo el proceso de desarrollo de la aplicación de una manera correcta y ágil.

A continuación, se describen los módulos funcionales más importantes de la aplicación móvil desarrollada.

2.5.1. Módulo inicio de sesión

Una vez ejecutada la aplicación se mostrará la pestaña de inicio de sesión, la cual cuenta con las opciones de ingresar cédula y contraseña del alumno o docente que ingrese a la aplicación.

Figura 29: Ensamblaje Módulo inicio de sesión



Fuente: Elaboración propia

2.5.2. Módulo portafolios

Una vez el alumno ingrese sesión al sistema, le aparecerán en el inicio sus portafolios de las asignaturas en las que está matriculado. Si no se encuentra ninguna, le aparecerá un texto que mencione que debe matricularse a alguna asignatura.

En cada asignatura/portafolio matriculado existirá el botón “Ver portafolio”, donde este redirigirá a la pestaña de estructura de dicho portafolio.

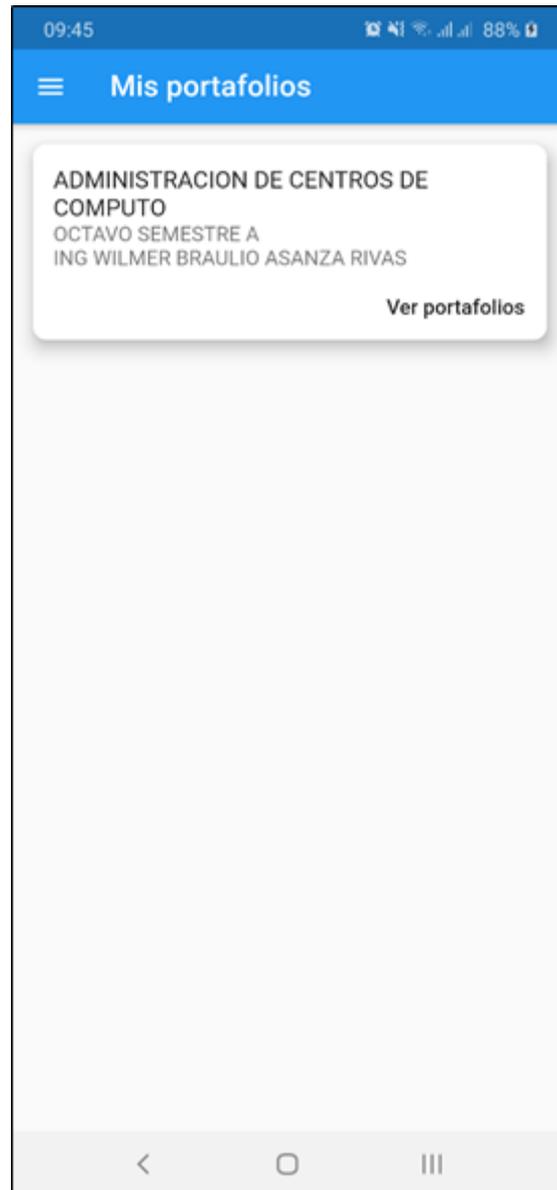
Figura 30: Ensamblaje Módulo portafolios del alumno



Fuente: Elaboración propia

El docente tendrá de igual manera una vista de los portafolios a las que le asignó una clave en el menú "Asignación de claves" donde existirá el botón "Ver portafolios".

Figura 31. Ensamblaje Módulo portafolios del docente

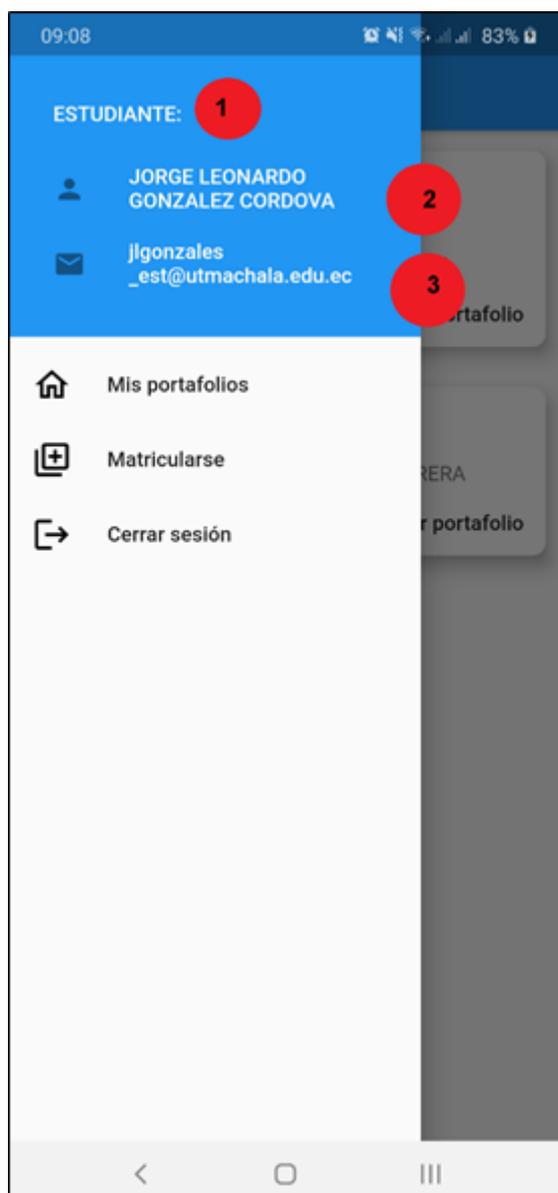


Fuente: Elaboración propia

2.5.3. Módulo de navegación

A partir de este punto se incorpora una barra de menú al lado izquierdo donde el usuario, como se muestra en las Figuras 17 y 18, podrá ver su rol (1), sus nombres y apellidos (2) y su correo institucional (3), además de los ítems de navegación correspondientes para cada usuario.

Figura 32: Ensamblaje Módulo de navegación del alumno

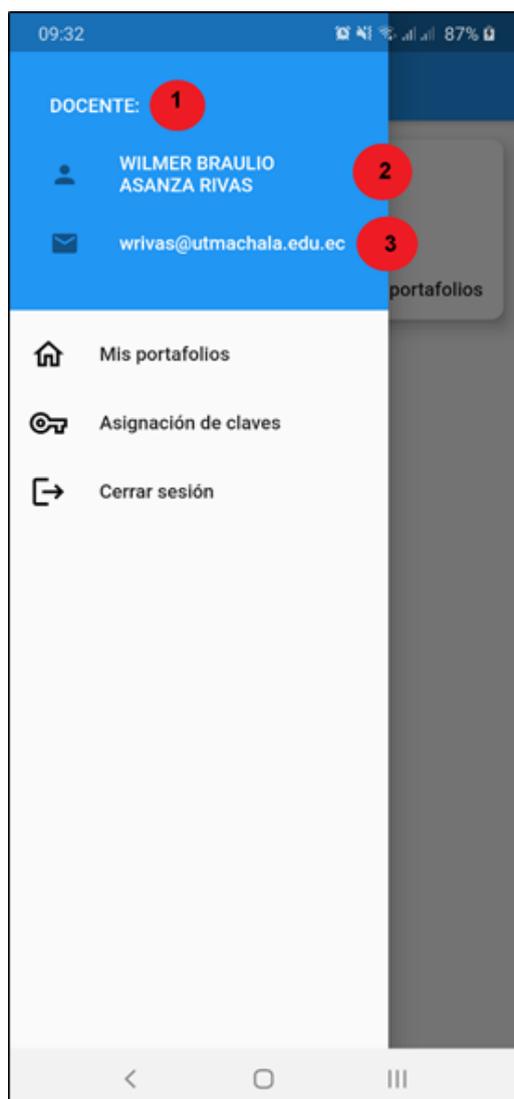


Fuente: Elaboración propia

Cuando el rol o tipo de usuario sea docente, el ítem de navegación “Matricularse” cambiará a “Asignación de claves” donde el docente podrá asignar una clave de acceso para las asignaturas que el imparte.

Una vez asignada la clave, se volverá a la pestaña de inicio que, al igual que el rol estudiante, podrá ver los portafolios que posee

Figura 33: Ensamblaje Módulo de navegación del docente



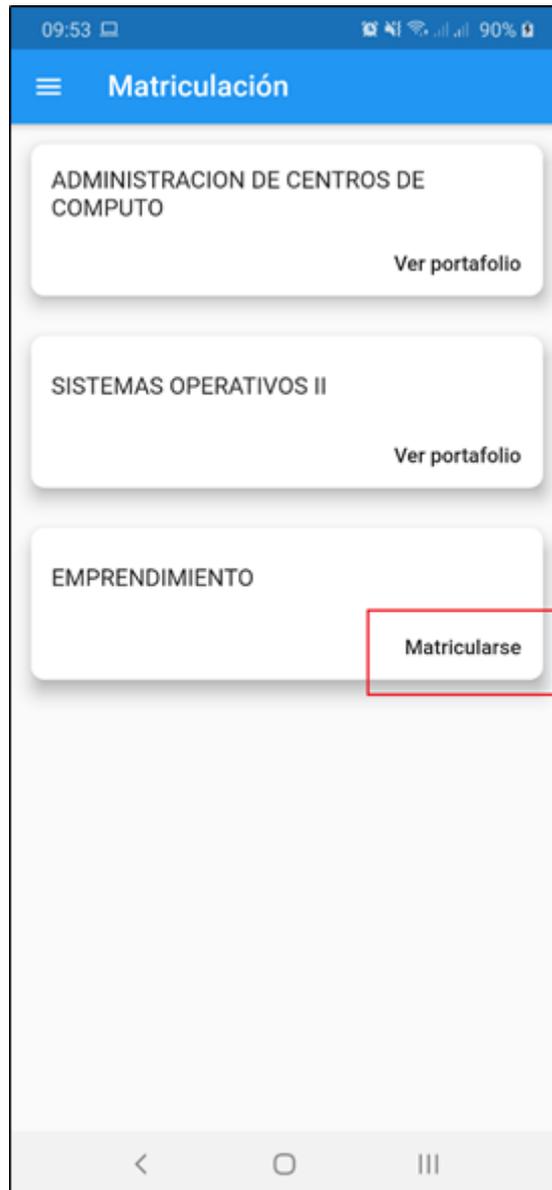
Fuente: Elaboración propia

2.5.4. Módulo matriculación

En el siguiente módulo el alumno podrá filtrar todas las asignaturas por cada carrera, donde se matricula a las que el cursa mediante una clave de acceso que anteriormente el docente asignó a dicha asignatura.

Si el alumno ya se encuentra matriculado a una asignatura del listado de asignaturas por carrera, en lugar de aparecerle el botón “Matricularse”, únicamente dirá “Ver portafolio”, como en el menú de sus portafolios.

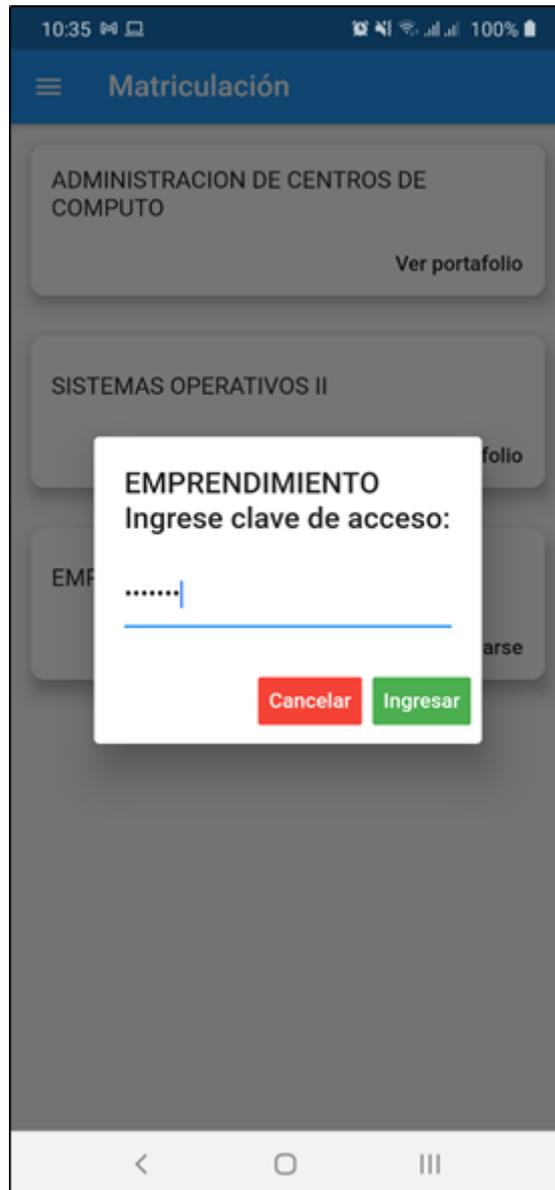
Figura 34: Módulo matriculación



Fuente: Elaboración propia

Al dar clic en matricularse, aparecerá un InputDialog donde se deberá colocar la clave de acceso y al dar clic en Ingresar, se redirigirá a la pestaña de inicio donde el alumno podrá ver sus portafolios y el último matriculado agregado.

Figura 35: Módulo matriculación colocar clave de acceso



Fuente: Elaboración propia

Al estar en la pestaña de inicio saldrá un mensaje “Matriculación exitosa” con la visualización de todos los portafolios matriculados por el estudiante y adicionalmente el ultimo agregado.

Figura 36: Módulo matriculación matrícula exitosa

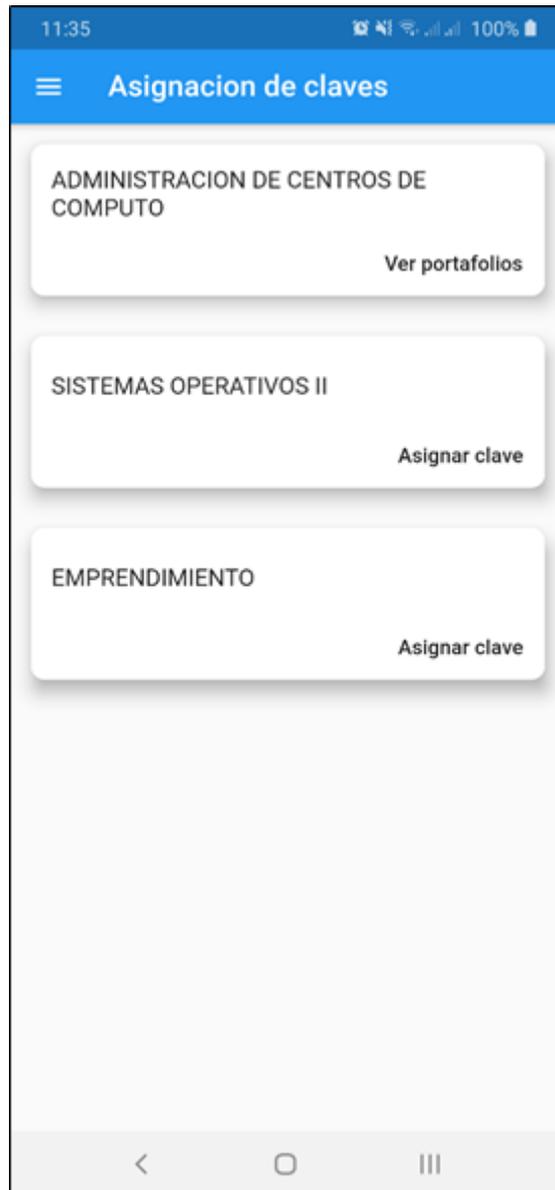


Fuente: Elaboración propia

2.5.5. Módulo asignación de clave

En el menú “Asignación de claves” del usuario docente aparecerán las asignaturas filtradas por carrera. De igual manera que el rol alumno, aquellas asignaturas que ya asignó el docente una clave de acceso, en lugar del texto del botón decir “Asignar clave”, dirá “Ver portafolios”.

Figura 37: Módulo asignación de clave

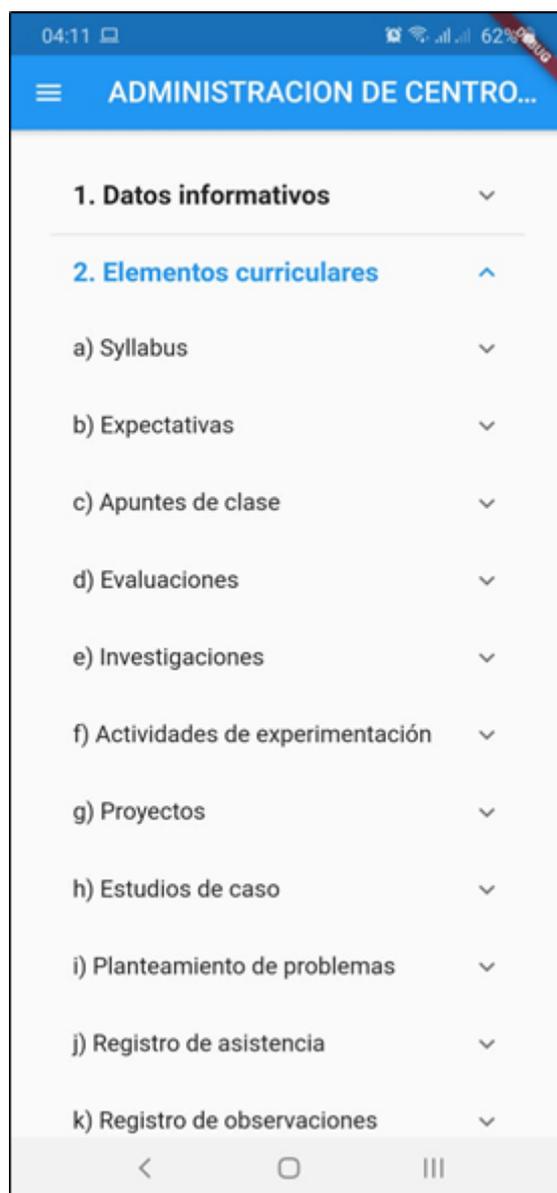


Fuente: Elaboración propia

2.5.6. Módulo estructura de portafolio

Una vez creado el portafolio, dando clic en el botón “Ver portafolio” se desplegará la pestaña de estructura del portafolio automáticamente.

Figura 38: Módulo estructura del portafolio



Fuente: Elaboración propia

2.5.7. Módulo diario metacognitivo

Dentro de la estructura, en el directorio “c) Apuntes de clase”, se generan automáticamente el número de diarios a desarrollar según las horas de clase de tal asignatura. Al dar clic en alguno de ellos se despliega la pestaña para insertar la información correspondiente al diario de clase.

Figura 39: Módulo diario metacognitivo

04:16 62%

☰ DIARIO METACOGNITIVO 1

OBJETIVO:

ACTIVIDADES DE CLASE:

ESTRATEGIAS:

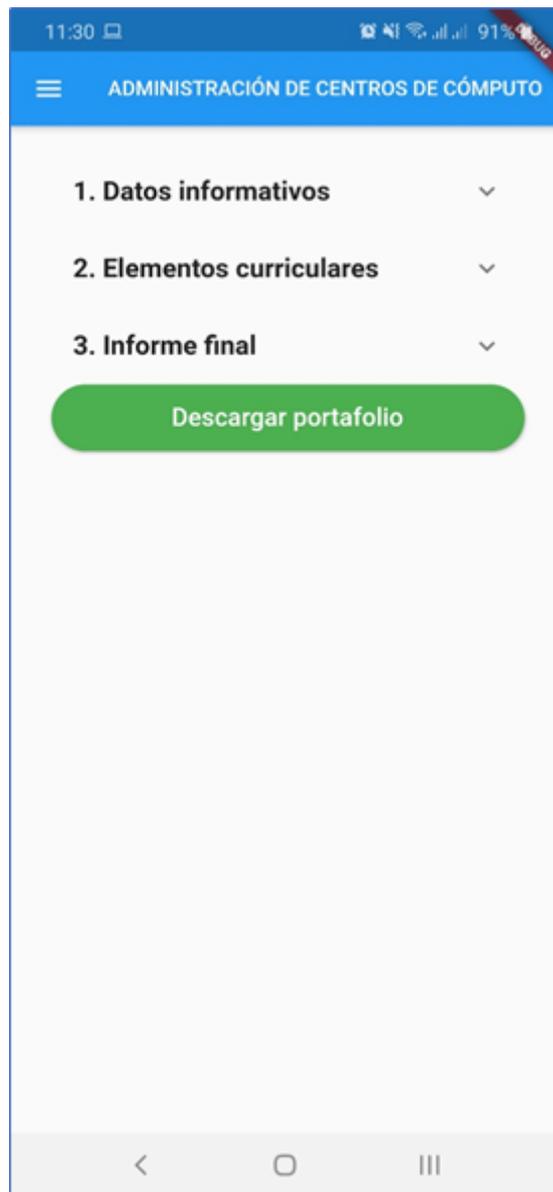
RESUMEN CONCEPTUAL:

Fuente: Elaboración propia

2.5.8. Módulo descarga del portafolio

Dentro de la estructura del portafolio, se encontrará un botón que de la opción tanto a docentes como alumnos descargar el portafolio de la asignatura. Este documento se descargará en formato .rar siendo un comprimido de todos los archivos y diarios de clases del estudiante.

Figura 39: Módulo descarga del portafolio

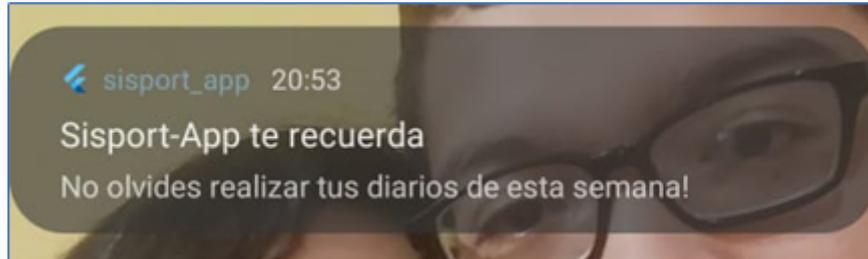


Fuente: Elaboración propia

2.5.9. Módulo notificaciones

A los usuarios de la aplicación les llegará una notificación diaria a manera de recordatorio que se deben realizar los diarios de clase de ese día en concreto.

Figura 40: Módulo notificaciones

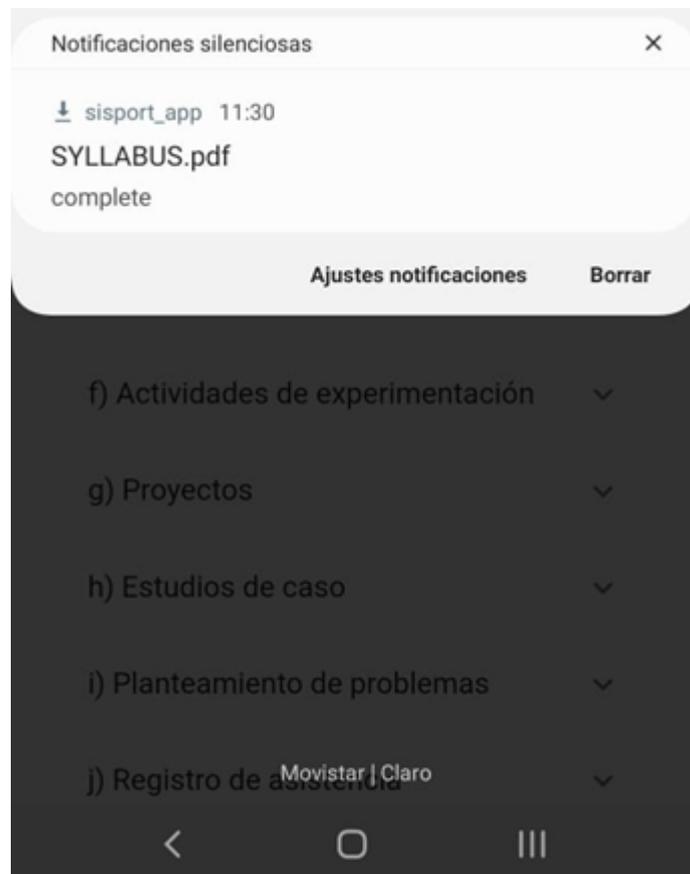


Fuente: Elaboración propia

2.5.10. Módulo descargar de archivos

Los archivos subidos por los alumnos se podrán descargar mediante la aplicación, dando clic en uno de ellos y se almacenarán en la memoria interna del dispositivo.

Figura 41: Módulo descargar archivos



Fuente: Elaboración propia

3. CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO

3.1. PLAN DE EVALUACIÓN

Para la evaluación de la aplicación “Sisport App” desarrollada a lo largo del presente trabajo, se optó por una evaluación de calidad. Esta se realiza bajo las normas y métricas de la norma ISO/IEC 9126. El objetivo de la realización de esta evaluación es verificar que las funcionalidades del sistema satisfagan los requerimientos de los usuarios. Las métricas que evalúa esta norma son las siguientes:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Operatividad

Estas a su vez están divididas en subcategorías y lineamientos, los mismos que al final de la evaluación permiten determinar la calidad del sistema.

3.2. RESULTADO DE LA EVALUACIÓN

Los resultados de la aplicación de la norma ISO/IEC 9126 como evaluación de calidad han permitido conocer el nivel de calidad del sistema bajo las métricas de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y operatividad. A continuación, se muestran los resultados obtenidos de dicho análisis mediante una escala de Likert.

Tabla 13: Resultado de evaluación de calidad

Característica	Sub-características	Criterio	Escala de Likert				
			Excelente	Bueno	Regular	Malo	Muy malo
Funcionalidad	Adecuación	El sistema cumple las necesidades del usuario	x				
	Exactitud	El sistema obtiene los resultados correctos sin errores	x				
	Interoperabilidad	Las funciones del sistema están de acuerdo al rol del usuario	x				

	Seguridad	El sistema cuenta con la seguridad correspondiente.		x			
Fiabilidad	Madurez	El sistema muestra mensajes a posibles fallos	x				
	Recuperabilidad	El sistema continúa funcionando ante cualquier inconveniente	x				
	Tolerancia a fallos	El sistema es tolerante a fallas		x			
	Cumplimiento de Fiabilidad	El sistema puede apegar a normas de fiabilidad	x				
Usabilidad	Aprendizaje	El sistema es fácil de aprender a utilizar	x				
	Comprensión	El sistema permite entender el rol de cada usuario	x				
	Operatividad	El sistema permite adaptarse rápidamente a el	x				
	Atractividad	El sistema posee una interfaz agradable	x				
Eficiencia	Comportamiento en el tiempo	El sistema posee tiempos de espera aceptables	x				
	Comportamiento de recursos	El sistema no interfiere con otros procesos		x			
Mantenibilidad	Estabilidad	El sistema se adapta a cambios y estos no interfieren en su funcionamiento		x			

	Facilidad de análisis	El sistema permite identificar fácilmente los posibles fallos	x				
	Facilidad de cambio	La implementación de cambios dentro del sistema se realiza de manera sencilla	x				
	Facilidad de pruebas	El sistema permite implementar un proceso de pruebas		x			
Portabilidad	Capacidad de instalación	El sistema es fácil de instalar	x				
	Capacidad de reemplazamiento	El sistema puede ser reemplazado por otro				x	

Fuente: Elaboración propia

A continuación, se presenta un resumen de la tabla anterior con los valores obtenidos para cada sub característica permitiendo así observar más exacto el resultado obtenido.

Tabla 14: Resumen resultado de evaluación de calidad

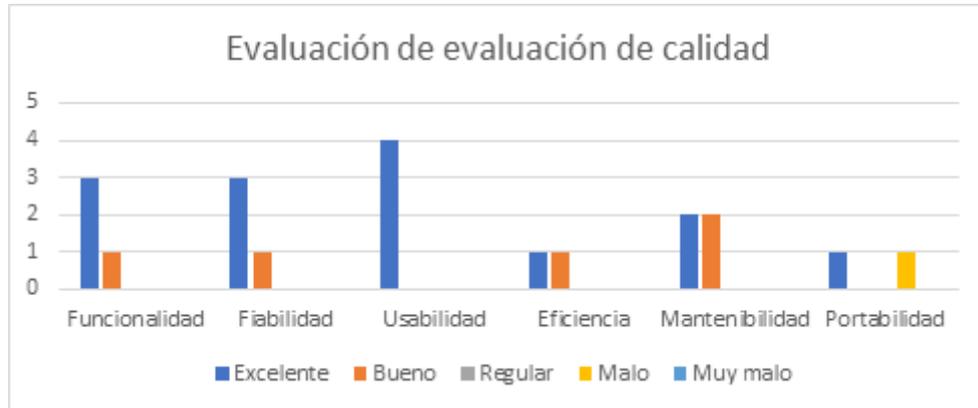
Características	Excelente	Bueno	Regular	Malo	Muy malo	Total
Funcionalidad	3	1				4
Fiabilidad	3	1				4
Usabilidad	4					4
Eficiencia	1	1				2
Mantenibilidad	2	2				4
Portabilidad	1			1		2

Fuente: Elaboración propia

Mediante la evaluación de calidad se puede observar que la mayoría de características alcanzan un nivel entre excelente y bueno, lo cual nos permite inferir que la aplicación desarrollada posee un grado de calidad bastante aceptable.

Finalmente se presenta un gráfico con los resultados obtenidos de aplicar la evaluación de calidad bajo la norma ISO/IEC1926 a la aplicación móvil para gestión de portafolios académicos.

Figura 40: Resultado de evaluación de calidad



Fuente: Elaboración propia

3.3. EVALUACIÓN CON HERRAMIENTAS

La herramienta escogida para evaluar la accesibilidad del sistema fue Axe, la cual permite mediante el estándar WCAG 2.1 verificar el cumplimiento de las normas de accesibilidad de una aplicación móvil. Los resultados obtenidos de la aplicación se muestran a continuación.

Figura 41: Resultado evaluación Axe Edit TextName

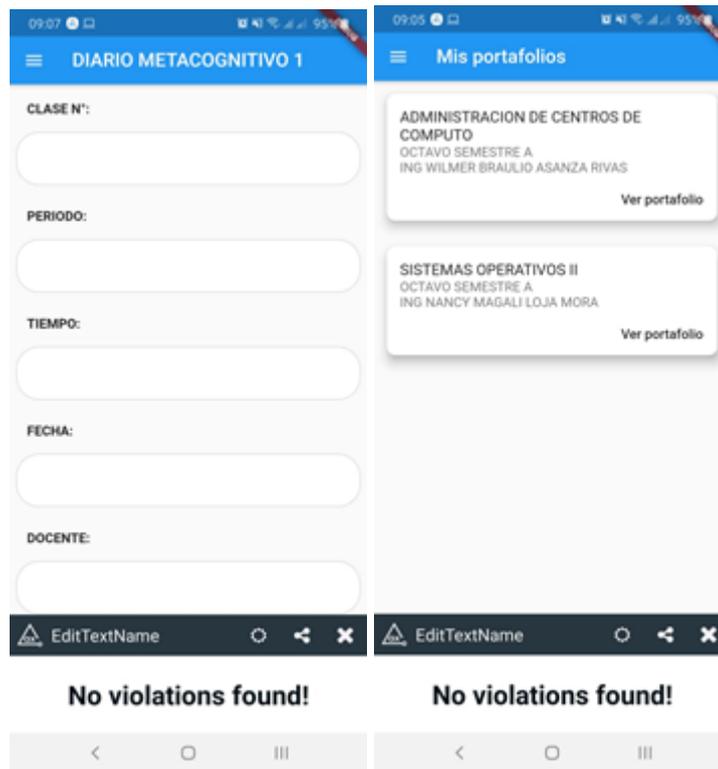


Fuente: Elaboración propia

La observación que realiza la herramienta según el estándar, aconseja que se use una etiqueta label para informar al usuario qué texto la aplicación requiere que se ingrese en un TextField, y no colocado dentro como un placeholder, tal es el caso del sistema.

La razón por la cual está colocado como placeholder es únicamente estético, para una mejor presentación del Login.

Figura 42: Resultado evaluación Axe No violations found



Fuente: Elaboración propia

Mediante la implementación de esta herramienta, se puede inferir que en cuanto a accesibilidad la aplicación móvil desarrollada cumple con los parámetros establecidos por la WCAG 2.1.

Se realizó otra evaluación de análisis dinámico en ambiente de pruebas de ejecución cuyos resultados se encuentran en los Anexos 1 y 2.

3.4. CONCLUSIONES

Finalizado el proceso de implementación de una aplicación móvil para la gestión de portafolios académicos utilizando la metodología SWIRL, se concluye que:

- Los frameworks Flutter, NodeJS y Flask poseen una correcta compatibilidad entre ellos mediante el uso de APIs, lo que permitió una correcta comunicación en las peticiones al servidor de acuerdo a los procesos funcionales de la aplicación móvil.
- La metodología de desarrollo de software SWIRL, mediante sus lineamientos y métricas, mejoraron el proceso de desarrollo de la aplicación móvil mediante su modelo iterativo que permite el surgimiento de cambios, resolución de errores y una buena comunicación con los interesados.
- Las herramientas CASE StarUML y Bizagi agilizaron el proceso de elaboración del sistema en la fase de modelado de datos, ahorrando tiempo que fue utilizado en la programación de la aplicación.
- El diseño de las interfaces del sistema fue realizado con la herramienta Balsamiq Mockups, ya que es muy completa en cuanto a estilos de diseño se refiere, lo que ayudó a tener una mejor idea en cuanto a la forma y ubicación de widgets en las pestañas de la aplicación móvil.
- El estándar ISO/IEC 9126 permitió, bajo sus métricas y lineamientos, evaluar el software con la finalidad de garantizar a los usuarios que el sistema realmente se ajusta a los requerimientos de los mismos en cuanto a calidad se refiere.

3.5. RECOMENDACIONES

- Implementar las herramientas Bizagi y StarUML para modelar los procesos del sistema, agilizando la creación de modelos UML que permitan comprender los requerimientos funcionales del software, optimizando los tiempos de desarrollo.
- Utilizar Flutter para el desarrollo de aplicaciones móviles multiplataforma elaborando una única base de código, permitiendo el desarrollo de interfaces amigables y dinámicas de manera ágil.
- Emplear la programación modular para agilizar la reutilización de componentes del sistema.
- Utilizar la función Hot Reload que brinda Flutter con el fin de acelerar las pruebas durante el entorno de desarrollo de la aplicación.
- Emplear herramientas de control de versiones de código tales como Git, que permiten generar respaldos del código de las aplicaciones asegurando los cambios del trabajo desarrollado.
- Realizar entrevistas con los interesados para recopilar sugerencias y opiniones que ayuden a satisfacer plenamente los requerimientos del sistema.
- Hacer uso de herramientas de evaluación de aplicaciones para garantizar el correcto funcionamiento del sistema y mejorarlo con las recomendaciones brindadas por las mismas.

BIBLIOGRAFÍA

- [1] M. Ledo, F. Gómez y A. Ruiz, «Software educativos/Educational softwares,» *Scielo*, vol. 1, nº 24, pp. 97-110, 2010.
- [2] J. L. Filippi, G. Lafuente y R. Bertone, «Aplicación móvil como instrumento de difusión Multiciencias,» *Redalyc*, vol. 16, nº 3, pp. 336-334, 2016.
- [3] Flutter, «Flutter,» Flutter, [En línea]. Available: <https://flutter.dev/>. [Último acceso: 5 Febrero 2021].
- [4] S. Tobón, Formación basada en competencias. Pensamiento complejo, diseño curricular y didáctica, Bogotá: ECOE, 2004.
- [5] M. Rodríguez, O. Paz, I. Sanchez y L. Pérez, «Portafolio de evaluación: una experiencia novedosa en la asignatura Introducción a la Medicina General Integral,» *Scielo*, vol. 12, nº 4, pp. 235-241, 2020.
- [6] M. Rojas y G. Juan, «El portafolio de evidencias del alumno: una estrategia didáctica de enseñanza-aprendizaje favorecedora de la evaluación clínica/The student's evidence portfolio: a teaching-learning didactic strategy that promotes clinical evaluation,» *Dialnet*, vol. 12, nº 1, p. 25, 2018.
- [7] A. Babativa, P. Briceño, C. Nieto y O. Salazar, «Desarrollo Agil de una Aplicación para Dispositivos Móviles. Caso de Estudio: Taxímetro Móvil/Agile Application Development for Mobile Devices. Case Study: Mobile Taximeter,» *Redalyc*, vol. 21, nº 3, pp. 260-275, 2016.
- [8] A. Rodríguez y L. Silva, «Arquitectura de software para el sistema de visualización médica Vismedic,» *Scielo*, vol. 8, nº 1, 2016.
- [9] V. Jimenez, W. Tello y J. Rios, «Lenguajes de Patrones de Arquitectura de Software: Una Aproximación Al Estado del Arte/Pattern Languages Software Architecture: An Approach To State of the Art,» *Redalyc*, vol. 19, nº 4, pp. 371-376, 2014.
- [10] V. Cortellessa, R. Eramo y M. Tucci, «From software architecture to analysis models and back: Model-driven refactoring aimed at availability improvement,» *ScienceDirect*, nº 127, pp. 1-28, 2020.
- [11] J. d. D. Murillo, «MEJORAMIENTO DE LA LATENCIA DE LA RED MEDIANTE EL CAMBIO DE TAMAÑO DE BÚFER PARA APLICACIONES FTP UTILIZANDO EL MODELO CLIENTE/SERVIDOR SEGÚN EL TAMAÑO PROMEDIO DE LOS ARCHIVOS A SER TRANSMITIDOS,» *Redalyc*, nº 24, pp. 74-81, 2010.
- [12] E. Salinas, N. Cerpa y P. Rojas, «Arquitectura orientada a servicios para software de apoyo para el proceso personal de software/A service oriented architecture for the implementation of the personal software process,» *Scielo*, vol. 19, nº 1, pp. 40-52, 2011.

- [13] G. Chodak, G. Suchacka y C. Yash, «HTTP-level e-commerce data based on server access logs for an online store,» *ScienceDirect*, vol. 183, 2020.
- [14] N. Palma, «Solución informática para la selección del servidor web durante la migración a código abierto/Computer solution for the selection of the web server during the migration to open source,» *Scielo*, vol. 14, nº 2, pp. 49-69, 2020.
- [15] Y. E. M. y. M. E. P. Martínez, «Sistema para la integración del proceso de normalización de bases de datos relacionales con gestores de bases de datos (SINORGES),» *Avances en Sistemas e Informática*, vol. 7, nº 3, pp. 17-26, 2010.
- [16] Y. Vazquez Ortiz, L. Mier Pierre y A. R. Sotolongo León, «Características no relacionales de PostgreSQL: incremento del rendimiento en el uso de datos JSON,» *Redalyc*, vol. 10, pp. 70-81, 2016.
- [17] A. Rodríguez Pérez, D. Rodríguez Hernández y E. Díaz Martínez, «Selección de Base de Datos No SQL para almacenamiento de Históricos en Sistemas de Supervisión,» *Redalyc*, vol. 10, nº 3, pp. 85-96, 2016.
- [18] J. Durán, E. Tandazo, M. Morales y S. Morales, «Rendimiento de bases de datos columnares/Performance of columnar database,» *Scielo*, nº 22, 2019.
- [19] G. Benigni y J. Celada, «Moomh case. Herramienta automatizada para la ayuda al desarrollo de sistemas hipermediales,» *Scielo*, vol. 23, nº 4, 2008.
- [20] R. Rodríguez, A. Rafael y J. León, «Estrategias para la indización de revistas académicas en Ciencias Aplicadas,» *Scielo*, vol. 11, nº 5, pp. 81-88, 2019.
- [21] jQuery, «jQuery,» jQuery, [En línea]. Available: <https://jquery.com/>. [Último acceso: 15 Marzo 2021].
- [22] A. Ochoa, L. Cangrejo y T. Delgado, «Alternativa Open Source en la implementación de un sistema IoT para la medición de la calidad del aire/Open Source alternative in the implementation of an IoT system for the measurement of air quality,» *Scielo*, vol. 12, nº 1, 2018.
- [23] NodeJS, «Introducción a Node.js,» NodeJS, [En línea]. Available: <https://nodejs.dev/learn>. [Último acceso: 16 Marzo 2021].
- [24] D. Coss, «Crear un API y una aplicación web con Flask,» Medium, 8 Junio 2018. [En línea]. Available: <https://blog.nearsoftjobs.com/crear-un-api-y-una-aplicacion-web-con-flask-6a76b8bf5383#:~:text=Flask%20es%20un%20micro%20framework,as%20AD%20como%20cualquier%20otra%20herramienta..> [Último acceso: 16 Marzo 2021].
- [25] S. López, «Firebase: qué es, para qué sirve, funcionalidades y ventajas,» Digital55, 17 Mayo 2020. [En línea]. Available: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>. [Último acceso: 27 Mayo 2021].

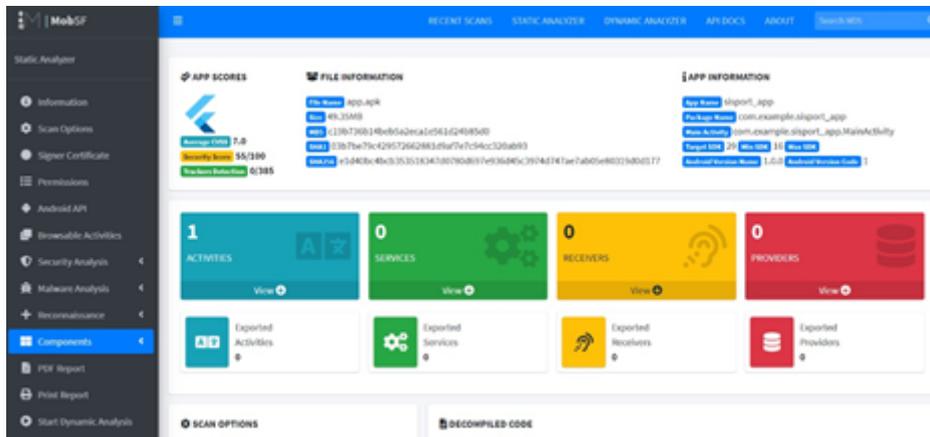
- [26] E. Vacacela, J. Tenecota, J. Torres y J. Celleri-Pacheco, «Automatización de procesos de Investigación, Vinculación, Prácticas/Pasantías Preprofesionales para universidades ecuatorianas,» *Dialnet*, vol. 19, nº 1, pp. 35-44, 2018.
- [27] F. Moreno, J. Quintero y R. Rueda, «UNA COMPARACIÓN DE RENDIMIENTO ENTRE ORACLE Y MONGODB/A PERFORMANCE COMPARISON BETWEEN ORACLE AND MONGODB,» *Scielo*, Vols. %1 de %226-1, pp. 109-129, 2016.
- [28] M. Á. Cañas C., Á. M. Bedoya Ramírez y L. Urrego Agudelo, «Modelo de requisitos para la renovación tecnológica de los sistemas de administración del,» *Redalyc*, vol. 15, nº 28, pp. 83-101, 2016.
- [29] Quest, «Toad Data Modeler,» Quest, [En línea]. Available: <https://www.quest.com/mx-es/products/toad-data-modeler/>. [Último acceso: 17 Marzo 2021].
- [30] M. Castro, L. Merchán y C. Pardo, «Implementación de métodos ágiles para la simulación de casos de uso y prototipado en el proceso de desarrollo de software/Implementing agile methods for simulating use case and prototyping in the software development process,» *Scielo*, vol. 7, nº 3, 2013.
- [31] D. Rodríguez, J. Vargas y E. González, «Arquitectura de información como un proceso para organizar sitios web usables e intuitivos: El caso del Centro Centroamericano de Población (CCP),» *Scielo*, vol. 9, nº 1, 2019.
- [32] S. Zamenfield, «StarUML una herramienta para modelado,» BrainLabs new IT, 1 Julio 2011. [En línea]. Available: <https://www.brainlabs.com.ar/novedad/staruml-una-herramienta-para-modelado/>. [Último acceso: 28 Mayo 2021].
- [33] Bizagi, «Bizagi Modeler,» Bizagi, [En línea]. Available: <https://www.bizagi.com/es/plataforma/modeler>. [Último acceso: 28 Marzo 2021].
- [34] P. Patanakul y R. Rufo-McCarron, «Transitioning to agile software development: Lessons learned from a government-contracted program,» *ScienceDirect*, vol. 29, nº 2, pp. 181-192, 2018.
- [35] E. García, O. Vite, M. Á. Navarrate, M. Á. García y V. Torres, «Metodología para el desarrollo de software multimedia educativo MEDESME/Methodology for the development of multimedia educational software MEDESME,» *Scielo*, nº 23, 2016.
- [36] R. R.-M. Peerasit Patanakul, «Transitioning to agile software development: Lessons learned from a government-contracted program,» *ScienceDirect*, vol. 29, nº 2, pp. 181-192, 2018.
- [37] M. Manimaran, A. Shanmugam, P. Parimalam, N. Murali y S. Murty, «Software development methodology for computer based I&C systems of prototype fast breeder reactor,» *ScienceDirect*, vol. 292, pp. 46-56, 2015.

- [38] J. Boaventura, E. Peña, P. Verdecia y Y. Fustiel, «Elección entre una metodología ágil y tradicional basado en técnicas de soft computing,» *Scielo*, vol. 10, pp. 145-158, 2016.
- [39] C. Boaventura, E. Peña, P. Verdecia y F. Yulia, «Elección entre una metodología ágil y tradicional basado en técnicas de soft computing/Choice between an agile and traditional methodology based on soft computing techniques,» *Scielo*, vol. 10, nº 1.1, 2016.
- [40] R. d. F. R. Julio Cesar Pereira, «Design Thinking Integrated in Agile Software Development: A Systematic Literature Review,» *ScienceDirect*, vol. 138, pp. 775-782, 2018.
- [41] C. M. Budoya, M. M. Kissaka y J. S. Mtebe, «Instructional Design Enabled Agile Method Using ADDIE Model and Feature Driven Development Process,» *ERIC*, vol. 15, nº 1, 2019.
- [42] C. Tam, E. J. d. C. Moura, T. Oliveira y J. Varajão, «The factors influencing the success of on-going agile software development projects,» *ScienceDirect*, vol. 38, nº 3, pp. 165-176, 2020.
- [43] I. Leiva y M. Villalobos, «Método ágil híbrido para desarrollar software en dispositivos móviles/Hybrid method for agile software develop mobile devices,» *Scielo*, vol. 23, nº 3, 2015.
- [44] S. Romero, A. Martín, E. López y N. Ezequiel, «Propuesta de Metodología Híbrida y Base de Documentación para el Desarrollo de Software Actual,» *Redalyc*, nº 60, 2020.
- [45] J. Molina y M. d. I. N. Pedreira, «SWIRL», METODOLOGÍA PARA EL DISEÑO Y DESARROLLO DE APLICACIONES WEB, España: 3 ciencias, 2019.
- [46] J. Molina y N. Pedreira, «Swirl,» Swirl, [En línea]. Available: <https://swirl.software/>. [Último acceso: 17 Marzo 2021].
- [47] Y. Chaviano Gómez y A. Hernández González, «HERRAMIENTAS AUTOMATIZADAS PARA LA GESTIÓN PROYECTOS,» *Redalyc*, vol. XXVII, nº 2-3, pp. 67-74, 2006.

ANEXOS

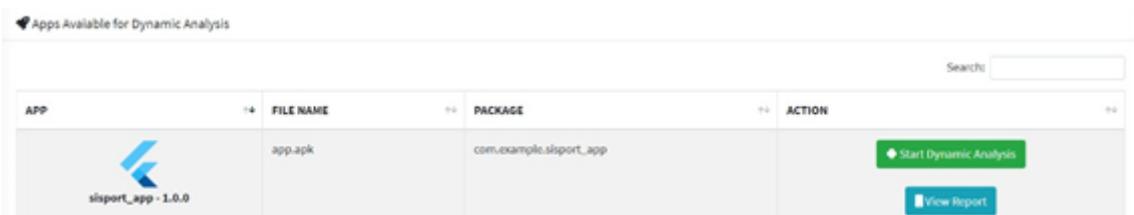
Anexo 1: Evaluación de análisis dinámico con la herramienta MobSF

Figura 43: Evaluación análisis dinámico calificación



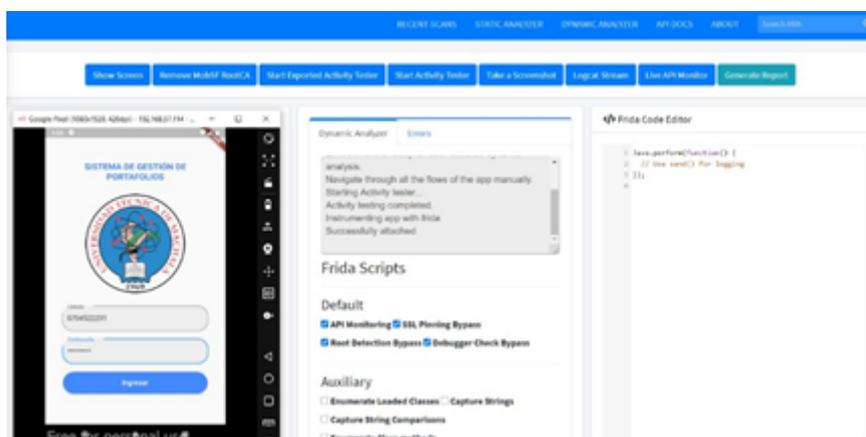
Fuente: Elaboración propia

Figura 44: Evaluación análisis dinámico opciones



Fuente: Elaboración propia

Figura 45: Evaluación análisis dinámico en ejecución



Fuente: Elaboración propia

Anexo 2: Resultado de evaluación de análisis dinámico con la herramienta MobSF

Figura 46: Evaluación análisis dinámico reporte



Fuente: Elaboración propia

Anexo 3: Modelo entrevista a alumnos de la carrera de Ingeniería de Sistemas para recolección de requerimientos

Pregunta 1: ¿Existe alguna dificultad en la realización del portafolio académico?

Pregunta 2: ¿Conoce usted el formato de portafolio de la asignatura estipulado por la Universidad Técnica de Machala?

Pregunta 3: ¿Qué proceso se hace más tedioso en la elaboración del portafolio académico?

Anexo 4: Modelo entrevista a docentes y coordinador de la carrera de Ingeniería de Sistemas para recolección de requerimientos

Pregunta 1: ¿Existe alguna dificultad en la revisión del portafolio académico?

Pregunta 2: ¿Los estudiantes siguen el formato de portafolio de la asignatura estipulado por la Universidad Técnica de Machala para la presentación del mismo?

Pregunta 3: ¿Qué proceso se hace más tedioso en la revisión del portafolio académico?

Anexo 5: Análisis de riesgos

Tabla 15: Análisis de riesgos

Número	Riesgo	Stakeholders	Detonante	Prevención
R-01	Incompatibilidad de herramientas utilizadas para el desarrollo del sistema	Equipo de desarrollo	Cambios en versiones de herramientas usadas	Investigar sobre nuevas tecnologías de desarrollo.
R-02	Daño de la arquitectura de hardware del proyecto.	Equipo de desarrollo	Cortes de luz	Tener respaldos en herramientas de versionamiento tales como Git.
R-03	Falta de conocimiento en nuevos lenguajes de programación	Equipo de desarrollo	Falta documentación de desarrollo	Investigar lenguajes que tengan suficiente documentación y una comunidad de guía.
R-04	Escasa comunicación entre los interesados del proyecto	Equipo de desarrollo Usuarios del sistema	Recolección de requerimientos escasa	Planificar reuniones para mantener el diálogo de los interesados.

Fuente: Elaboración propia