



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

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

PILALOA ZEA DARWIN JOSUE
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

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

PILALOA ZEA DARWIN JOSUE
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

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

PILALOA ZEA DARWIN JOSUE
INGENIERO DE SISTEMAS

HONORES TAPIA JOOFRE ANTONIO

MACHALA, 27 DE ABRIL DE 2021

MACHALA
2021

Proyecto de titulación Darwin Pilaloa

INFORME DE ORIGINALIDAD

9%

INDICE DE SIMILITUD

8%

FUENTES DE INTERNET

0%

PUBLICACIONES

5%

TRABAJOS DEL
ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

3%

★ Submitted to Universidad Cesar Vallejo

Trabajo del estudiante

Excluir citas

Apagado

Excluir coincidencias < 10 words

Excluir bibliografía

Activo

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

El que suscribe, PILALOA ZEA DARWIN JOSUE, en calidad de autor del siguiente trabajo escrito titulado IMPLEMENTACIÓN DE UNA APLICACIÓN WEB 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.

El autor 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.

El autor 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


PILALOA ZEA DARWIN JOSUE
0705463420

DEDICATORIA

El presente trabajo está dedicado de forma muy especial a mi abuelito José Guillermo Pilaloe Alvarado, quien en vida fue un segundo padre para mí, un hombre fuerte y cariñoso a la vez, dándome siempre su apoyo incondicional para que sea una persona de bien, lleno de valores y virtudes, recalcando lo orgulloso que se sentía de mí y de ser mi abuelito.

También lo dedico a mis padres Darwin Pilaloe Ruiz y Lorena Zea Herrera, quienes han sido mis guías a lo largo de los años, inculcando la importancia de seguir adelante y nunca rendirme hasta alcanzar mis sueños, brindándome siempre todo lo que está al alcance de sus manos para alcanzar el éxito a nivel personal y profesional.

A mis hermanos Marcel Pilaloe Zea y Elena Pilaloe Zea, y a mi perrito Max Pilaloe, ya que son el motivo por el cual intento ser cada día mejor.

Y Finalmente, a mis tíos, abuelos, amigos, a la persona que amé y se fue a otro país, a mis compañeros y toda mi familia en general, que estuvieron siempre acompañándome durante mi formación brindándome consejos, cariño y bendiciones para seguir adelante.

Sr. Pilaloe Zea Darwin Josue

AGRADECIMIENTO

En primer lugar, agradezco a Dios por haberme permitido culminar con éxito mi etapa profesional, a mis padres por su apoyo incondicional, dándome siempre consejos, guiándome con la inteligencia y el ejemplo en todo momento.

También agradezco a mis amigos y compañeros, que han estado presentes en todo momento de distintas formas y en varios acontecimientos, con los cuales he compartido logros, fracasos de la vida y que incluso aportaron con ideas sólidas para la elaboración de mi proyecto de titulación.

Finalmente agradezco a la Universidad Técnica de Machala, por la calidad de docentes que imparten conocimientos para poder formar grandes profesionales en el área de la informática y a mi tutor, el Ing. Joofre Honores Tapia, quien con su paciencia y conocimientos fue guía durante la elaboración de mi proyecto de titulación.

Sr. Pilaloe Zea Darwin Josue

RESUMEN

En el Ecuador el diseño e implementación de las plataformas web en el ámbito educativo ha tenido un enorme crecimiento en los últimos años, y cada vez son más las universidades que optan por automatizar sus procesos académicos, incluidos en el plan de educación superior. A pesar de que existen múltiples herramientas de desarrollo que facilitan la creación de sitios web dinámicos, se requiere de sistemas expertos capaces de centralizar las evidencias académicas de los estudiantes durante un determinado periodo, para lo cual se plantean distintos diseños de software específicos. Actualmente la Universidad Técnica de Machala posee plataforma web “MOODLE” para el entorno dinámico de aprendizaje llamado “EVEA UTMACH”, destinada a la gestión de los recursos académicos, tales como: trabajos autónomos, talleres en clase, evaluaciones, material didáctico, etc., los cuales son analizados por el docente de forma virtual y subida calificación la respectiva al sistema educativo durante un determinado tiempo; pero una vez concluido el período académico, toda esta información no se encuentra estructurada siguiendo el formato o modelo de “Portafolio académico” planteado por la universidad. Por tal motivo, el estudiante usualmente registra todas las evidencias de trabajo de forma manual y envían la información directamente a los correos institucionales de los docentes, es decir, no existe un control de formatos de documentos ni de evidencias académicas, a pesar de la existencia del modelo organizacional llamado “Portafolio de la asignatura”, resultando un proceso tedioso tanto para el estudiante a nivel de organización de la información, y al docente durante el proceso calificativo; Por lo tanto, se diseñó un sistema capaz de estructurar los portafolios académicos por cada estudiante, el cual se anexan: archivos, documentos, imágenes, biografía estudiantil e información sobre las actividades diarias de clase, de tal forma que al finalizar el semestre o período académico el sistema genera las carpetas, archivos y documentos en formatos disponibles para toda la suite ofimática, los cuales son comprimidos en un archivo de tipo “.rar” y de esta forma el portafolio de clases de la asignatura correspondiente a un período académico pueda ser descargado en cualquier momento para propósitos de evaluación o retroalimentación académica. Para lograr determinar el alcance y éxito del proyecto se analizó la problemática planteada desde diferentes puntos de vistas, se recopiló la información a nivel de requisitos y se presentó la propuesta tecnológica partiendo de la implementación de

una aplicación web para la gestión de portafolios académicos, con fundamentación de la metodología de desarrollo ágil “SWIRL” para la Facultad de Ingeniería Civil de la Universidad Técnica de Machala, la cual posee las fases necesarias para un desarrollo web de calidad, utilizando Frameworks en tendencia de desarrollo web como: ReactJS, NodeJS y Flask; utilizando como base de datos a PostgreSQL y MongoDB; con arquitectura de tipo cliente-servidor, permitiendo el manejo preciso y eficaz de la información almacenada en los portafolios de clases, además en el apartado final se realizó la respectiva evaluación del sistema, siguiendo el estándar para la evaluación de calidad de software ISO/IEC-9126, logrando de esta forma un aporte al desarrollo tecnológico de la universidad.

Palabras claves: SWIRL, ReactJS, NodeJS, Flask, Gestión de Portafolios, Desarrollo web.

ABSTRACT

In Ecuador, the design and implementation of web platforms in the educational field has had enormous growth in recent years, and more and more universities are opting to automate their academic processes, included in the higher education plan. Despite the fact that there are multiple development tools that facilitate the creation of dynamic websites, expert systems are required capable of centralizing the academic evidence of students during a certain period, for which different specific software designs are proposed. Currently, the Technical University of Machala has a "MOODLE" web platform for the dynamic learning environment called "EVEA UTMACH", aimed at managing academic resources, such as: autonomous work, class workshops, evaluations, teaching materials, etc. , which are analyzed by the teacher in a virtual way and the respective qualification is raised to the educational system during a certain time; but once the academic period is over, all this information is not structured following the format or model of "Academic Portfolio" proposed by the university. For this reason, the student usually registers all the work evidences manually and sends the information directly to the institutional emails of the teachers, that is, there is no control of document formats or academic evidence, despite the existence of the organizational model called "Portfolio of the subject", resulting in a tedious process both for the student at the information organization level, and for the teacher during the qualifying process; Therefore, a system was designed capable of structuring the academic portfolios for each student, which are attached: files, documents, images, student biography and information on daily class activities, in such a way that at the end of the semester or period academic the system generates folders, files and documents in formats available for the entire office suite, which are compressed in a file of type ".rar" and in this way the portfolio of classes of the subject corresponding to an academic period can be downloaded at any time for academic feedback or evaluation purposes. In order to determine the scope and success of the project, the problem raised from different points of view was analyzed, the information was compiled at the requirements level and the technological proposal was presented based on the implementation of a web application for the management of academic portfolios, with Foundation of the agile development methodology "SWIRL" for the Faculty of Civil Engineering of the Technical University of Machala, which has the necessary phases for quality web

development, using Frameworks in trend of web development such as: ReactJS, NodeJS and Flask ; using PostgreSQL and MongoDB as databases; with client-server type architecture, allowing the precise and efficient management of the information stored in the class portfolios, in addition, in the final section, the respective evaluation of the system was carried out, following the standard for the evaluation of ISO/IEC-9126 software quality, thus achieving a contribution to the technological development of the university.

Keywords: SWIRL, ReactJS, NodeJS, Flask, Portfolio Management, Web Development

INTRODUCCIÓN

La industria del desarrollo de aplicaciones web para la educación superior ha aumentado drásticamente por la tendencia hacia la actualización y optimización de las plataformas de enseñanza aprendizaje, donde frecuentemente se requieren de la implementación sistemas expertos capaces de gestionar el proceso educativo desde el cumplimiento del plan académico establecido en el syllabus de clases, hasta la recopilación de evidencias por parte de los estudiantes. Pero uno de los factores que triplicó la creación de dichas plataformas fue el confinamiento iniciado en el año 2020, causado por la enfermedad “COVID-19”, donde se produjo el cierre masivo de las actividades presenciales de las instituciones educativas a nivel: básico, medio y superior. Sin embargo, la educación no puede detenerse y sigue siendo un derecho fundamental de los ciudadanos, por lo tanto, se crearon estrategias de enseñanzas que incluyen el acceso a la “Teleeducación” [1], permitiendo el acceso a diversos materiales educativos alojados en la web, convirtiéndose en un recurso fundamental en todo sistema de educación [2].

Por ello, la Universidad Técnica de Machala actualmente cuenta con una plataforma web “MOODLE” para el entorno dinámico de aprendizaje llamado “EVEA UTMACH”, en donde los estudiantes pueden acceder a sus clases virtuales, subir actividades autónomas e intraclases, completar evaluaciones y ver las calificaciones por cada reactivo planteado por el docente. Dicha plataforma solo contiene información del semestre en curso, es decir, la información generada es eliminada por cada nuevo período académico y además de que contiene una estructura diversificada, es decir, a preferencia de cada docente, dificultando la organización generalizada de la información.

Cabe recalcar que la universidad posee un modelo para la organización de la información y el registro de evidencias de cada asignatura durante un período académico, conocido como: “Portafolio de la Asignatura”, pero muchas de las veces, no se cumple en detalle el formato estandarizado de presentación de documentos, evidencias de clases, e incluso en la propia estructura propuesta, sin embargo, esta información es recopilada por el estudiante de forma manual, para luego proceder a digitalizar y enviarla al correo institucional de cada docente, representando una

carga excesiva de revisión de documentos, debido a la cantidad de errores y falta de información presentada en el resumen de actividades diarias de clase.

Debido a la problemática existente, surge la necesidad de la creación de una aplicación web que administre los portafolios académicos de las distintas asignaturas de clases, con la utilización de herramientas de programación web del lado del servidor, tales como: NodeJS y Flask; y del lado del cliente: ReactJS, jQuery y Bootstrap, trabajando en conjunto con las bases de datos: PostgreSQL y MongoDB, usando la metodología de desarrollo ágil “SWIRL”, siguiendo el proceso de modelado, planificación, análisis, pruebas e implementación, para la elaboración de un software de calidad [3]. Teniendo como objetivo principal la gestión de los portafolios académicos de las distintas asignaturas de clases impartidas en la Facultad de Ingeniería Civil de la Universidad Técnica de Machala.

A continuación, se detalla una estructura por capítulos del proyecto tecnológico propuesto:

Capítulo I: En esta primera sección se detalla la situación problemática del proyecto, tanto en análisis del caso de estudio presentado, como en el diseño y la propuesta para la solución del mismo, por lo tanto, se recolecta la información de forma detallada, elaborando un listado de requisitos y con la respectiva fundamentación de la selección de herramientas y tecnologías a implementar.

Capítulo II: En esta segunda sección se elaboran los diagramas y modelos necesarios para la creación del prototipo a implementar; además se define el diseño y visualización de las interfaces, para finalmente realizar las pruebas de ejecución del sistema web planteado.

Capítulo III: En esta sección final se recopila información de todos los resultados de las pruebas realizadas al sistema planteado, siguiendo el estándar internacional de calidad ISO/IEC 9126, determinando la eficacia y eficiencia del sistema, con las respectivas recomendaciones y conclusiones obtenidas del proyecto total.

CONTENIDO

DEDICATORIA.....	1
AGRADECIMIENTO.....	2
RESUMEN.....	3
ABSTRACT.....	5
INTRODUCCIÓN.....	7
1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUISITOS.	13
1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.....	13
1.2. ESTABLECIMIENTO DE REQUISITOS.....	14
1.3. JUSTIFICACIÓN DE REQUERIMIENTOS.....	15
2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO.....	16
2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.....	16
2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.....	20
2.3. OBJETIVOS DEL PROTOTIPO.....	27
2.4. DISEÑO DEL PROTOTIPO.....	28
2.5. EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO.....	68
3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO.....	77
3.1. PLAN DE EVALUACIÓN.....	77
3.2. RESULTADOS DE EVALUACIÓN.....	77
3.3. CONCLUSIONES.....	81
3.4. RECOMENDACIONES.....	82
BIBLIOGRAFÍA.....	83
ANEXOS.....	86
ANEXO A: ESTRUCTURA DEL PORTAFOLIO ACADÉMICO – UTMACH.....	86
ANEXO B: FORMATO DE DISEÑO DE ENTREVISTA.....	86
ANEXO C: ANÁLISIS DE RIESGOS.....	87

ÍNDICE DE TABLAS

Tabla 1: Funcionalidades del sistema.....	28
Tabla 2: Limitaciones del sistema.....	29
Tabla 3: Factibilidad operativa del sistema.....	30
Tabla 4: Factibilidad técnica del sistema.....	31
Tabla 5: Factibilidad económica del sistema.....	32
Tabla 6: Matriz de Interesados o Stakeholders.....	33
Tabla 7: Requerimientos funcionales del sistema.....	33
Tabla 8: Requerimientos no funcionales del sistema.....	34
Tabla 9: Historia de usuario N1 - Inicio de sesión - RF01.....	35
Tabla 10: Historia de usuario N2 - Clave de matriculación RF-02.....	36
Tabla 11: Historia de usuario N3 - Proceso de matriculación - RF03.....	37
Tabla 12: Historia de usuario N4 - Visualización del portafolio - RF04.....	37
Tabla 13: Historia de usuario N5 - Diarios de clase - RF05.....	38
Tabla 14: Historia de usuario N6 - Subida de archivos - RF06.....	39
Tabla 15: Historia de usuario N7 - Descarga de portafolio - RF07.....	39
Tabla 16: Historia de usuario N8 - Gestión de estudiantes - RF08.....	40
Tabla 17: Historia de usuario N9 - Perfil de usuarios - RF09.....	41
Tabla 18: Historia de usuario N10 - Gestión de docentes – RF10.....	41
Tabla 19: Historia de usuario N11 - Gestión de asignaturas - RF11.....	42
Tabla 20: Evaluación de calidad ISO/IEC 9126.....	77
Tabla 21: Evaluación de calidad - usabilidad y portabilidad.....	80
Tabla 22: Gestión de Riesgos.....	87

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Arquitectura de Hardware - SISPORT.....	16
Ilustración 2: Modelo de Tres capas.....	18
Ilustración 3: Fundamentación teórica.....	20
Ilustración 4: Ciclo de vida SWIRL.....	21
Ilustración 5: Cronograma de actividades.....	43
Ilustración 6: Modelo de Base de Datos Relacional.....	45
Ilustración 7: Modelo de Base de Datos No Relacional.....	47
Ilustración 8: Diseño navegacional - Estudiante.....	48
Ilustración 9: Diseño navegacional - Docente.....	49
Ilustración 10: Diseño navegacional - Coordinador.....	49
Ilustración 11: Diseño de inicio del sitio.....	50
Ilustración 12: Diseño de Inicio de Sesión.....	51
Ilustración 13: Diseño del menú principal del estudiante.....	51
Ilustración 14: Diseño de portafolios del estudiante.....	52
Ilustración 15: Diseño del Portafolio del estudiante.....	52
Ilustración 16: Diseño de Perfil de usuario.....	53
Ilustración 17: Diseño de vista de portafolios del docente.....	53
Ilustración 18: Diseño de activación de asignaturas por el docente.....	54
Ilustración 19: Diseño de portafolios por asignatura del docente.....	55
Ilustración 20: Diseño de portafolio del estudiante desde el docente.....	55
Ilustración 21: Diseño de gestión de estudiantes.....	56
Ilustración 22: Diseño de creación de estudiantes.....	56
Ilustración 23: Diseño de gestión de docentes.....	57
Ilustración 24: Diseño de creación de docentes.....	57
Ilustración 25: Diseño de gestión de asignaturas.....	58
Ilustración 26: Diseño de creación de asignaturas.....	58
Ilustración 27: Diseño del portafolio del estudiante desde el coordinador.....	59
Ilustración 28: Diagrama de caso de uso - General.....	60
Ilustración 29: Diagrama de caso de uso - Inicio de sesión.....	61
Ilustración 30: Diagrama de caso de uso – Visualización de portafolios.....	61
Ilustración 31: Diagrama de caso de uso - Matriculación.....	62
Ilustración 32: Diagrama de caso de uso - Docente.....	62
Ilustración 33: Diagrama de caso de uso - Coordinador.....	63

Ilustración 34: Diagrama de actividades - Inicio de Sesión.....	64
Ilustración 35: Diagrama de actividades - Matriculación.....	65
Ilustración 36: Diagrama de actividades - Portafolios.....	65
Ilustración 37: Diagrama de actividades - Asignaturas.....	66
Ilustración 38: Diagrama de secuencia - Inicio de sesión.....	66
Ilustración 39: Diagrama de secuencia - Portafolios.....	67
Ilustración 40: Diagrama de secuencia - Asignaturas.....	67
Ilustración 41: Implementación - Página de bienvenida.....	68
Ilustración 42: Implementación – Inicio de sesión.....	69
Ilustración 43: Implementación - Menú principal del estudiante.....	69
Ilustración 44: Implementación – Menú portafolios del estudiante.....	70
Ilustración 45: Implementación - Portafolio de estudiante.....	70
Ilustración 46: Implementación - Diarios metacognitivos.....	71
Ilustración 47: Implementación - Subir archivos al portafolio.....	71
Ilustración 48: Implementación - Validación de subida de archivos.....	72
Ilustración 49: Implementación - Comprobación de archivos.....	72
Ilustración 50: Implementación - Perfil estudiante.....	73
Ilustración 51: Implementación – Menú portafolios de docentes.....	73
Ilustración 52: Implementación - Portafolio desde la vista del docente.....	74
Ilustración 53: Implementación - Gestión de estudiantes.....	74
Ilustración 54: Implementación - Gestión de docentes.....	75
Ilustración 55: Implementación - Gestión de asignaturas.....	75
Ilustración 56: Implementación - Portafolio desde la vista del coordinador.....	76
Ilustración 57: Evaluación de seguridad - Run FAE.....	78
Ilustración 58: Evaluación de seguridad - W3C Validator.....	78
Ilustración 59: Evaluación de seguridad - GTmetrix.....	79
Ilustración 60: Diagrama de barras - usabilidad y portabilidad.....	80
Ilustración 61: Estructura del portafolio UTMACH.....	86
Ilustración 62: Formato de entrevista.....	86

1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUISITOS.

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

En la Facultad de Ingeniería Civil de la Universidad Técnica de Machala existen cuatro carreras ligadas al desarrollo y progreso de la provincia de El Oro y el Ecuador en general, dentro de las ingenierías que se ofertan tenemos: Ingeniería de Sistemas, Ingeniería en Tecnologías de la Información, Ingeniería Civil e Ingeniería Ambiental, las cuales dentro del perfil académico es crear profesionales altamente capacitados, competitivos, con altos niveles de experiencia y desenvolvimiento en el mundo laboral.

Dentro del desarrollo del proceso académico, los estudiantes elaboran un portafolio con la finalidad de registrar diariamente las actividades realizadas en cada asignatura impartida, siguiendo el modelo de registro de competencias dentro del enfoque socioformativo, durante un tiempo determinado, de esta forma las autoridades educativas puedan evidenciar de forma didáctica, el trabajo desarrollado por su respectivo docente [4].

El portafolio contiene información detallada sobre: syllabus académicos, actividades autónomas, talleres, diarios de clases, casos de estudios que incorporan: planos, codificación de programas, manuales, además de las investigaciones, evaluaciones, entre otros, siguiendo el objetivo de la asignatura [5].

Todas estas evidencias académicas deben seguir el modelo de “Portafolio Académico” planteado por la Universidad (**ver anexo A**), que surgió con base a la coordinación entre directivos académicos para la estandarización del documento final de presentación estudiantil.

Sin embargo, la universidad no cuenta con una plataforma o aplicación web que gestione la información académica generada por cada estudiante, por lo que actualmente el docente de forma manual revisa cada portafolio estudiantil, para luego proceder a elegir el más detallado o completo y subirlo al drive académico, resultando un proceso tedioso a partir de la verificación de los formatos de diarios de clases, fechas y el plan de clase establecido en

el syllabus académico, convirtiéndolo en un actividad exhaustiva y con falta de dinamismo.

Como trabajo de titulación propuesto se busca satisfacer la problemática planteada, a través del diseño e implementación de una aplicación alojada en la web que contengan diferentes herramientas de desarrollo ágiles y potentes para el entorno web, utilizando como base la metodología de desarrollo "SWIRL".

1.2. ESTABLECIMIENTO DE REQUISITOS

En la Facultad de Ingeniería Civil existen departamentos administrativos, docentes y de coordinación, con los cuales se efectuó un diálogo sobre el proceso actual para la gestión de portafolios académicos y los recursos informáticos disponibles para la implementación de la plataforma web.

Adicionalmente se entrevistó a los respectivos presidentes de curso y estudiantes en general de la carrera de Ingeniería de Sistemas y Tecnologías de la Información, sobre las funciones y actividades que deberían realizarse en el sistema, con base a su nivel de experiencia en el ámbito de la informática.

Con el diálogo realizado, se aplicó el proceso de la ingeniería de requerimientos con todos los posibles interesados del proyecto [6], obteniendo los siguientes requisitos o funcionalidades:

- Gestión de estudiantes y asociación con las asignaturas correspondientes a través de una clave de acceso al portafolio.
- Creación del portafolio estructurado por carpetas, subir archivos y documentos en múltiples formatos.
- Generación de los diarios de clase con base al plan de clase establecido en el syllabus académico.
- Permitir descargar el portafolio completo.

Entre los requisitos complementarios cabe recalcar la gestión de roles y permisos para los coordinadores de carrera, estudiantes y docentes;

adicionalmente los requisitos funcionales y no funcionales serán detallados en la fase de análisis de la metodología a implementar.

A pesar que el proyecto está pensado para la Facultad de Ingeniería Civil, el sistema web tiene la visión de ser implementado a nivel general, en todas las carreras existentes en la Universidad Técnica de Machala, sin afectar el rendimiento y el proceso académico del mismo.

1.3. JUSTIFICACIÓN DE REQUERIMIENTOS

El sistema web para la gestión de portafolios, tiene como propósito la optimización de los procesos a través de la automatización de la recolección de evidencias académicas, durante todo el período académico estudiantil, debido a que resulta un proceso exhaustivo y una carga de trabajo excesiva al momento de evaluar y desarrollar cada actividad académica por estudiante.

Debido al nivel alto de información que se genera constantemente por asignatura y estudiantes, se requiere de un sistema robusto, centralizado y con una arquitectura de tipo “cliente – servidor”, a través de una infraestructura con mecanismos de direccionamiento y transporte de la información [7], capaz de soportar la afluencia de solicitudes diarias, con una conexión constante e independientemente del medio o dispositivo que se conecte para poder consumir todos los servicios de la plataforma.

Dentro de los roles y permisos, existen tres tipos o niveles de perfiles de usuarios, los cuales son: (Nivel I) Coordinador, (Nivel II) Docente y (Nivel III) estudiante, lo que hace al sistema totalmente administrable.

Todos los requisitos planteados serán desarrollados aplicando las tendencias en las herramientas de desarrollo web tanto en “Back-end” como en “Front-end”, siguiendo la metodología ágil de desarrollo web “SWIRL” debido a sus fases muy bien definidas para elaborar software de calidad por medio de la planificación, desarrollo y la ejecución de las pruebas.

2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO

A diferencia de un sitio web común o de gestión de información, el presente proyecto tecnológico posee una estructura específica o entorno de ejecución, tanto para el acceso, visualización y distribución de la información entre múltiples servidores con distintos lenguajes de programación y herramientas de conexión, por lo tanto, se detalla el sistema interno para el funcionamiento del aplicativo web.

2.1.1. Arquitectura de hardware

La arquitectura de hardware se basa en ver el estado arquitectónico del proyecto, es decir, los equipos físicos e infraestructura necesaria para soportar las aplicaciones desarrolladas, aquí se detalla la cantidad de servidores, tipo de conexión, almacenamiento y velocidad necesaria para el correcto funcionamiento.

2.1.2. Arquitectura Cliente – Servidor

El modelo Cliente – Servidor es muy popular, y es uno de los más implementados en sistemas de información web, debido a que se dividen los procesos en múltiples servidores y a su vez cooperan entre sí, de esta forma, a través, de una infraestructura de comunicación, se puedan enviar datos por medio de las “solicitudes” del lado del cliente y estar a la espera de las “respuestas” del lado del servidor, sobre la operación solicitada [8].

2.1.3. Arquitectura a implementar

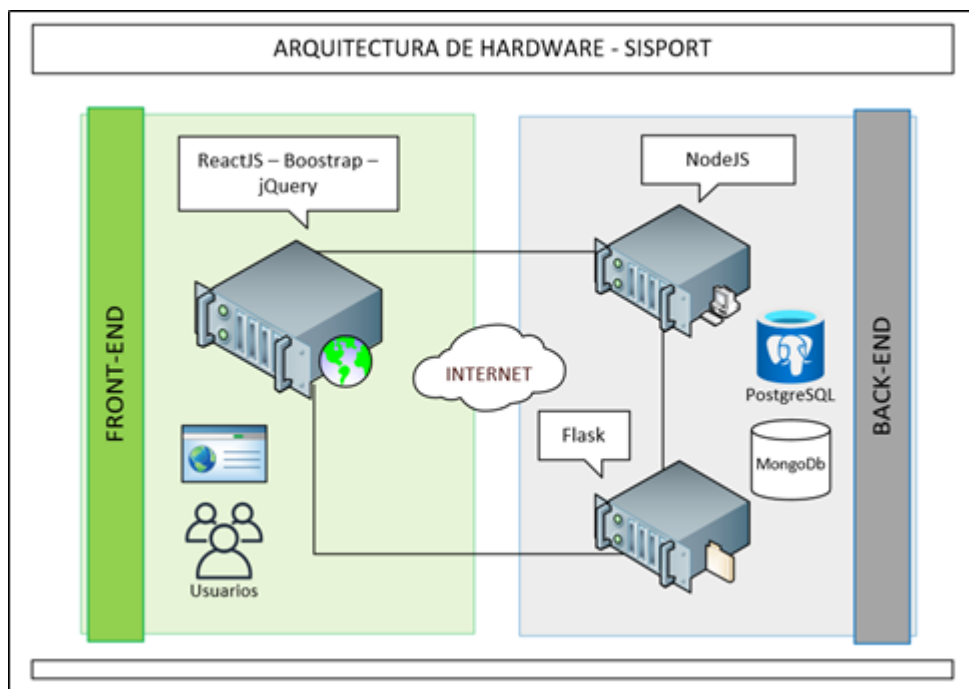
El sistema web a implementar requiere de varios equipos computacionales capaces de soportar la afluencia de información y usuarios conectados al mismo tiempo, para lograr esta funcionalidad se planteó la arquitectura de tipo cliente-servidor anteriormente mencionada, logrando la distribución de los recursos para la funcionalidad del aplicativo web.

En el lado del cliente tendremos un servidor independiente, con los frameworks de “Front-end” necesarios para la visualización de la información.

En el lado del servidor tendremos dos servidores, el primero contiene una API para la gestión de la información a través de conexiones a las bases de datos y el segundo servidor está dedicado netamente al almacenamiento de la información, es decir, todo lo correspondiente al “Back-end”.

A continuación, presentaremos la distribución de los servidores, para el sistema web a implementar (SISPORT):

Ilustración 1: Arquitectura de Hardware - SISPORT



Fuente: Elaboración propia

Como observamos en la **ilustración 1**, la distribución de los servidores y los distintos frameworks enlazados entre sí, permite elaborar un sitio web robusto a través de la separación de los servicios, es decir, si deja de funcionar el “servidor de archivos”, el sitio web seguirá funcionando sin problemas, hasta solucionar los errores presentados, creando flexibilidad tanto para el desarrollo, como para el manejo de la información de los usuarios.

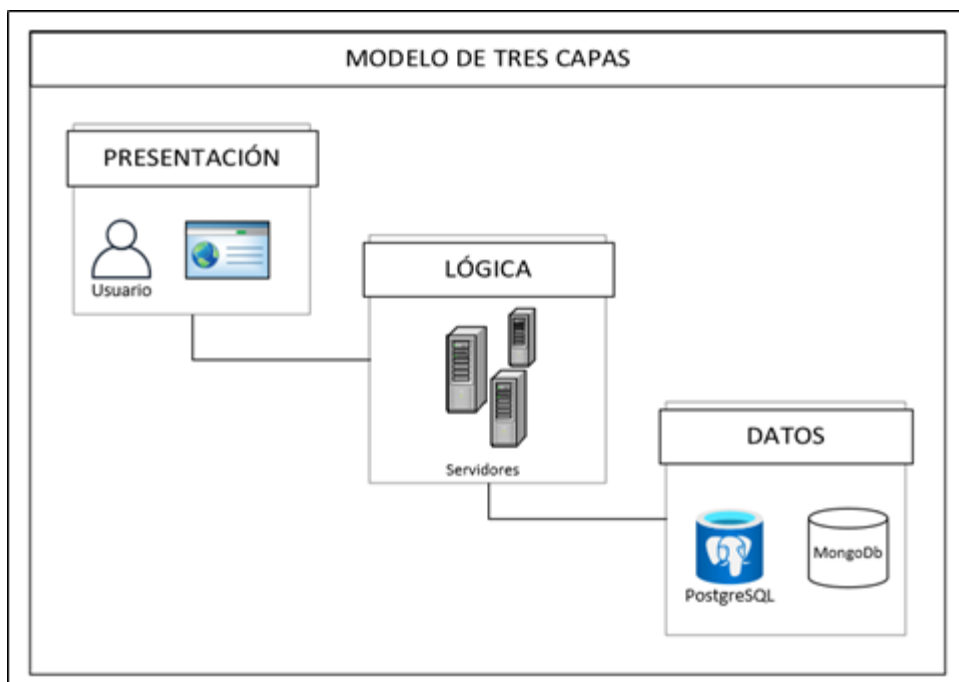
2.1.4. Arquitectura de software

La arquitectura de software se considera un paradigma de desarrollo, donde los componentes afectan directamente a los requisitos funcionales y no funcionales del sistema, garantizando que el software sea de calidad, detallando las herramientas de desarrollo, inversión, seguridad y el impacto que tendrá sobre el proyecto [9].

2.1.5. Arquitectura de software

El sistema web propuesto requiere de una distribución a nivel de capas o niveles, de tal forma que se estructure toda la plataforma web, de esta manera en caso de existir alguna modificación a futuro, no resulte afectado todo el sistema o las demás capas; dando una mayor flexibilidad al momento de gestionar el desarrollo general del proyecto, por tal motivo, se implementará el modelo de tres capas donde se separa la lógica de los datos, la información y su presentación al usuario [10].

Ilustración 2: Modelo de Tres capas



Fuente: Elaboración propia

El modelo presentado en la **ilustración 2**, está fundamentalmente diseñado en un entorno de ejecución Linux, que utiliza la potencia de los sistemas

operativos de código abierto, en este caso, “CentOS 8”, para desarrollar proyectos de software a gran escala y de nivel “empresarial” con la mayor seguridad posible, utilizando las herramientas o complementos necesarios alojados en sus plataformas digitales [11].

2.1.6. Capa de presentación

En este nivel de capa se elaboran los diseños o plantillas que son visualizadas para cliente, a esto se conoce como “Front-end”, y es donde se produce la interacción del usuario con las funcionalidades ofrecidas por las aplicaciones [12].

2.1.7. Capa lógica

La capa lógica corresponde a todos los procesos inmersos en la gestión de la información, es decir, realiza la acción de un intermediario entre los datos almacenados en las bases de datos y el usuario que solicita dicha información, para lo cual existen diferentes lenguajes de programación que facilitan las conexiones múltiples; en el caso de la propuesta presentada, se requiere un servidor con el lenguaje de programación Python (Flask) y otro servidor con el lenguaje JavaScript (NodeJS).

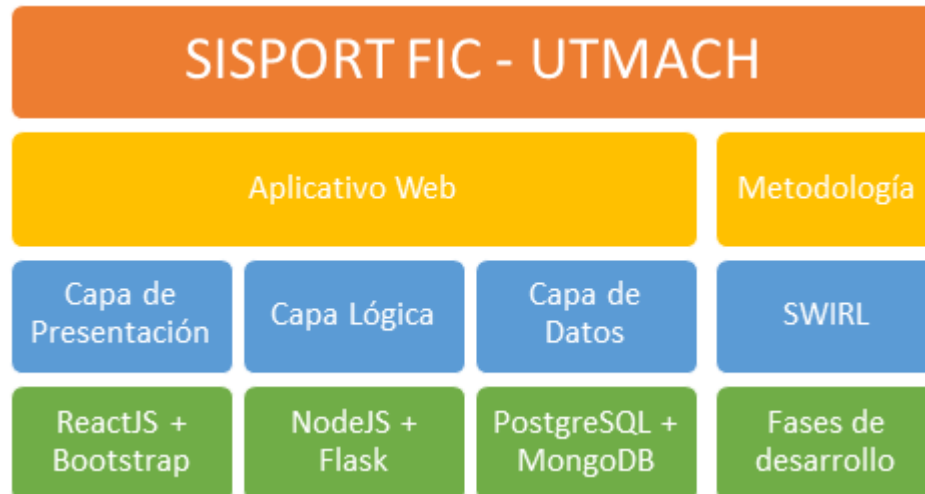
2.1.8. Capa de datos

La capa de datos consiste en el almacenamiento de la información, tanto de usuarios, compras, ventas, inventario, etc., a fin de ser utilizada para el análisis de información a través de modelos de bases de datos.

2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO

El proyecto tecnológico propuesto tiene bases de fundamentación sólidas, que permite el direccionamiento de software de calidad, a continuación, se detalla la estructura en componentes de las herramientas y las tecnologías aplicadas:

Ilustración 3: Fundamentación teórica



Fuente: Elaboración propia

2.2.1. Metodología de desarrollo de software

Las metodologías de desarrollo permiten mantener un estándar de desarrollo, creando un modelo y ambiente agradable sobre las técnicas y procesos que permitan la elaboración de pruebas, implementación y gestión de proyectos de software, siguiendo el modelo de ciclo de vida del software, a partir de los análisis de requisitos, diseño, desarrollo, pruebas y mantenimiento, siendo estos los factores para la determinación de un software de calidad [13].

2.2.1.1. Metodologías Tradicionales

Las metodologías tradicionales consisten en el proceso continuo de documentación, planificación y distintos procesos en cada una de las etapas del desarrollo del software, es decir, un seguimiento detallado en cada fase del proyecto, con la pequeña desventaja de falta de flexibilidad y dificultad en las evaluaciones de prototipos a medida que crece el proyecto [14].

2.2.1.2. Metodologías Ágiles

Las metodologías ágiles consisten en la realización de procesos iterativos con objetivos a corto plazo, disminuyendo la cantidad de documentación a realizar y enfocarse en el desarrollo de calidad del producto, a través de la realización de prototipos de forma periódica [15].

2.2.1.3. Metodologías Híbridas

Las metodologías híbridas consisten en la agrupación de distintas metodologías, a partir de las debilidades de las metodologías ágiles y tradicionales, con la finalidad de ser una metodología robusta y flexible a la vez, dando mejores prácticas en el desarrollo de software [16].

2.2.1.4. Metodología SWIRL

SWIRL (Software Web Iterativo Relacional Lógico), es una metodología ágil de desarrollo, que implementa una guía para la elaboración de procesos iterativos, siguiendo el proceso estructural del desarrollo de aplicaciones web a través de seis fases definidas en el ciclo de vida del desarrollo de software [3].

Ilustración 4: Ciclo de vida SWIRL



Fuente: Elaboración propia

Como se presenta en la **ilustración 4**, las fases forman parte de un proceso iterativo, desde el análisis de los requerimientos, hasta la puesta en marcha del producto web resultante.

2.2.2. Cliente

El cliente web se basa en la interacción entre el usuario final y el servidor de aplicación usando como medio un sistema basado en lenguaje de Front-end (HTML, CSS y JavaScript), el cual realiza peticiones o solicitudes hacia un servidor con una API lista para ser consumida, entregando una respuesta hacia el cliente [17].

2.2.2.1. Bootstrap

Bootstrap es un framework de “CSS”, que posee las herramientas necesarias para elaborar diseños de sitios y aplicaciones web responsivas y visualmente usables para el usuario que interactúa con la página, reduciendo la creación de archivos “CSS” comúnmente utilizados [18].

2.2.2.2. jQuery

jQuery es un conjunto de librerías de JavaScript que permite la interacción en tiempo real entre el usuario y las páginas web, de esta forma se puede manipular documentos, controlar eventos, animaciones y complementar con acciones en botones e ingreso de información al sitio web [19].

2.2.2.3. Framework ReactJS

ReactJS es un framework de tipo “Front-end” que facilita la creación de las múltiples interfaces de usuario de manera interactiva, simple y veloz; siguiendo el modelo de renderización de componentes por cada elemento del sitio web [20] , convirtiéndolo en una herramienta ideal para un sitio web potente.

2.2.3. Servidor

Un servidor es un equipo informático que responde a múltiples solicitudes de distinto propósito, puede ser de tipo: web, base de datos, sesiones, ftp,

correo, etc., utilizando distintos protocolos y puertos, atendiendo las peticiones de los clientes que se conectan a él; los cuales pueden ser otro equipo informático o un usuario final [21].

2.2.3.1. Servidor Web

Un servidor web es aquel equipo que tiene la función de distribuir o presentar sitios webs, a través de peticiones de un cliente de web. La información que se envía corresponde a archivos de tipo JavaScript, HTML y CSS, pero para simplificar la elaboración de aquellos sitios se utilizan frameworks de desarrollo web que poseen potentes recursos para las conexiones a bases de datos, gestión de información y más [22].

En la implementación del sistema para la gestión de portafolios se implementará un servidor web con el sistema operativo CentOS 8, el cual es una distribución de Linux.

2.2.3.2. Framework NodeJS

Node JS es una herramienta para el desarrollo web el cual monta un entorno de ejecución y utiliza librerías o módulos de contenido para conexiones a bases de datos, servicios en tiempo real y creación de una API para el consumo de información, lo cual lo convierte en una herramienta ideal para múltiples conexiones de usuarios [23].

2.2.3.3. Framework Flask

Flask es un framework de desarrollo de servidores web basado en el lenguaje de programación Python, con la ventaja de ser simple, ligero y rápido, dando a los desarrolladores la capacidad de aprovechar la potencia de Python para crear sitios webs grandes y complejos [24], además es ideal para el manejo de archivos, documentos, y grandes volúmenes de datos.

2.2.4. Base de datos

Las bases de datos son un elemento fundamental para los sistemas computarizados, debido a que no solo se registra información, al contrario, se

puede realizar transacciones para el análisis estadístico, gestión de procesos, administración y la propia seguridad de la información [25].

Existen actualmente dos tipos de bases de datos, las estructuradas o de tablas, que son las que admiten operaciones de tipo SQL, y otras no estructuradas o NoSQL, las cuales se manejan por colecciones de información; ambas son implementadas según las necesidades del sistema a desarrollar.

2.2.4.1. Base de datos relacionales

Las bases de datos relacionales constituyen un conjunto de datos no redundantes interrelacionados entre sí, con estructura de tablas, utilizando el modelo de filas y columnas, en donde se almacena información para ser analizada y procesada con base al diseño relacional elaborado [26].

2.2.4.1.1. PostgreSQL

PostgreSQL es un sistema gestor de base de datos de tipo SQL u objeto relacional, el cual utiliza la estructura de tablas, es decir, filas y columnas para el manejo de la información; además de poder relacionar información entre sí a través de claves foráneas y distintos scripts para la gestión de los datos [27].

2.2.4.2. Base de datos no relacionales

Las bases de datos no relacionales o NoSQL, nacen con la necesidad de un cambio del manejo de la información, donde la velocidad y el volumen de datos es primordial, creando la posibilidad de sistemas más distribuidos, con estructura cambiante sin afectar la integridad de los datos, convirtiéndose en una herramienta primordial para el soporte de aplicaciones a gran escala [28].

2.2.4.2.1. MongoDB

MongoDB es un sistema gestor de base de datos de tipo NoSQL, en el cual los registros de información se realizan a través de documentos en formato JSON, el cual posee una estructura similar a los archivos XML, y se agrupan

en “Colecciones”; este tipo de base de datos es ideal cuando se trata de manejar grandes volúmenes de información debido a la flexibilidad de la información al momento de insertar, editar y eliminar datos [29].

2.2.5. Herramientas CASE

Las herramientas CASE permiten diseñar, desarrollar y dar mantenimiento a los proyectos de software, incorporando las metodologías para un proceso disciplinado de desarrollo, de esta forma, estas herramientas sirven de guía en la implementación de software de calidad, aceleran el proceso de desarrollo y posibilita la reutilización de los componentes de software [30].

2.2.5.1.1. Microsoft Visio

Microsoft Visio es una herramienta ideal para la ampliación de soluciones personalizadas con base al desarrollo de diagramas, topologías de requisitos y caso de negocios, además de la colaboración en tiempo real cumpliendo los parámetros de diseños de sistemas [31].

2.2.5.1.2. Microsoft Project

Microsoft Project es una herramienta de la suite ofimática que permite enfocar o dirigir el proceso de gestión de proyectos multipropósitos, a través de la creación de cronogramas de actividades y diagramas de “Gantt” para la simplificación de la administración de los sistemas complejos a través de una interfaz simple y dinámica [32].

2.2.5.1.3. Enterprise Architect

Enterprise es una herramienta perfecta para poder modelar, diseñar y analizar los sistemas de desarrollo de software, tanto a nivel de los procesos y arquitecturas, manteniendo el control del proyecto e incluso trabajando en conjunto con múltiples desarrolladores, elaborando software de calidad [33].

2.2.5.1.4. Toad Data Modeler

Toad es una herramienta que permite crear modelos de distintas bases de datos, sin importar el tipo de estructura SQL que posean, ya que construye

las sentencias DDL a partir de modelos lógicos y físicos, definiendo incluso las claves principales y foráneas para establecer las relaciones entre tablas, de esta forma se simplifica su creación en los sistemas gestor de base de datos [34].

2.2.5.1.5. Balsamiq Mockups

Balsamiq es una herramienta dedicada al desarrollo del maquetado de interfaces, presentando de forma visual el posible resultado del sitio web, es decir, se acerca a lo que vería el usuario final, enriqueciendo de forma interactiva los requisitos funcionales del sistema a implementar [35].

2.2.5.1.6. StarUML

StarUML es una sofisticada tecnología para el modelamiento de software de carácter ágil y simple, debido a que permite crear modelos UML y todo tipo de diagramas para el desarrollo de software, con la opción de generar código de programación a través de modelado de clases [36].

2.2.5.1.7. Bizagi Modeler

Bizagi Modeler permite crear el modelado de procesos y diagramas de actividades de forma intuitiva, optimizando incluso el flujo de trabajo que posean las actividades del sistema, con la alternativa de ejecutar simulaciones y aumentar los procesos de eficacia en la organización [37].

2.3. OBJETIVOS DEL PROTOTIPO

2.3.1. Objetivo General

Desarrollar una plataforma en la web para la gestión de portafolios académicos en la Facultad de Ingeniería Civil, de la Universidad Técnica de Machala, mediante el uso de tecnologías para el desarrollo web y la metodología ágil “SWIRL”, para la automatización del proceso de recolección de evidencias académicas a lo largo del período académico de cada estudiante.

2.3.2. Objetivos Específicos

- Analizar información relacionada sobre las herramientas para el desarrollo web de tipo “Back-end” y “Front-end”.
- Implementar el proceso iterativo de la metodología SWIRL, mediante la aplicación de sus fases de desarrollo.
- Determinar los requisitos de tipo funcionales y no funcionales a partir de las historias de usuarios.
- Diseñar prototipos visuales del funcionamiento de la aplicación web utilizando las herramientas CASE.
- Codificar las clases y componentes de cada servidor, para el funcionamiento en conjunto de la aplicación web.
- Evaluar la plataforma web, utilizando el estándar de calidad ISO/IEC 9126.

2.4. DISEÑO DEL PROTOTIPO

El diseño de la plataforma web propuesta está fundamentada en la metodología “SWIRL”, por lo tanto, en esta etapa se detallan las fases del ciclo de vida de la metodología, a través de la planificación y modelado del sistema.

2.4.1. Fase de análisis

La fase de análisis se realiza previo al diseño y desarrollo de la propuesta, en donde se obtiene información sobre los requisitos funcionales y no funcionales del sistema, emitidos por los interesados del proyecto, definiendo su alcance con las respectivas limitaciones y costos de inversión, permitiendo determinar el grado de factibilidad de la implementación del aplicativo web.

2.4.1.1. Alcance del proyecto

El alcance del proyecto surge al momento de analizar las funcionalidades del sistema, resumiendo los módulos que se deben gestionar con menor nivel de prioridad y dar paso a los procesos esenciales para alcanzar el objetivo planteado.

Tabla 1: Funcionalidades del sistema

FUNCIONALIDADES DE SISPORT
Permitir la gestión de estudiantes y asociación con las asignaturas correspondientes a través de una clave de acceso al portafolio.
Permitir la gestión de docentes y asociación con las asignaturas que imparten.
Permitir la visualización del portafolio de las asignaturas, estructurados por carpetas.
Permitir la subida de archivos y documentos en múltiples formatos.
Permitir la generación de diarios de clase con base al plan de clase establecido en el syllabus académico.
Permitir descargar el portafolio completo.

Fuente: Elaboración propia

Con base a las funcionalidades del aplicativo web, se establecen las limitaciones del sistema según los objetivos propuestos, cabe recalcar, que podrían implementarse modificaciones en el futuro, para mejora de en caso de ser necesario.

Tabla 2: Limitaciones del sistema

LIMITACIONES DE SISPORT
No se generan reportes visuales de carácter complejo.
No se controla el contenido de información que ingresa el estudiante al portafolio académico.
Se carece de un super usuario para el manejo de la información en todas las facultades.
No existe un módulo de notificaciones.
Los portafolios se elaboran únicamente de forma individual.

Fuente: Elaboración propia

Las limitaciones planteadas en la **tabla 2**, permite el enfoque en el desarrollo de los procesos de carácter fundamental, evitando la programación de módulos no prioritarios en el aplicativo web.

2.4.1.2. Análisis de la factibilidad

En términos de inversión se realiza una estimación del costo total para determinar qué tan factible es la implementación del aplicativo web, cabe recalcar que se requiere de varios equipos tecnológicos capaces de soportar un volumen alto de información y procesamiento.

2.4.1.2.1. Factibilidad operativa

La factibilidad operativa permite verificar si la universidad posee las condiciones necesarias para implementar el sistema web, a continuación, se presenta la tabla de análisis.

Tabla 3: Factibilidad operativa del sistema

NÚMERO	ACTIVIDAD	PRIORIZACIÓN
1	Compra de un rango de direcciones IP públicas para el acceso a servidores desde la web.	Alta
2	Compra de un dominio de internet para facilitar el acceso de los usuarios a la plataforma.	Alta
3	Se necesita implementar instructivos o capacitaciones para los usuarios, sobre el correcto uso del sistema.	Medio
4	El sistema está apegado a las políticas y reglamento universitario.	Alto

Fuente: Elaboración propia

2.4.1.2.2. Factibilidad técnica

A diferencia de un aplicativo web tradicional montado en un hosting de pago mensual en internet, esta plataforma requiere ser montado en un servidor físico dentro de la universidad, por motivos de que se necesita manejar un amplio volumen de información por cada estudiante, por ende, el costo anual de un servidor web subiría a cantidades exorbitantes en plataformas que cobran por cada “GB” de información y velocidad de carga y descarga, sin contar la administración de las bases de datos.

Si analizamos detenidamente, en la Facultad de Ingeniería Civil, existe un promedio de 600 estudiantes y cada uno de ellos requiere de un portafolio por cada asignatura que se encuentre matriculado; considerando que el promedio es 6 asignaturas por cada semestre, tendríamos un total de 3600 portafolios académicos; los cuales poseen un máximo 32 diarios de clases y contando las evaluaciones, actividades autónomas, intraclases, casos de uso, proyectos, prácticas de laboratorios, etc., se formaría un total aproximado de 200 archivos por estudiante y si redondeamos a 5 mb por cada archivo, tendremos un total de 1 GB por estudiante, es decir al final del semestre debemos almacenar en promedio 3.6 TB de información, y por tal motivo se

requiere de un servidor de archivos potente y con alto volumen de almacenamiento.

Cabe destacar que, por motivo de ahorros de recursos, se puede analizar la posibilidad de eliminación de portafolios “incompletos” o aquellos que no son seleccionados como el “mejor” portafolio de clase.

A continuación, se presentan los recursos de hardware y software por cada servidor a utilizar.

Tabla 4: Factibilidad técnica del sistema

FACTIBILIDAD TÉCNICA			
RECURSOS DE HARDWARE		RECURSOS DE SOFTWARE	
CANT.	DESCRIPCIÓN	CANT.	DESCRIPCIÓN
1	Procesador i7 de 8va generación o superior.	3	CentOS 8
2	Procesador i5 de 8va generación o superior.	1	Base de datos MongoDB y PostgreSQL
1	Memoria RAM DE 16GB.	1	IP pública
2	Memoria RAM de 8GB.	1	Dominio
3	Adaptadores de red.	1	ReactJS
3	SSD de 480GB.	1	NodeJS
1	HDD de 4TB.	1	Flask

Fuente: Elaboración propia

Con las características planteadas en la **tabla 4**, se puede tener una buena gestión de la plataforma web, ofreciendo calidad de servicio en todo momento.

2.4.1.2.3. Factibilidad económica

A pesar que el desarrollo del sistema implica un gasto de recursos a nivel personal, no se consideró el costo de la codificación y pruebas del proyecto debido a que forma parte de la propuesta de trabajo de titulación, por tal

motivo, los costos son nulos. Sin embargo, se determinará un análisis del precio de los equipos tecnológicos que se requieren para la elaboración de la plataforma web, considerando que existe un solo programador para todo el sistema a implementar.

Tabla 5: Factibilidad económica del sistema

FACTIBILIDAD ECONÓMICA		
RECURSOS DE HARDWARE		
CANT.	DESCRIPCIÓN	COSTO
1	PC Ryzen 7 / 12 GB RAM / SSD 480GB + 1TB HDD	\$1200
TOTAL		\$1200
RECURSOS DE SOFTWARE		
CANT.	DESCRIPCIÓN	COSTO
1	Robo 3T	\$0
1	PgAdmin 4	\$0
1	Visual Studio Code	\$0
TOTAL		\$0

Fuente: Elaboración propia

2.4.1.3. Identificación de interesados o “stakeholders”

Los stakeholders son todos los actores que contribuyen directa e indirectamente a la toma de decisiones de un proyecto, su rol es fundamental para llevar a cabo la dirección operacional del sistema a implementar [38]; por lo tanto, en el sistema web propuesto se establecen como interesados del proyecto a los representantes administrativos, estudiantes y docentes de la Facultad de Ingeniería Civil, específicamente de las carreras de Ingeniería de sistemas y Tecnologías de la Información, debido a que poseen las pautas necesarias para la dirección del aplicativo web, basándose en la problemática existente de la falta de gestión de los portafolios académicos, tanto para docentes y estudiantes, por lo tanto, se presenta la matriz general de

interesados o “stakeholders” de donde se obtuvo la mayoría de los requisitos funcionales de la aplicación.

Tabla 6: Matriz de Interesados o Stakeholders

MATRIZ GENERAL DE STAKEHOLDERS			
Nombre	Rol	Profesión	Responsabilidad
Ing. Joofre Honores Tapia	Coordinador de carrera / Docente / Gerente de proyecto	Ing. De Sistemas	Limitaciones del sistema y revisión de entregables
Darwin Pilaloe Zea	Desarrollador	Estudiante	Diseño, planificación, implementación y pruebas

Fuente: Elaboración propia

2.4.1.4. Requerimientos

Para la recolección de requisitos se planteó un diálogo a modo de entrevista de tipo verbal (**ver anexo B**), con el personal administrativo, docentes, y estudiantes de la carrera de Ingeniería de Sistemas y Tecnologías de la información, tomando como referencia la experiencia académica en la creación de diferentes tipos de sistemas informáticos.

2.4.1.5. Requerimientos Funcionales (RF)

Los requerimientos funcionales son todos los procesos y servicios que debe realizar el sistema web, para alcanzar el objetivo del proyecto, a continuación, presentamos la tabla detallada de los requerimientos:

Tabla 7: Requerimientos funcionales del sistema

REQUERIMIENTOS FUNCIONALES (RF)	
CÓDIGO	DESCRIPCIÓN
RF-01	Página de bienvenida detallando la finalidad del sitio web para los visitantes, y la autenticación de usuarios con base al tipo de rol que posea.

RF-02	El docente debe habilitar las asignaturas que imparte por medio de una clave de matriculación.
RF-03	Matriculación de estudiantes a las respectivas asignaturas por medio de una clave de acceso
RF-04	Visualización del portafolio de la asignatura organizada por carpetas
RF-05	Ingreso de información detallada de cada diario de clase según el plan académico establecido en el syllabus.
RF-06	Gestión de subida de archivos y documentos en múltiples formatos.
RF-07	Descargar el portafolio de un estudiante determinado
RF-08	Visualización del perfil de usuario
RF-09	Gestión de estudiantes (Coordinador)
RF-10	Gestión de docentes (Coordinador)
RF-11	Gestión de asignaturas (Coordinador)

Fuente: Elaboración propia

2.4.1.6. Requerimientos No Funcionales (RNF)

Los requerimientos no funcionales, se enfocan en la calidad que tendrá el sistema con respecto a la velocidad, integridad de la información, seguridad, eficiencia y la usabilidad del mismo [39].

Tabla 8: Requerimientos no funcionales del sistema

REQUERIMIENTOS NO FUNCIONALES (RNF)	
CÓDIGO	DESCRIPCIÓN
RNF-01	El sitio web debe ser responsivo para diferentes dispositivos de conexión.
RNF-02	El sistema debe ser usable y de fácil acceso a las opciones existentes para usuarios poco experimentados.
RNF-03	La aplicación debe ser rápida al momento de agregar contenido al sistema.

RNF-04	Los mensajes de errores o advertencias deben estar ubicados en una región visible.
RNF-05	Las bases de datos relacionales y o relacionales deben estar bien estructuradas y validadas

Fuente: Elaboración propia

2.4.2. Fase de planificación

La fase de planificación consiste desde el análisis de requisitos de los interesados del proyecto, a través de las historias de usuario, y la gestión del cronograma de actividades por iteraciones; de esta forma, se creará un proyecto que cumpla con las expectativas y necesidades planteadas anteriormente.

2.4.2.1. Historias de usuario

Las historias de usuario fueron planteadas con base a los requisitos funcionales que debe tener el sistema, limitando el alcance del proyecto para facilitar el desarrollo y funcionamiento en general de todo el aplicativo web [40].

Cabe recalcar que pueden plantearse a futuro más historias de usuario según el proyecto lo requiera.

Tabla 9: Historia de usuario N1 - Inicio de sesión - RF01

Historia de usuario	
Número: 1	Usuarios: Estudiante, Docente y Coordinador
Nombre de historia: Ingreso al sistema web	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-01	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloe Zea	
Descripción: Los distintos tipos de usuarios existentes, deben ingresar al sistema con su cédula y contraseña; y se debe mostrar la interfaz correspondiente al tipo de usuario activo.	

Observaciones: Presentar el mensaje de “credenciales incorrectas” al momento de intentar autenticarse de forma errónea.

Fuente: Elaboración propia

La aplicación web debe poder autenticar las credenciales de los usuarios que intentan ingresar al sistema, y dependiendo del tipo de usuario se presentan las múltiples opciones e interfaces.

Tabla 10: Historia de usuario N2 - Clave de matriculación RF-02

Historia de usuario	
Número: 2	Usuarios: Docente
Nombre de historia: Clave de matriculación de asignaturas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-02	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: El usuario docente debe ingresar una clave para la matriculación en la asignatura que imparte, y dicha clave debe ser compartida a sus estudiantes con el objetivo de que puedan acceder al portafolio de la asignatura.	
Observaciones: Las claves de matriculación solo puede agregarse de acuerdo al docente que la imparte.	

Fuente: Elaboración propia

Debe existir un tiempo determinado para la validez de la clave de la asignatura, a fin de que no se pueda matricular fuera del periodo académico, además cada vez que un Docente quiera activar una asignatura, debe agregar el horario de clases respectivo en caso de no estar incorporado, a fin de que el sistema plantee los diarios de clase según la cantidad de horas vistas por semana.

Tabla 11: Historia de usuario N3 - Proceso de matriculación - RF03

Historia de usuario	
Número: 3	Usuarios: Estudiante
Nombre de historia: Proceso de matriculación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-03	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: El estudiante se matricula a la asignatura correspondiente a través de una clave de acceso brindada por el docente.	
Observaciones: Se debe presentar un mensaje de error, al momento de ingresar una clave de matriculación invalida.	

Fuente: Elaboración propia

El Estudiante al momento de matricularse, podrá tener acceso a su portafolio de la asignatura.

Tabla 12: Historia de usuario N4 - Visualización del portafolio - RF04

Historia de usuario	
Número: 4	Usuarios: Estudiante, Docente y Coordinador
Nombre de historia: Visualización del portafolio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-04	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: Los distintos usuarios podrán tener acceso a la visualización del portafolio académico; en caso de ser estudiante solo tendrá acceso a su portafolio, mientras que el docente podrá ver el de todos los estudiantes matriculados en su asignatura, y finalmente el coordinador podrá visualizar el de todos los estudiantes de su carrera.	
Observaciones: Solo el estudiante puede agregar información a la asignatura que está matriculado.	

Fuente: Elaboración propia

El Coordinador de carrera y el Docente de la asignatura, a pesar de tener un rango mayor de permisos, no podrán interferir en la recolección de evidencias del estudiante, solo son observadores con opción a descargar, mientras que los Coordinadores podrán eliminar portafolios en caso de ser estrictamente necesario.

Tabla 13: Historia de usuario N5 - Diarios de clase - RF05

Historia de usuario	
Número: 5	Usuarios: Estudiante
Nombre de historia: Diarios de clase	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-05	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloe Zea	
Descripción: El estudiante ingresa información respectiva al diario de clase, tales como: objetivos, contenidos, resumen, reflexión, imágenes.	
Observaciones: Cada diario de clase contiene la hora y fecha exacta, la cual es completado automáticamente por el sistema.	

Fuente: Elaboración propia

Los diarios de clase son ingresados completando campos específicos dentro de la aplicación, para luego generar automáticamente su diseño en un documento "Word".

Tabla 14: Historia de usuario N6 - Subida de archivos - RF06

Historia de usuario	
Número: 6	Usuarios: Estudiante
Nombre de historia: Subida de archivos y documentos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-06	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: El estudiante puede subir evidencias en distintos formatos de la suite ofimática (Word, Excel, PowerPoint, etc.).	
Observaciones: Se limita el peso máximo de los archivos a subir.	

Fuente: Elaboración propia

Los archivos y documentos que se suben al sistema deben ser controlados, para evitar el uso incorrecto del espacio de almacenamiento de la plataforma.

Tabla 15: Historia de usuario N7 - Descarga de portafolio - RF07

Historia de usuario	
Número: 7	Usuarios: Estudiante, Docente y Coordinador
Nombre de historia: Descarga del portafolio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Requerimiento funcional: RF-07	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: Los distintos usuarios pueden descargar los portafolios académicos según su rol, es decir, el estudiante solo puede descargar del cual está matriculado, mientras que el docente puede hacerlo por cada estudiante de la asignatura que imparte, y el coordinador tendrá acceso a todos los portafolios de la carrera.	
Observaciones: El archivo resultante es de tipo “.rar” que contiene toda la información académica.	

Fuente: Elaboración propia

El portafolio puede ser descargado en cualquier momento, sin restricción de contenido, es decir, en caso de existir diarios que falten de elaborar o archivos por subir, igualmente se puede realizar el proceso de descarga.

Tabla 16: Historia de usuario N8 - Gestión de estudiantes - RF08

Historia de usuario	
Número: 8	Usuarios: Coordinador
Nombre de historia: Gestión de estudiantes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-08	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloe Zea	
Descripción: El coordinador realiza las operaciones de tipo CRUD de cada estudiante ingresado al sistema.	
Observaciones: Se debe validar que al eliminar el estudiante no tenga relación con las asignaturas y portafolios.	

Fuente: Elaboración propia

El Coordinador de carrera tiene la opción de subir un archivo de tipo “.csv”, que contenga el listado de estudiantes matriculados, facilitando el ingreso de información de cada estudiante.

Tabla 17: Historia de usuario N9 - Perfil de usuarios - RF09

Historia de usuario	
Número: 9	Usuarios: Estudiante, docente y coordinador
Nombre de historia: Perfil de usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-08	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloe Zea	
Descripción: Cada usuario que esté actualmente con la sesión activa, podrá visualizar su información de perfil personal.	
Observaciones: La información a visualizar son campos no visibles para otros usuarios.	

Fuente: Elaboración propia

Tabla 18: Historia de usuario N10 - Gestión de docentes – RF10

Historia de usuario	
Número: 10	Usuarios: Coordinador
Nombre de historia: Gestión de docentes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-10	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloe Zea	
Descripción: El coordinador realiza las operaciones de tipo CRUD de cada docente ingresado al sistema.	
Observaciones: Se debe validar que al eliminar el docente no tenga relación con las asignaturas que imparte.	

Fuente: Elaboración propia

El Coordinador de carrera tiene la opción de subir un archivo de tipo “.csv”, que contenga el listado de Docentes que imparten clases en la carrera, facilitando el ingreso de información.

Tabla 19: Historia de usuario N11 - Gestión de asignaturas - RF11

Historia de usuario	
Número: 11	Usuarios: Coordinador
Nombre de historia: Gestión de asignaturas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Requerimiento funcional: RF-11	Iteración asignada: 1
Programador responsable: Darwin Josue Pilaloa Zea	
Descripción: El coordinador realiza las operaciones de tipo CRUD de todas las asignaturas ingresadas al sistema.	
Observaciones: Se debe validar que las asignaturas no tengan estudiantes y docentes asociados.	

Fuente: Elaboración propia

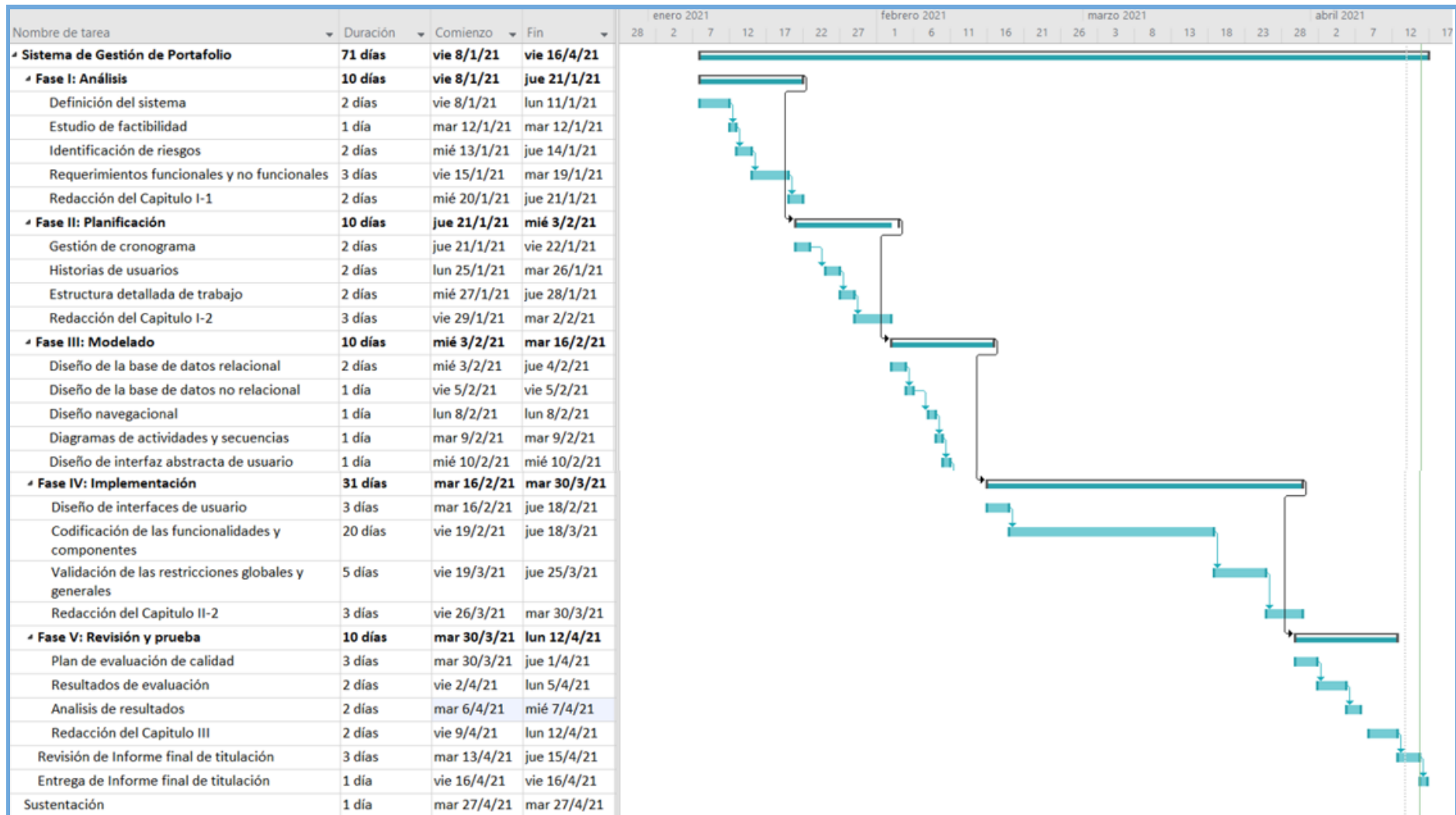
El Coordinador de carrera tiene la opción de subir un archivo de tipo “.csv”, que contenga el listado de las asignaturas que se imparten dentro de la carrera, facilitando el ingreso de información de cada asignatura y la asociación respectiva con su docente de área.

2.4.2.2. Gestión del cronograma

El cronograma de actividades posee de forma detallada el proceso para llevar a cabo el desarrollo de la plataforma web, desde: la recolección de requisitos, factibilidad, diseños de interfaces, e incluso el alcance del proyecto, hasta la fecha establecida para la presentación final de la documentación escrita.

A continuación, presentaremos el cronograma con las actividades y fechas establecidas y el respectivo diagrama de Gantt:

Ilustración 5: Cronograma de actividades



Fuente: Elaboración propia

Como se observa en la **ilustración 5**, existe un desglose secuencial de las fases de la metodología “SWIRL” con su respectivo diagrama de Gantt, a través de un tiempo definido (71 días) para la finalización del aplicativo web, exceptuando los tiempos de entrega del documento y la sustentación.

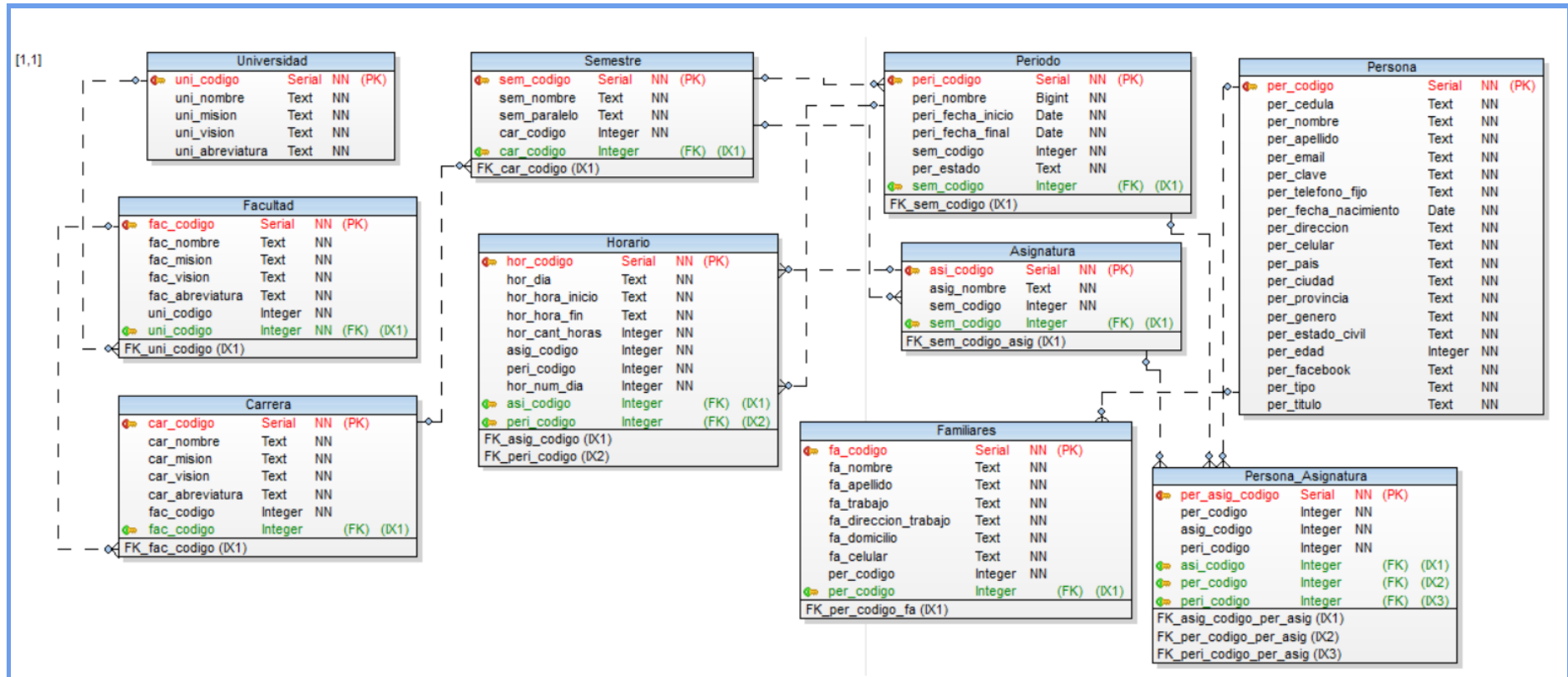
2.4.3. Fase de modelado

En esta sección se detalla el diseño lógico del manejo de la información a nivel de bases de datos, utilizando diagramas visuales que permitan identificar la relación que existe entre los distintos puntos del sistema. En nuestro proyecto, se implementan dos tipos de estructuras de bases de datos, las cuales serán detalladas según el uso de la información.

2.4.3.1. Diseño de base de datos Relacional

La Universidad Técnica de Machala maneja un esquema de bases de datos de tipo relacional, el cual contiene información administrativa y de organización por carrera y facultad, por lo tanto, se plantea un diseño que fusione ciertos módulos académicos como lo son: datos personales de estudiantes y docentes, syllabus de asignaturas, planes de clases, periodos académicos, etc., los cuales serán usados para organizar la estructura de los portafolios académicos en el sistema web; dando como resultado el siguiente modelo relacional:

Ilustración 6: Modelo de Base de Datos Relacional



Fuente: Elaboración propia

Como podemos observar en la **ilustración 6**, posee una estructura compacta y los campos necesarios para la distribución académica de la información correspondiente a evidencias diarias de clases, almacenadas en los portafolios académicos de las asignaturas.

Otra observación es la integración de los estudiantes y docentes en una sola tabla, con la diferencia del campo “tipo” que corresponde a los roles que posee dicho usuario o persona.

La concentración de los datos se basa en la tabla “asignaturas”, debido a que existen estudiantes que pueden ver múltiples asignaturas en diferentes paralelos e incluso semestres, por lo que la asociación se produce de tipo: “estudiante – asignatura” y “docente – asignatura”.

La incorporación de los horarios, permite determinar la cantidad de diarios metacognitivos de clases a elaborar por cada semestre según las horas impartidas establecidas en el syllabus académico.

Las tablas de facultad, carrera y universidad corresponden a información administrativa, tales como la misión y visión respectivamente, a fin de dejar las bases para expandir a más carreras y facultades en una próxima implementación.

2.4.3.2. Diseño de base de no datos Relacional o NoSQL

Para el manejo de un volumen alto de información, se diseñó una estructura de base de datos de tipo “no relacional” el cual es flexible y capaz de organizar los múltiples campos que maneja un portafolio de clases, desde la subida de archivos, hasta el ingreso de diferentes tipos de información por cada estudiante; además se organizó de tal forma que pueda expandirse para múltiples facultades y por ende las distintas carreras que conforman la universidad.

A continuación, presentamos el esquema de diseño de la base de datos “no relacional” en formato de tipo “JSON”, el cual se desplazará por subniveles de información.

Ilustración 7: Modelo de Base de Datos No Relacional

```
1 {
2   "sisport_utmach": {
3     "fic.is": {
4       {
5         "id": {
6           "cod_asignatura": "1",
7           "periodo": "2021-E1",
8           "nombre_asignatura": "PROGRAMACIÓN V",
9           "identificador": "I.S.B.12345",
10          "syllabus": "syllabus.pdf",
11          "docente": "Ing. Joofre Honores Tapia",
12          "clave": "ProgV2021E1",
13          "portafolios": [
14            {
15              "datos_informativos": {
16                "codigo_estudiante": "1055",
17                "autoretrato": "/foto_perfil.jpg",
18                "biografia": "Mi meta es ser Ingeniero de Sistemas"
19              },
20              "elementos_curriculares": {
21                "expectativas": "Aprender Python de forma profesional",
22                "apuntes": [
23                  {
24                    "unidad": "UNIDAD I - INSTRUCCIONES DE CONTROL DEL PROGRAMA",
25                    "tema": "Generalidades de Python",
26                    "fecha": "Viernes, 08 de Enero del 2021",
27                    "objetivo": "Analizar conceptos básicos de la programación python",
28                    "contenidos": "Encuadre de la asignatura, Historia, Tipos de datos",
29                    "resumen": "Python es un lenguaje potente, diseñado para proyectos multipropósitos",
30                    "reflexión": {
31                      "pregunta1": "Comprender en detalle la sintaxis del lenguaje de programación Python",
32                      "pregunta2": "Analizar la lógica de programación del lenguaje Python",
33                      "pregunta3": "Porque Python es un lenguaje de programación multiparadigma",
34                      "pregunta4": "Aprendí sobre las generalidades de Python"
35                    },
36                    "imagenes": "imagen1.jpg, imagen2.jpg"
37                  }
38                ],
39                "evaluaciones": [
40                  {
41                    "ruta": "/evaluación1.docx"
42                  }
43                ],
44                "investigaciones": [
45                  {
46                    "ruta": "/investigación1.docx"
47                  }
48                ],
49                "actividades": [
50                  {
51                    "ruta": "/actividad1.xls"
52                  }
53                ],
54                "proyectos": [
55                  {
56                    "ruta": "/proyecto.pdf"
57                  }
58                ],
59                "caso_estudio": [
60                  {
61                    "ruta": "/caso-de-estudio.docx"
62                  }
63                ],
64                "planteamientos": [
65                  {
66                    "ruta": "/planteamiento1.docx"
67                  }
68                ],
69                "asistencia": "/asistencia.docx",
70                "observaciones": [
71                  {
72                    "ruta": "/observación1.doc"
73                  }
74                ],
75                "intraclases": [
76                  {
77                    "ruta": "/intraclase1.doc"
78                  }
79                ],
80                "autonomos": [
81                  {
82                    "ruta": "/autonomo1.doc"
83                  }
84                ],
85                "refuerzo": [
86                  {
87                    "ruta": "/refuerzo1.doc"
88                  }
89                ],
90                "informe_final": "/informefinal.docx"
91              }
92            }
93          ]
94        }
95      }
96    }
97  }
98 }
```

Fuente: Elaboración propia

Como se puede observar en la **ilustración 7**, la información se encuentra estructurada por facultades y carreras a través de un “id” que combina sus abreviaciones, por ejemplo, en el caso de la Facultad de Ingeniería Civil con la carrera de Ingeniería de sistemas, tendría como resultado el id de “fic.is”, y dentro de dicha colección contiene un conjunto de arreglos que equivale a cada portafolio académico.

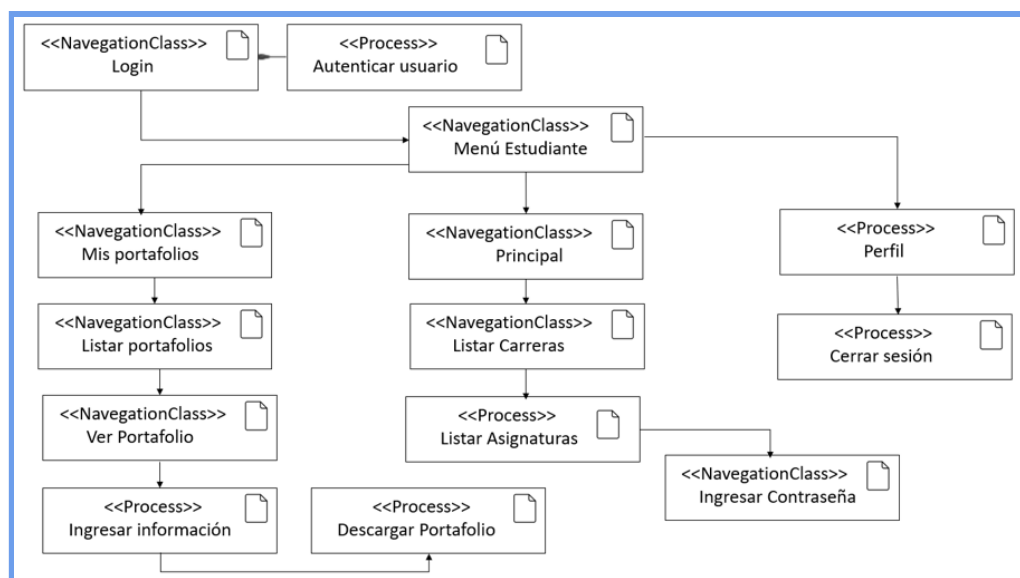
Dentro de cada portafolio existe un “_id” generado aleatoriamente para evitar índices duplicados, además de los datos esenciales de la asignatura y su respectiva clave de acceso, de esta manera, los estudiantes podrán matricularse a la asignatura y automáticamente se crea la estructura de portafolio con los datos del estudiante.

La información contenida en los portafolios sirve para almacenar rutas de archivos, documentos, e inserción de datos, manteniéndola siempre organizada, facilitando el acceso y descarga de la información.

2.4.3.3. Diseño navegacional

El diseño navegacional parte de un modelo conceptual, el cual mantiene una estructura organizacional de todos los procesos u entorno que componen la plataforma web [41].

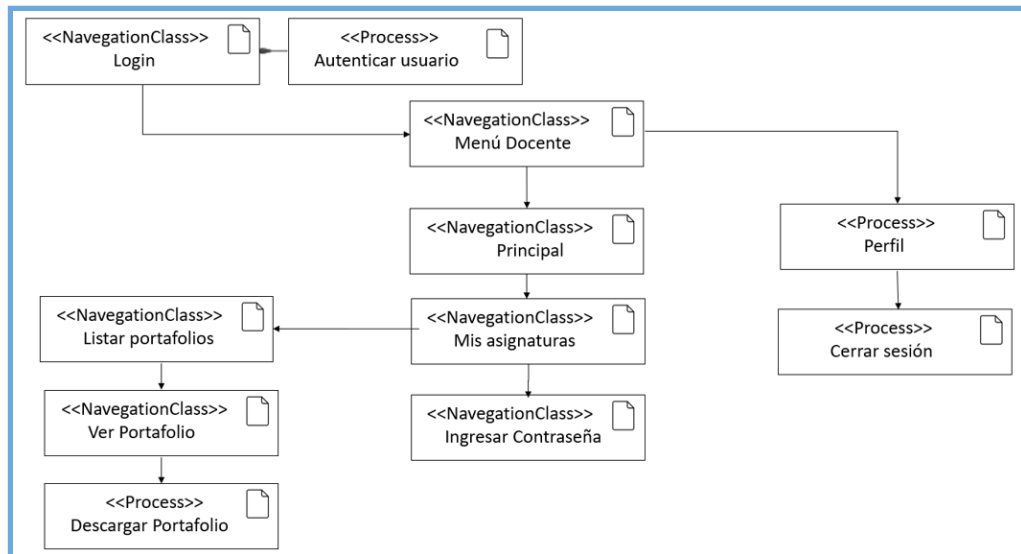
Ilustración 8: Diseño navegacional - Estudiante



Fuente: Elaboración propia

Como podemos observar en la **ilustración 8**, el usuario estudiante tiene acceso al listado de asignaturas por carreras, de tal forma que pueda filtrar la asignatura que debe matricularse para gestionar su portafolio académico.

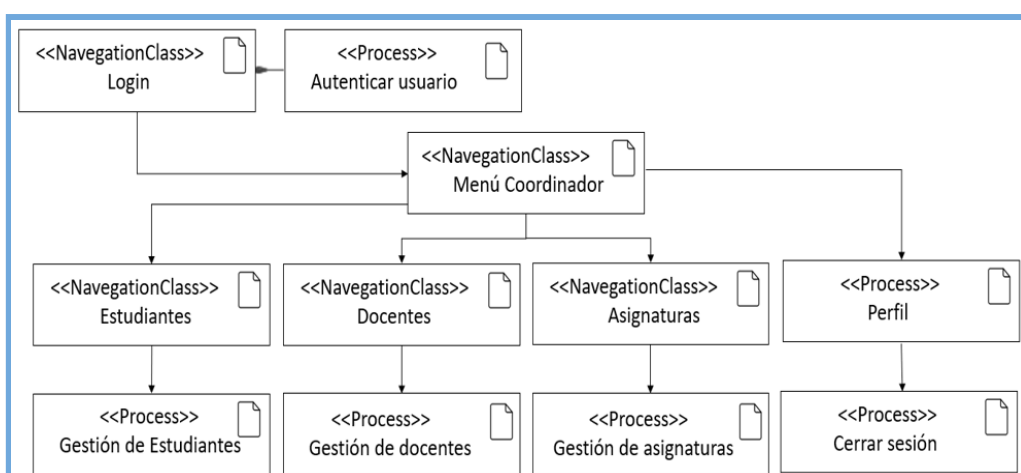
Ilustración 9: Diseño navegacional - Docente



Fuente: Elaboración propia

Como se visualiza en la **ilustración 9**, los docentes solo podrán observar las asignaturas que imparte, con opción a activarlas por una clave digitada por ellos, y acceder a los portafolios de los estudiantes, con opción a descarga.

Ilustración 10: Diseño navegacional - Coordinador



Fuente: Elaboración propia

Como se observa en la **ilustración 10**, el coordinador de carrera, tiene acceso a la navegación para la gestión de docentes, estudiantes y asignaturas.

2.4.3.4. Diseño de interfaces

Para el diseño de las interfaces se utilizó la herramienta Balsamiq Mockups, la cual es ideal para representar un modelo más apegado a la realidad del sistema a desarrollar, a continuación, se presentará todas las vistas de la plataforma web, con base a los requisitos definidos anteriormente.

Página Principal

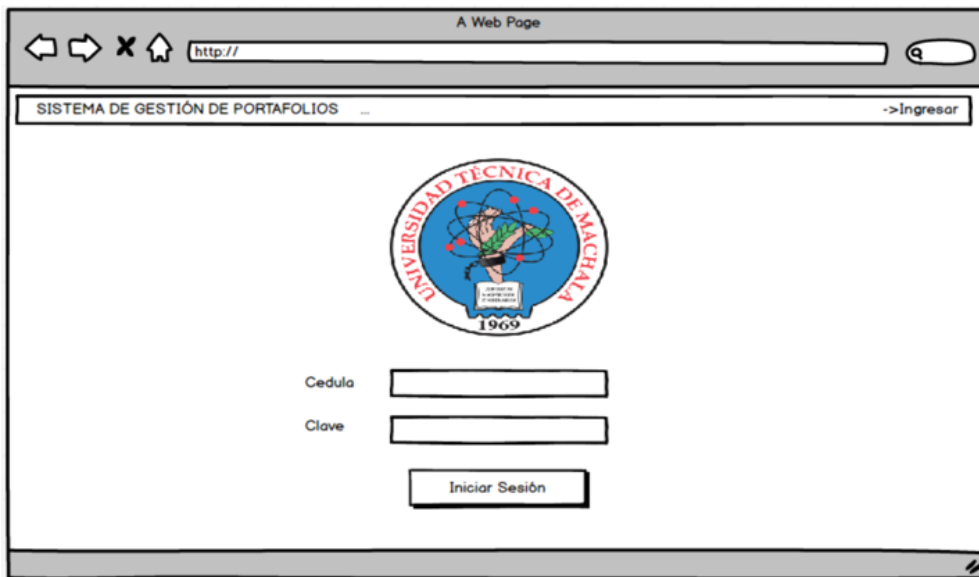
Ilustración 11: Diseño de inicio del sitio



Fuente: Elaboración propia

En la **ilustración 11**, se puede observar la página de bienvenida de la plataforma web, detallando que es un proyecto de tipo universitario por medio de los logos e imágenes académicas; además existe un panel de navegación de pestañas con el nombre del aplicativo web y la opción de ingreso al sistema.

Ilustración 12: Diseño de Inicio de Sesión

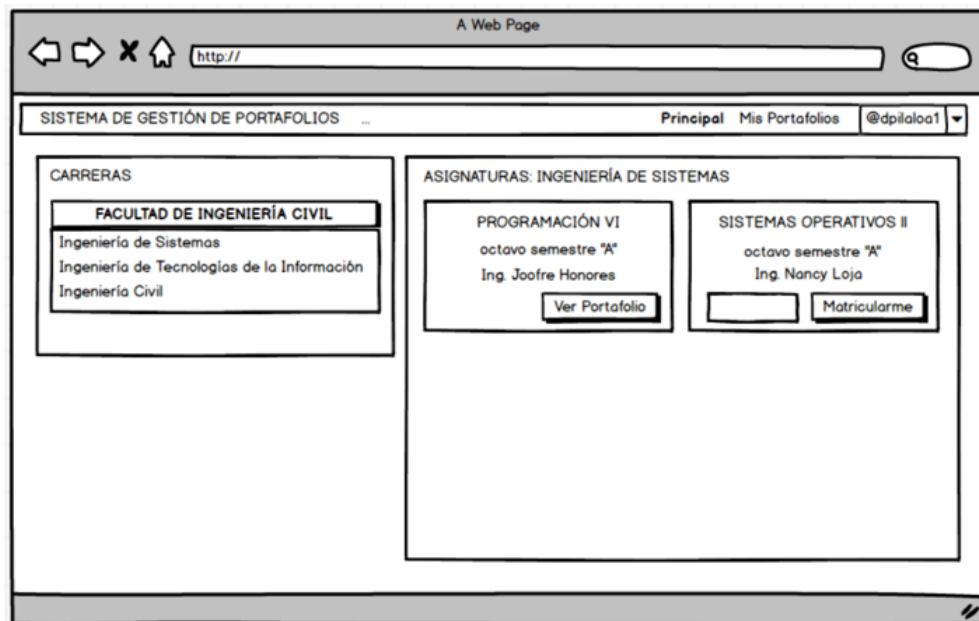


Fuente: Elaboración propia

En la **ilustración 12**, se procede a realizar el proceso de ingreso al sistema a través de la cédula y clave del estudiante (igual a la cédula del estudiante) hasta que el usuario cambie la contraseña por defecto.

Vistas de Estudiantes

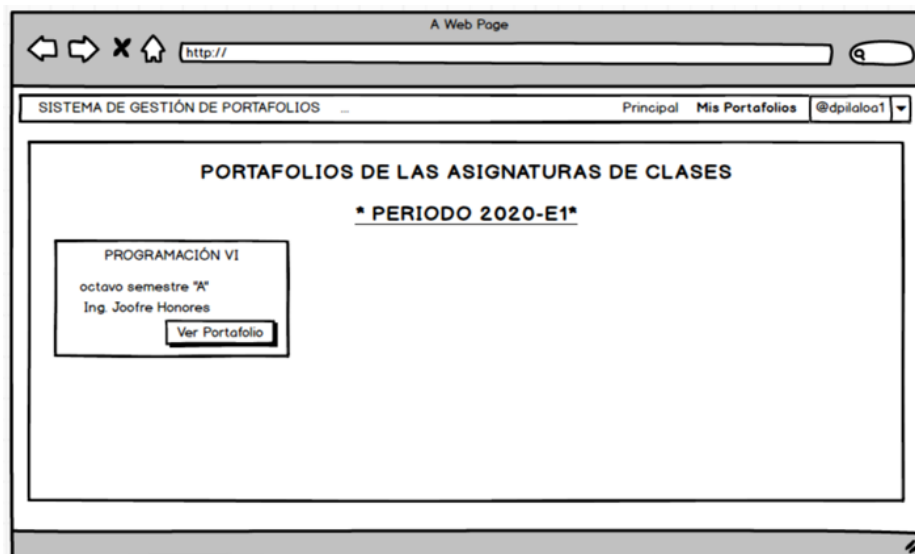
Ilustración 13: Diseño de menú principal del estudiante



Fuente: Elaboración propia

En la **ilustración 13**, se puede realizar la búsqueda de las asignaturas existentes por cada facultad y carreras, con su correspondiente periodo académico, incluyendo la opción de poder matricularse en una asignatura para poder elaborar su respectivo portafolio.

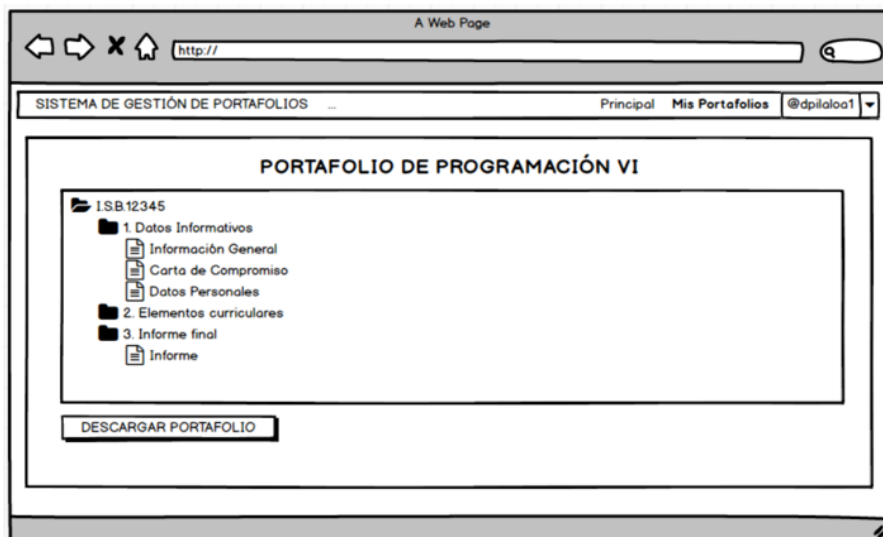
Ilustración 14: Diseño de portafolios del estudiante



Fuente: Elaboración propia

En la **ilustración 14**, se puede visualizar los portafolios organizados por periodos académicos en el cual está matriculado el estudiante actualmente activo.

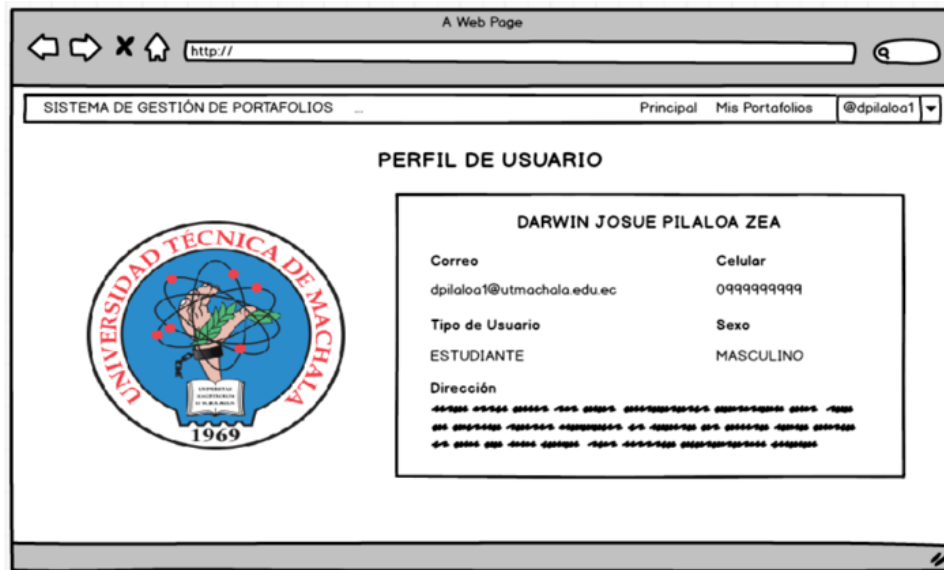
Ilustración 15: Diseño del portafolio del estudiante



Fuente: Elaboración propia

En la **ilustración 15**, se visualiza la estructura de cada portafolio académico, siguiendo el formato de “Portafolio de la asignatura”, con opción a ingresar información, archivos y documentos, y en caso necesario poder descargarlo.

Ilustración 16: Diseño de Perfil de usuario

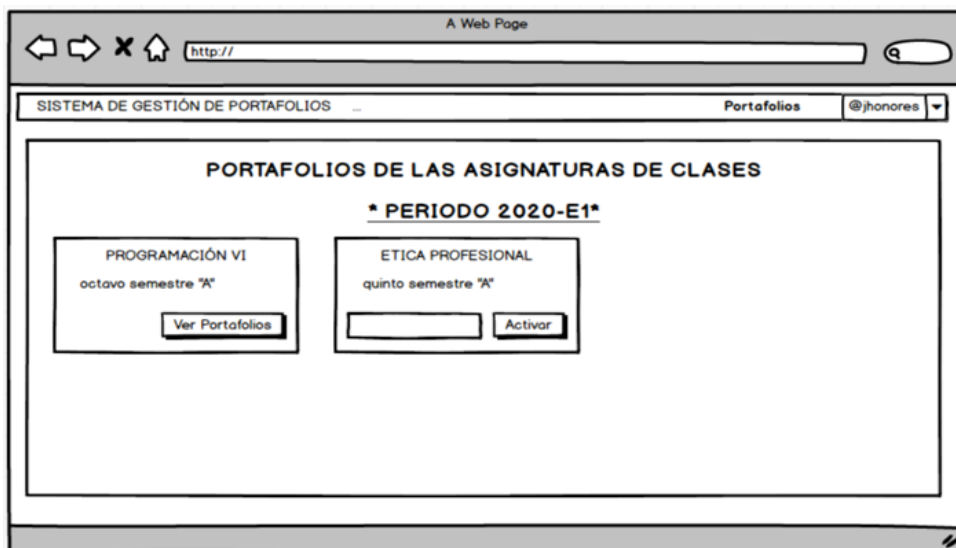


Fuente: Elaboración propia

En la **ilustración 16**, se obtiene información personal sobre el usuario actualmente activo, tanto docente, estudiante e incluso de rol coordinador.

Vistas de Docentes

Ilustración 17: Diseño de vista de portafolios del docente



Fuente: Elaboración propia

En la **ilustración 17**, los docentes pueden visualizar las asignaturas que imparten con la opción de ver todos los portafolios de los estudiantes y activar las asignaturas que consideren llevar a cabo el seguimiento de evidencias, para que sus estudiantes puedan agregar la información respectiva al portafolio académico.

Ilustración 18: Diseño de activación de asignaturas por el docente

	Hora Inicio		Hora Fin	
<input type="checkbox"/> Lunes	0	0	0	0
<input type="checkbox"/> Martes	0	0	0	0
<input type="checkbox"/> Miércoles	0	0	0	0
<input type="checkbox"/> Jueves	0	0	0	0
<input type="checkbox"/> Viernes	0	0	0	0

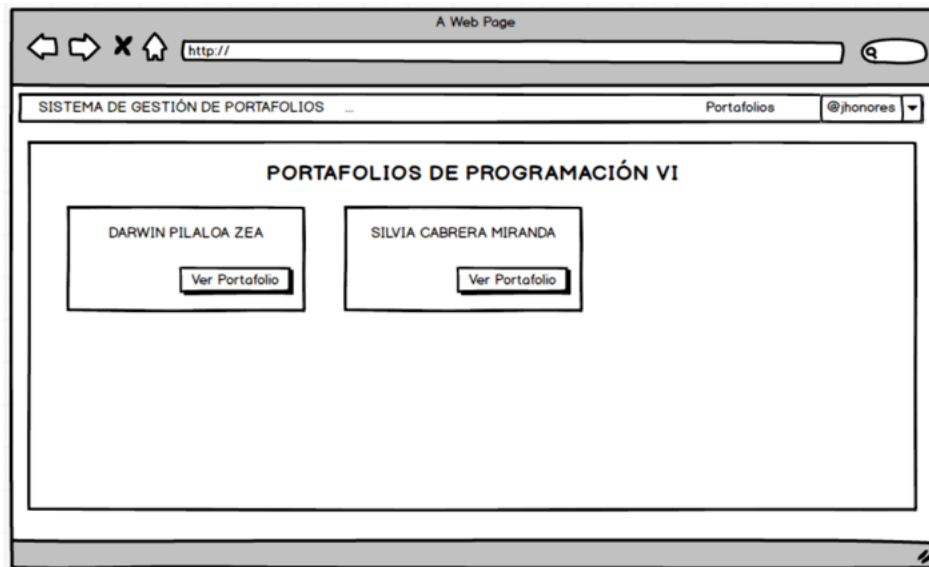
TOTAL DE HORAS: 48

Guardar Horario

Fuente: Elaboración propia

En la **ilustración 18**, se detalla el proceso para activar el portafolio académico de una asignatura en específico, en este caso, solicita el horario de clases en el cual se imparte la asignatura, para poder elaborar según las horas y días, la cantidad de diarios a elaborar durante el periodo académico.

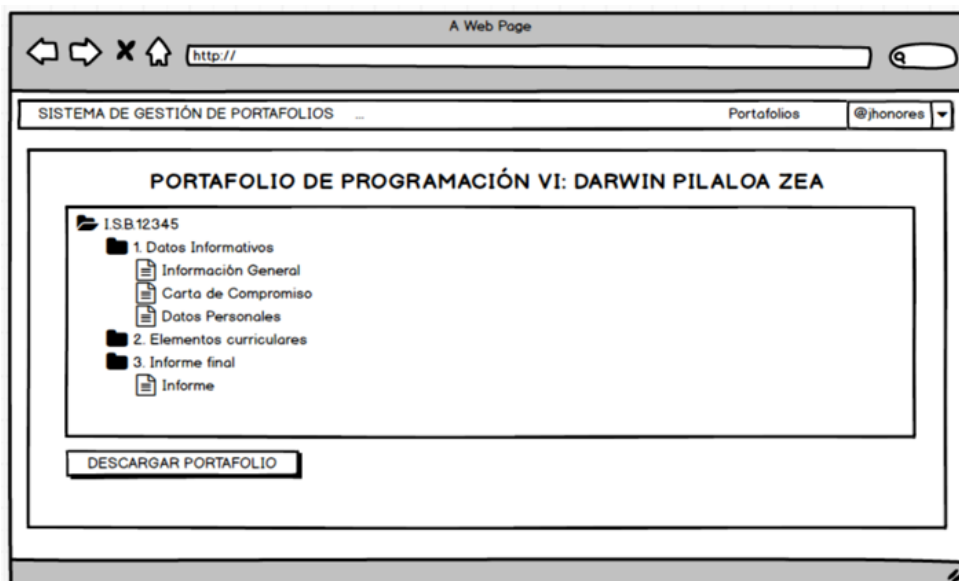
Ilustración 19: Diseño de portafolios por asignatura del docente



Fuente: Elaboración propia

En la **ilustración 19**, aparece en listado los portafolios elaborados por cada estudiante correspondiente a la asignatura impartida por el docente actualmente activo.

Ilustración 20: Diseño de portafolio del estudiante desde el docente

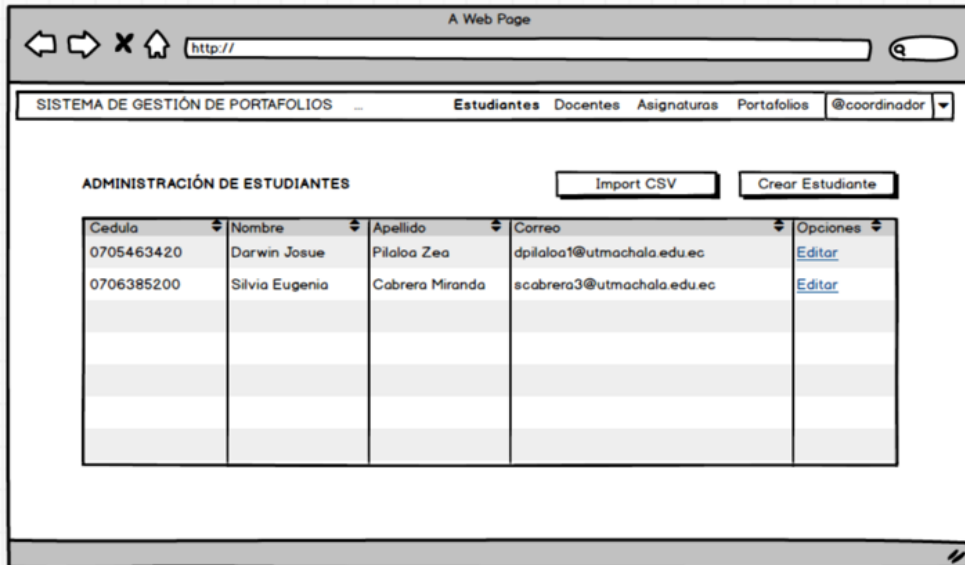


Fuente: Elaboración propia

En la **ilustración 20**, se visualiza en detalle el portafolio del estudiante, pero el docente solo tiene el rol de observador, y puede descargarlo en caso de ser necesario.

Vistas de Coordinadores

Ilustración 21: Diseño de gestión de estudiantes



Fuente: Elaboración propia

En la **ilustración 21**, se puede visualizar todos los estudiantes ingresados al sistema con opción a editar campos, y en caso de ser necesario poder agregar estudiantes con la importación de un archivo de tipo “.csv”.

Ilustración 22: Diseño de creación de estudiantes

CREACIÓN DE ESTUDIANTES

Cedula: 0705463420 Nombres: Darwin Josue Apellidos: Pilaloo Zea Correo: dpilaloo1@utmachala.edu.ec

Genero: Masculino Celular: 0999999999 Fec_nac: / / Estado Civil: Soltero

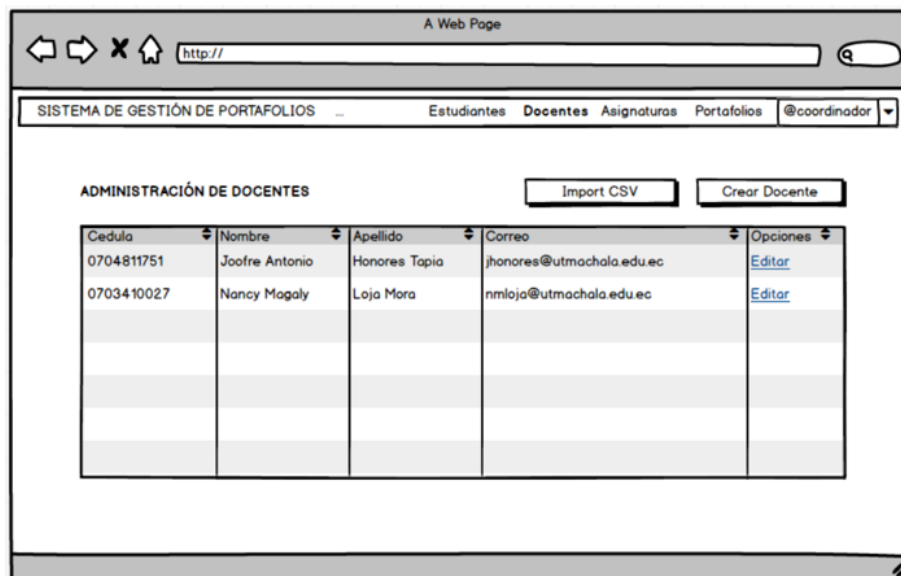
Direccion:

Crear Estudiante

Fuente: Elaboración propia

En la **ilustración 22**, se inserta información de forma manual, de los estudiantes, para la gestión de sesiones y creación de portafolios respectivamente.

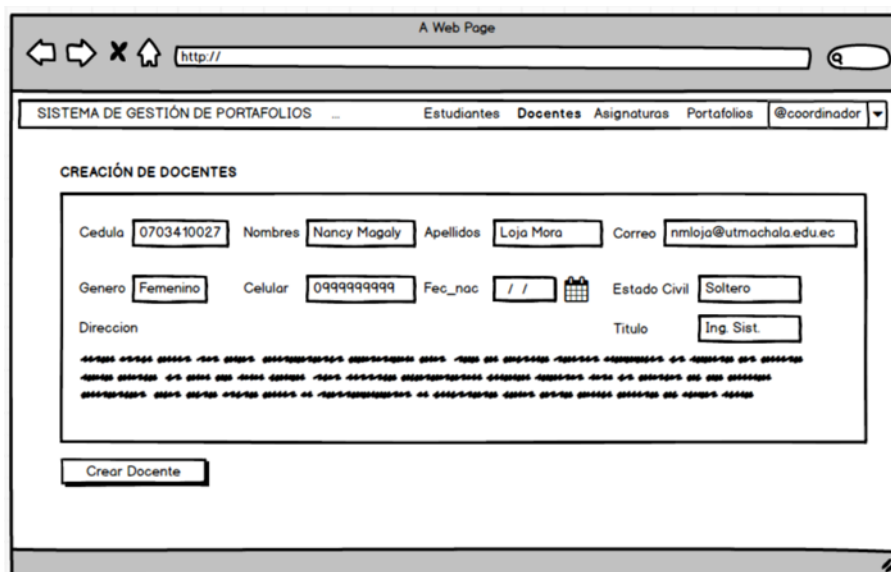
Ilustración 23: Diseño de gestión de docentes



Fuente: Elaboración propia

En la **ilustración 23**, se puede visualizar todos los docentes ingresados al sistema, y en caso de contar con un archivo “.csv”, poder realizar la respectiva importación de datos.

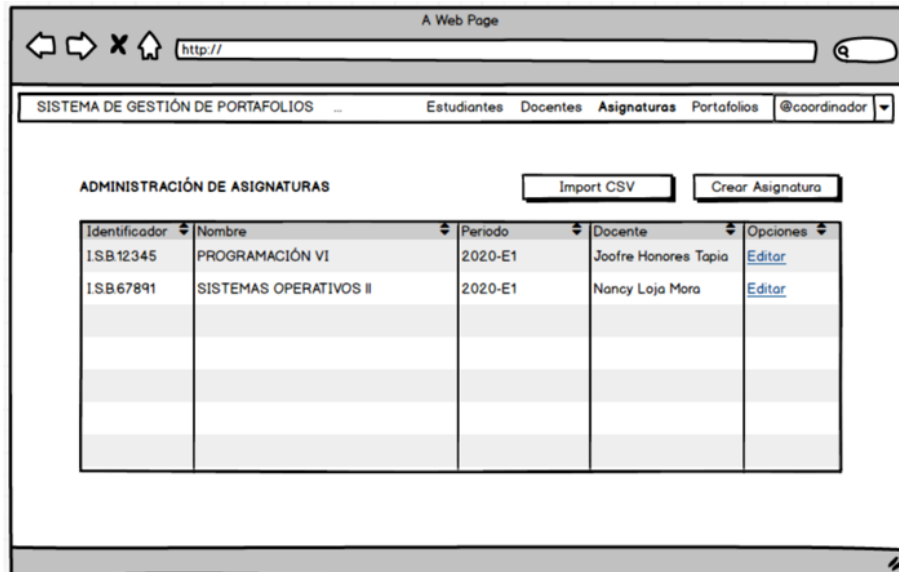
Ilustración 24: Diseño de creación de docentes



Fuente: Elaboración propia

En la **ilustración 24**, se puede crear docentes para la asociación con las asignaturas que imparte; la información solicitada es muy similar a la de los estudiantes.

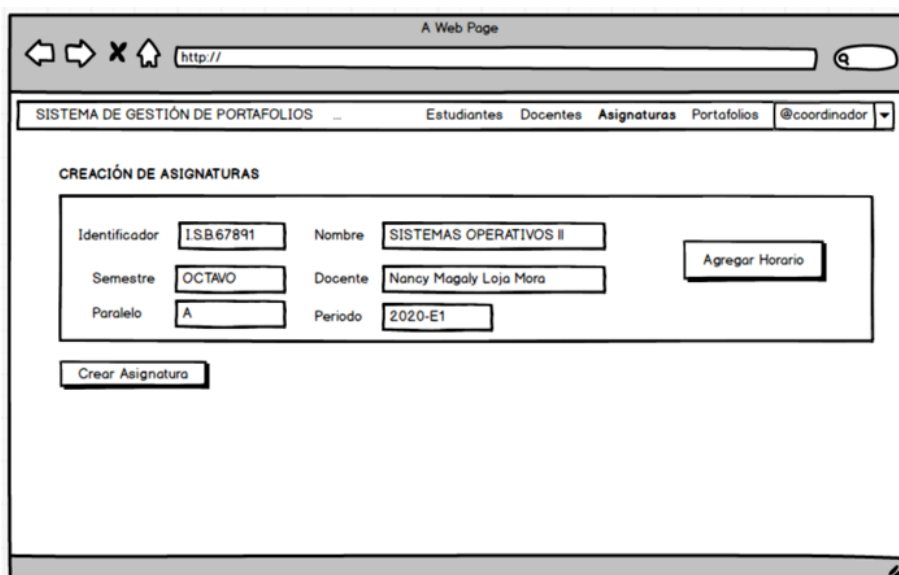
Ilustración 25: Diseño de gestión de asignaturas



Fuente: Elaboración propia

En la **ilustración 25**, se visualiza las asignaturas por la facultad y carrera a la coordinación competente, con opción a modificar ciertos campos académicos.

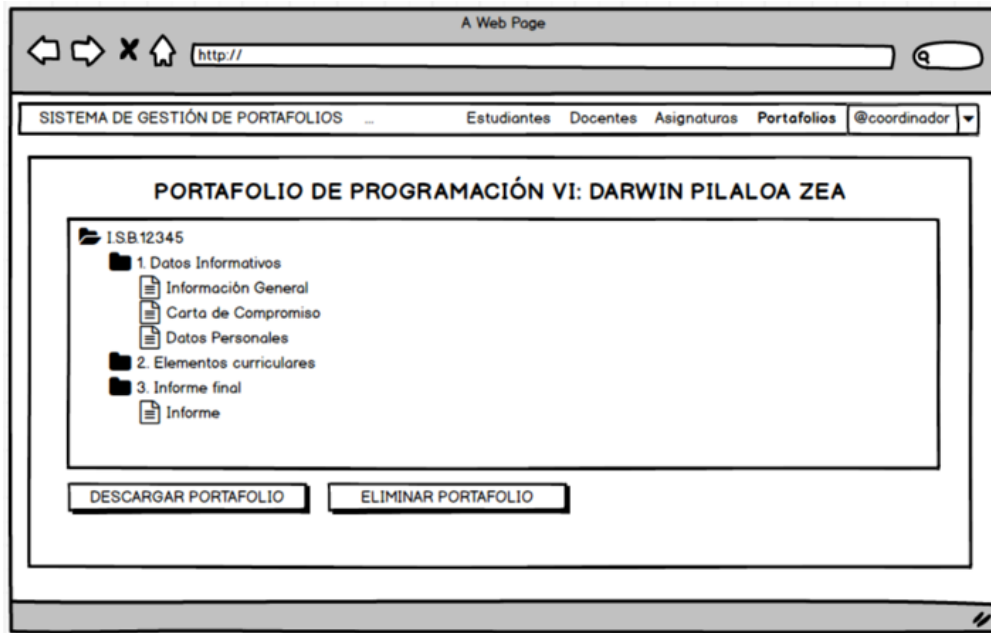
Ilustración 26: Diseño de creación de asignaturas



Fuente: Elaboración propia

En la **ilustración 26**, se crean las asignaturas con base a un periodo, semestre y docente que impartirá dicha asignatura.

Ilustración 27: Diseño del portafolio del estudiante desde el coordinador



Fuente: Elaboración propia

En la **ilustración 27**, se visualiza el portafolio de asignatura de un estudiante en específico perteneciente a una determinada carrera y facultad, con la alternativa de poder descargar la información e incluso borrar dicho portafolio en caso de ser necesario.

2.4.3.5. Diseño de diagramas UML

Los diagramas de “Lenguaje de Modelado Unificado”, facilitan la visualización desglosada de: procesos, clases, atributos, objetos y actores inmersos dentro de un proyecto, a tal punto de entender la forma de interacción de los usuarios con el sistema, siguiendo múltiples condiciones hasta cumplir el objetivo del proceso [41].

Dentro de la propuesta tecnológica, se presentarán los diagramas de caso de uso, que permiten observar la interacción de los diferentes tipos usuarios con los módulos del aplicativo web; además se elaboraron los diagramas de actividades y secuencias, los cuales permiten ver el detalle de las actividades

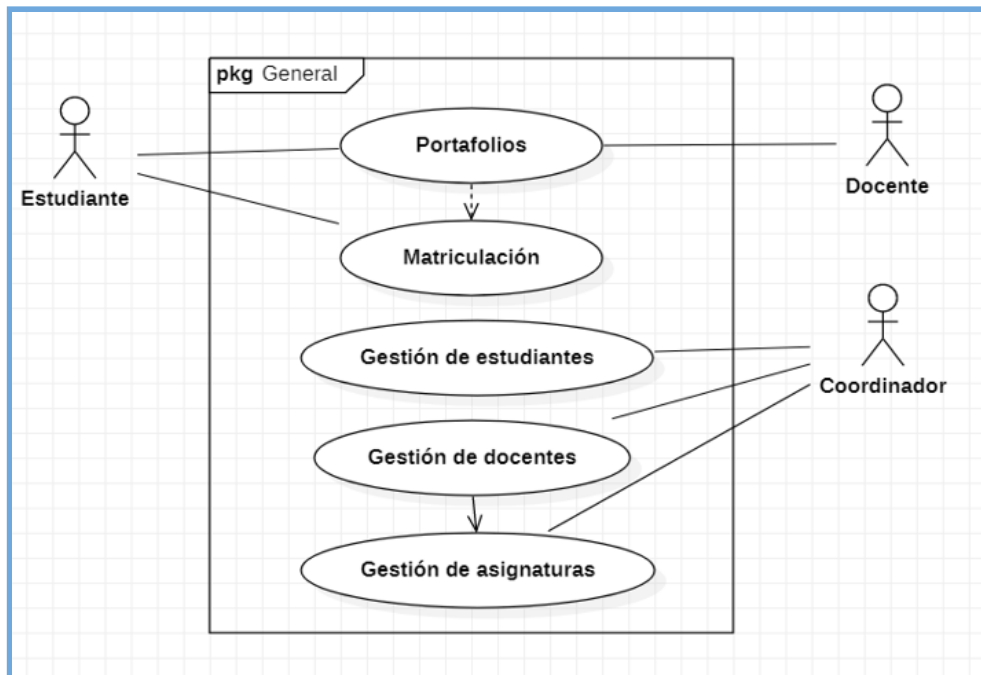
y su respectivo flujo de eventos, con base a una operación hecha por un actor.

2.4.3.5.1. Diagramas de caso de uso

Los diagramas de caso de uso permiten representar de forma visual cada uno de los requerimientos funcionales de la aplicación web a desarrollar, detallando las operaciones de inserción, lectura, eliminación e incluso validación de información [42].

A continuación, se presenta el diagrama de caso de uso general del proyecto, el cual detalla de forma breve el sistema con base a los módulos disponibles.

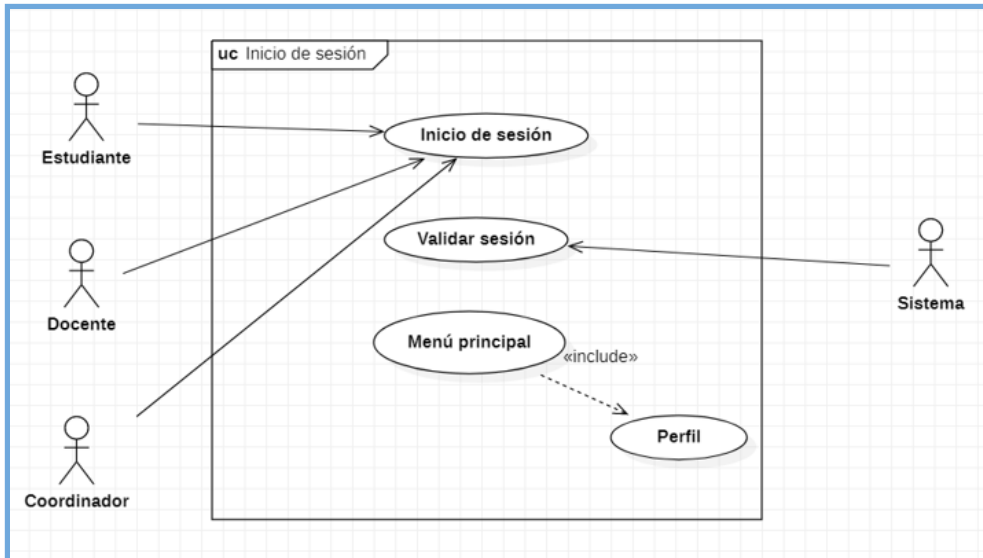
Ilustración 28: Diagrama de caso de uso - General



Fuente: Elaboración propia

En la **ilustración 28**, el estudiante puede realizar procesos de matriculación con respecto a las asignaturas de clases y acceder a los portafolios académicos al igual que los docentes; mientras que el coordinador de carrera tiene módulos específicos dedicados a la gestión académica.

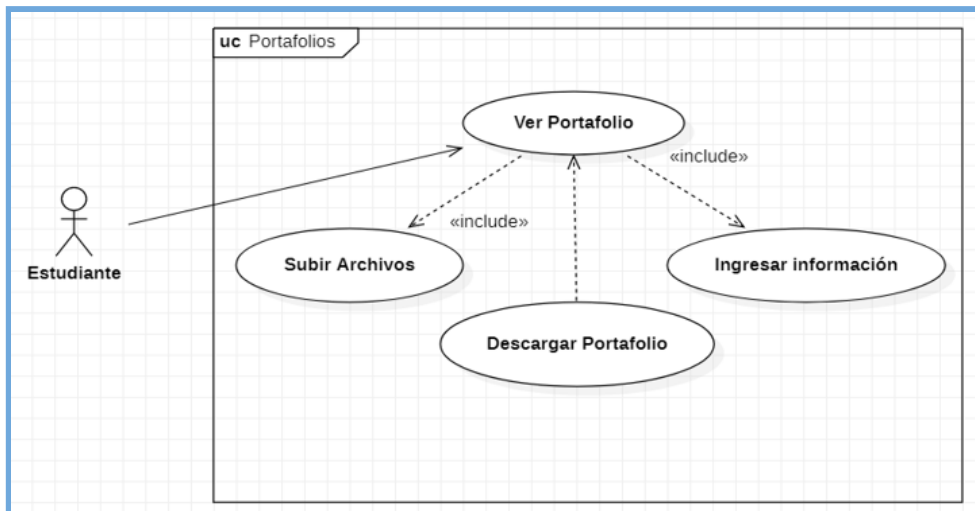
Ilustración 29: Diagrama de caso de uso - Inicio de sesión



Fuente: Elaboración propia

En la **ilustración 29**, se puede determinar que todos los usuarios pueden ingresar a la plataforma web por medio de credenciales que son validadas por el sistema, para tener acceso a las opciones del menú principal e información de sus perfiles basándose en el tipo de usuario.

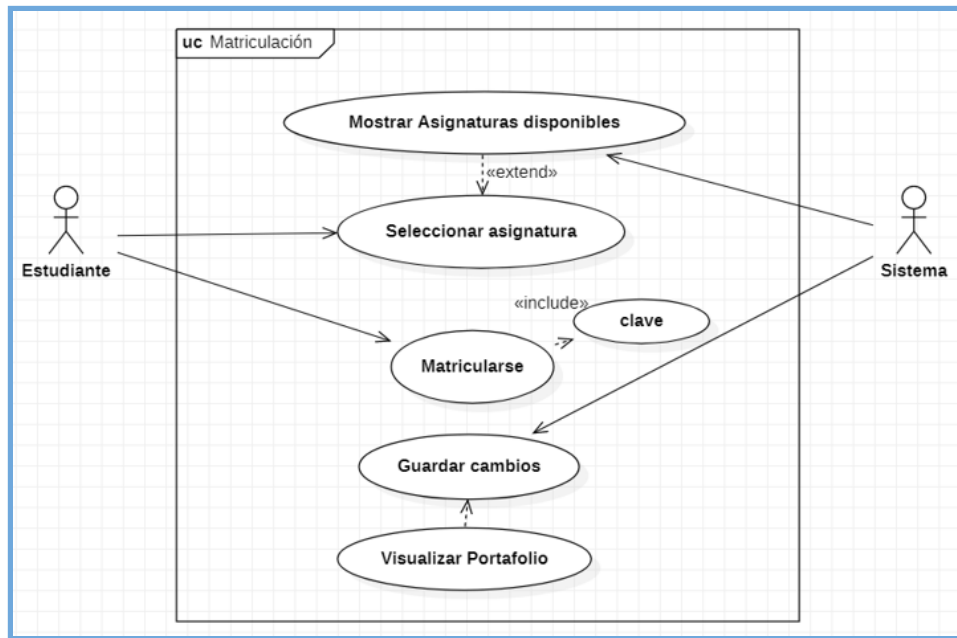
Ilustración 30: Diagrama de caso de uso - Visualización de portafolios



Fuente: Elaboración propia

En la **ilustración 30**, el módulo de visualización de portafolios académicos, del cual solo el estudiante tiene acceso a las distintas opciones de modificación de la información.

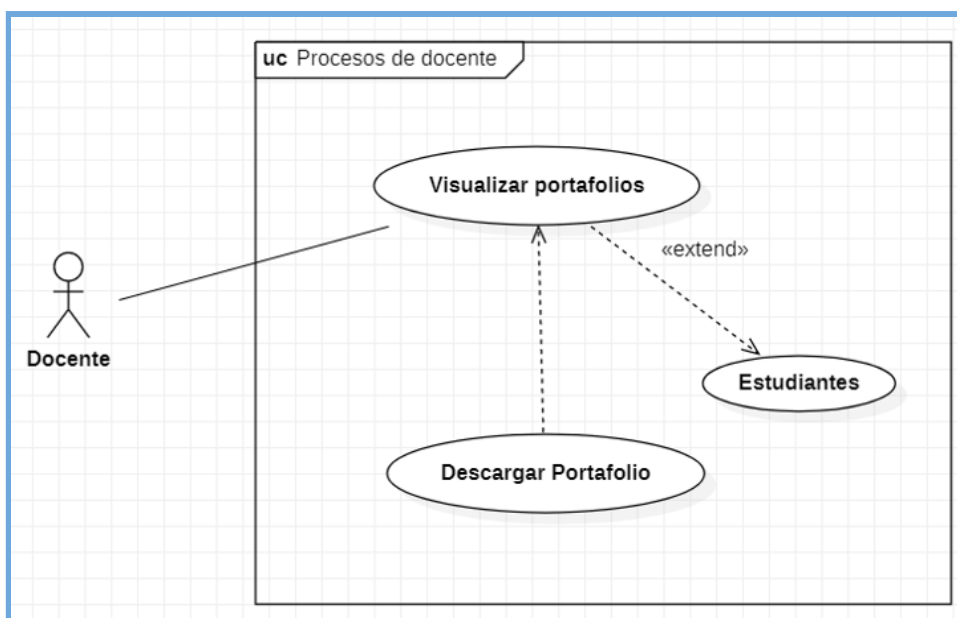
Ilustración 31: Diagrama de caso de uso - Matriculación



Fuente: Elaboración propia

En la **ilustración 31**, el proceso de matriculación se lleva a cabo con la selección de una asignatura impartida por un docente, para proceder a digitar una clase de acceso, el cual se verifica la información a través del sistema, registrándose todos los datos y creando la visualización del portafolio del estudiante.

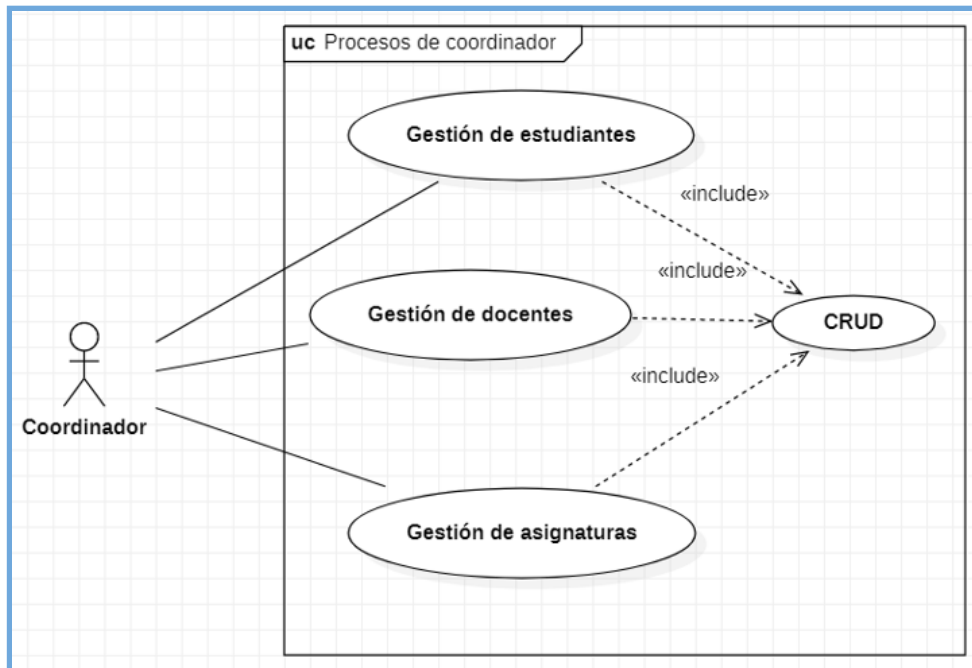
Ilustración 32: Diagrama de caso de uso - Docente



Fuente: Elaboración propia

En la **ilustración 32**, el docente tiene acceso de visualización y descarga de los portafolios de todos los estudiantes que vean la asignatura que imparte, facilitando el proceso de revisión.

Ilustración 33: Diagrama de caso de uso - Coordinador



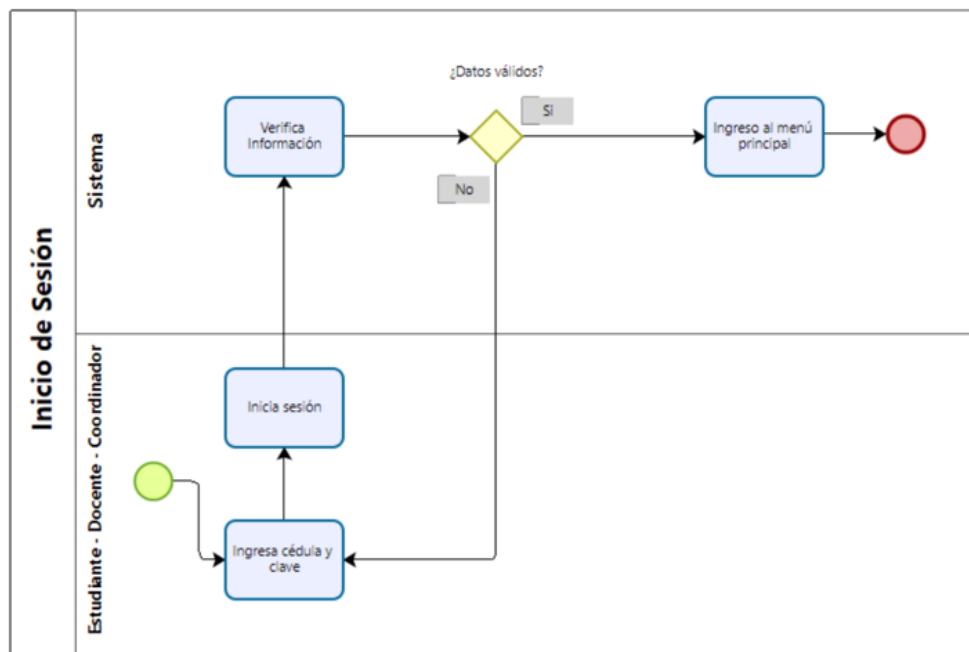
Fuente: Elaboración propia

En la **ilustración 33**, el Coordinador de carrera puede gestionar los distintos módulos de estudiantes, docentes y asignaturas, para poder realizar operaciones C.R.U.D (creación, lectura, actualización y borrado).

2.4.3.5.2. Diagramas de actividades

Los diagramas de actividades muestran en detalle de inicio a fin el intercambio de procesos u actividades de cada módulo del sistema, siguiendo un conjunto de condicionantes y validaciones respectivas, siendo muy parecidos a los “diagramas de flujo” con la diferencia que pueden interactuar múltiples actores con el sistema.

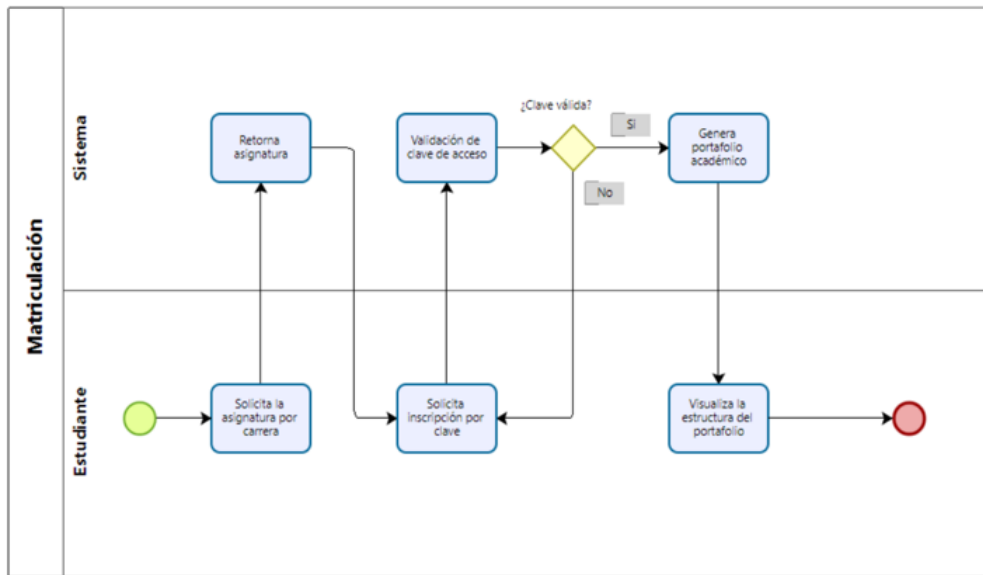
Ilustración 34: Diagrama de actividades - Inicio de Sesión



Fuente: Elaboración propia

En la **ilustración 34**, se observa la validación interna que realiza el sistema para dar paso al acceso a la plataforma según el rol que pertenece.

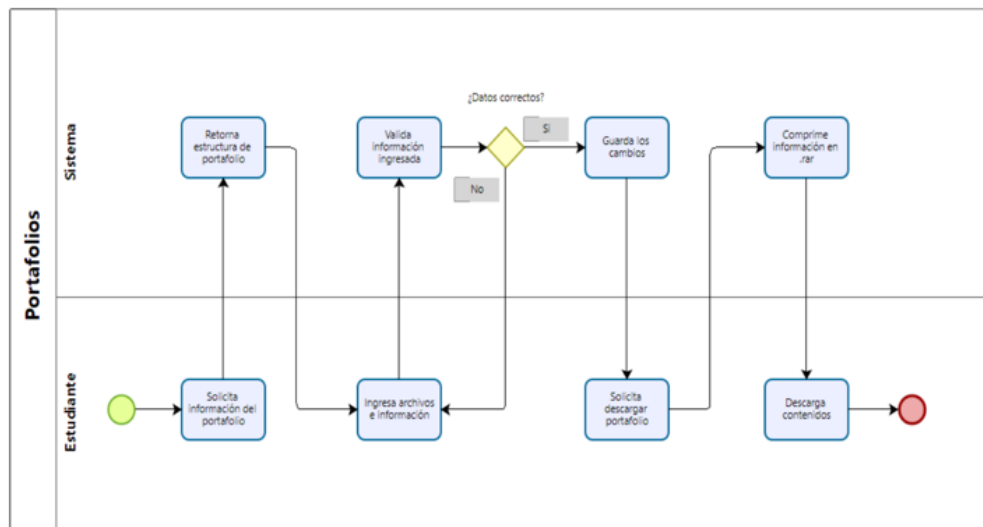
Ilustración 35: Diagrama de actividades - Matriculación



Fuente: Elaboración propia

En la **ilustración 35**, se componen las acciones realizadas para la matriculación de un estudiante a una asignatura y finalmente poder visualizar la estructura de dicho portafolio.

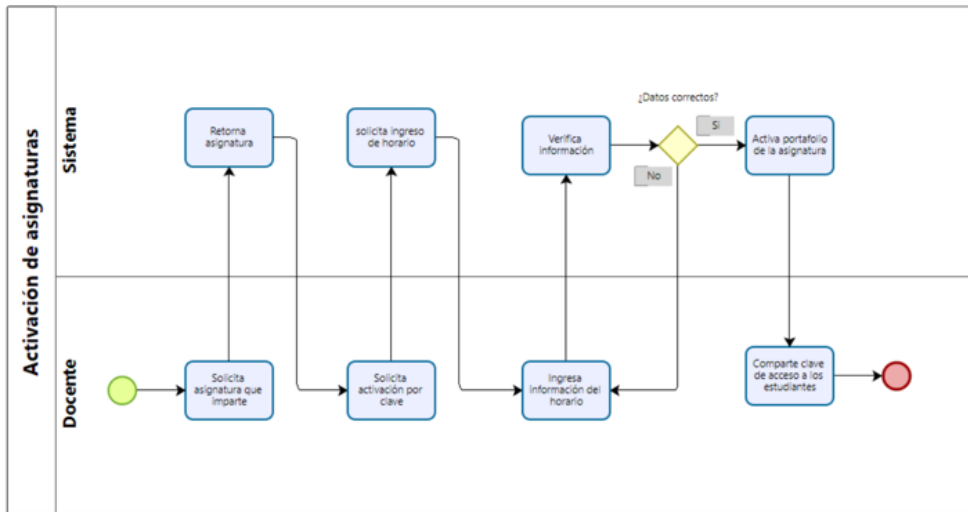
Ilustración 36: Diagrama de actividades - Portafolios



Fuente: Elaboración propia

En la **ilustración 36**, hace referencia a la inserción de información a un determinado portafolio de la asignatura, y proceder a su descarga respectiva.

Ilustración 37: Diagrama de actividades - Asignaturas



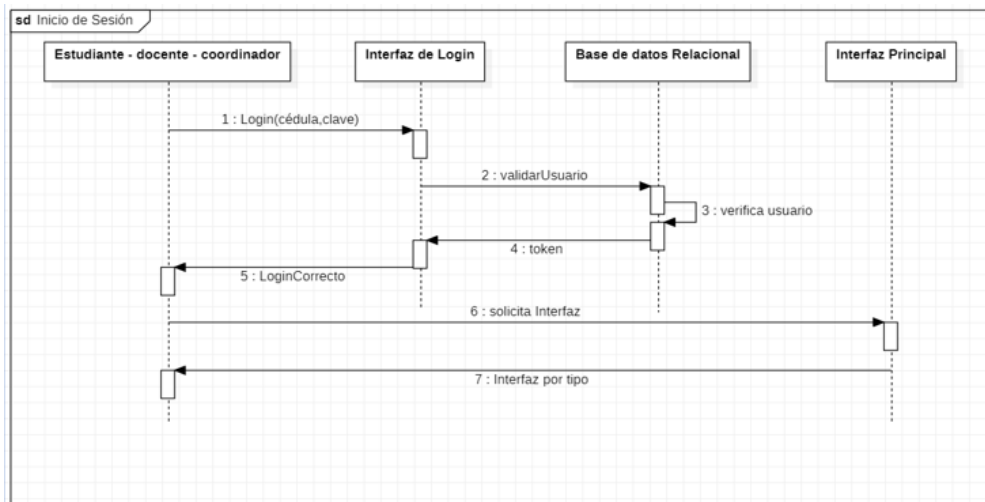
Fuente: Elaboración propia

En la **ilustración 37**, se detalla el proceso que realiza el docente para la activación de la asignatura, mediante el ingreso de una clave (dictada por el) de matriculación y el horario correspondiente en caso de no tener registrado, para luego compartir dicha clave a los estudiantes.

2.4.3.5.3. Diagramas de secuencias

Los diagramas de secuencia sirven para visualizar la interacción de los objetos, definiendo la línea de vida o tiempo de ejecución de los procesos inmersos en cada actividad del sistema.

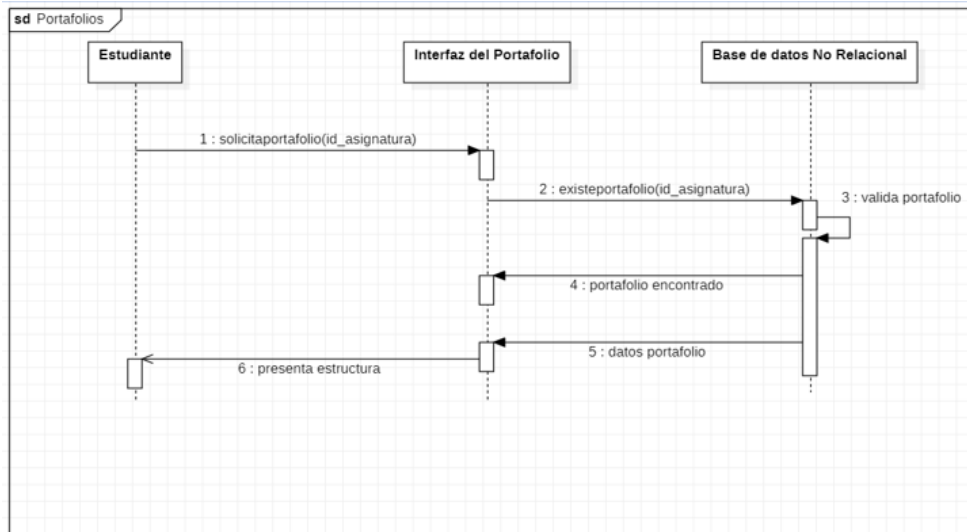
Ilustración 38: Diagrama de secuencia - Inicio de Sesión



Fuente: Elaboración propia

En la **ilustración 38**, se observa que el proceso de validación es a través de una petición a la base de datos relacional y se responde con un token.

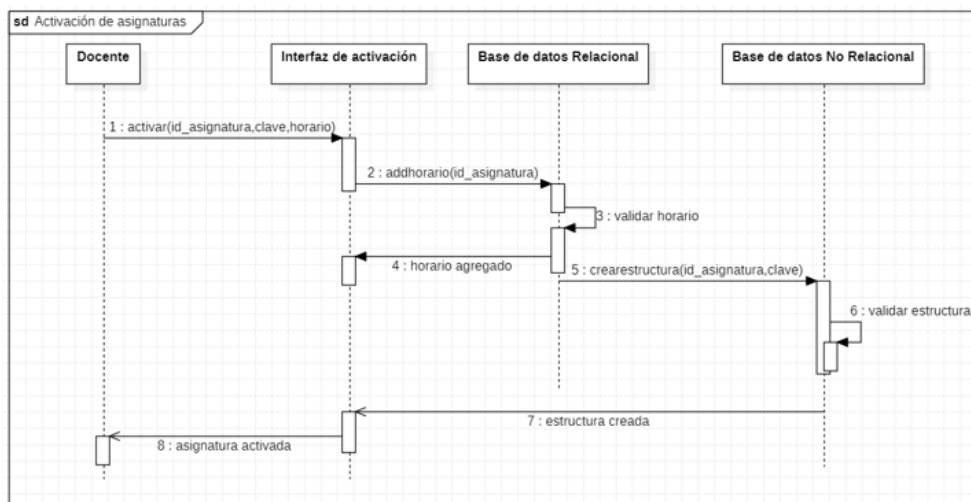
Ilustración 39: Diagrama de secuencia - Portafolios



Fuente: Elaboración propia

En la **ilustración 39**, se determina el proceso que realiza el sistema para la visualización del contenido de un portafolio, a través de una petición a la base de datos no relacional.

Ilustración 40: Diagrama de secuencia - Asignaturas



Fuente: Elaboración propia

En la **ilustración 40**, se observa el proceso de activación de la asignatura, a través de múltiples peticiones enviadas tanto a la base de datos relacional

como a la no relacional, mientras el docente se encuentra a la espera de la respuesta por parte de la interfaz de activación.

2.5. EJECUCIÓN Y/O EMBALAJE DEL PROTOTIPO

El prototipo del proyecto se basa en la visualización resultante de la plataforma web a implementar, por lo tanto, se presentarán los módulos ya implementados a través de las tecnologías y arquitectura anteriormente mencionadas (**punto 2.1.**), cumpliendo con el objetivo de cumplimiento de los requerimientos del proyecto.

2.5.1. Página de bienvenida y sin restricción

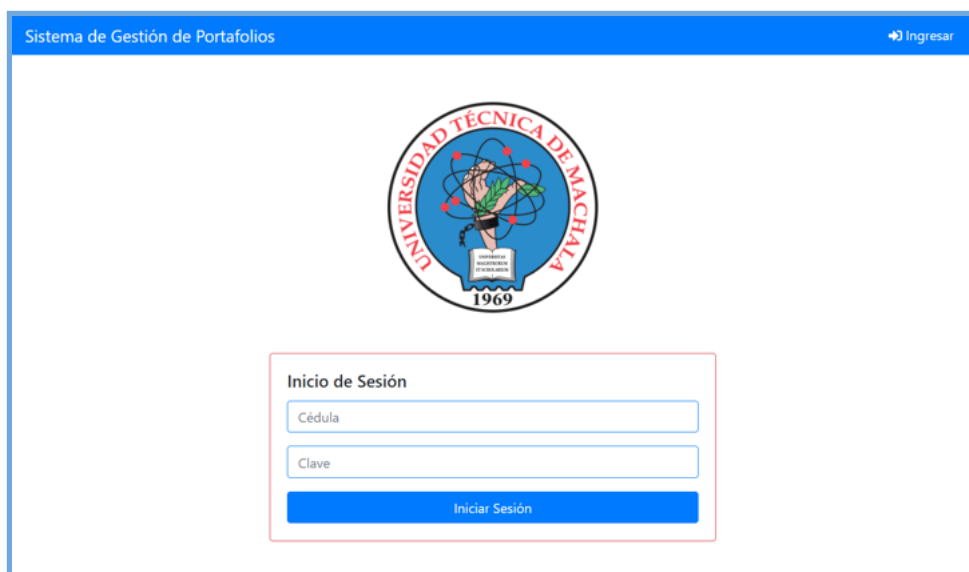
Ilustración 41: Implementación - Página de bienvenida



Fuente: Elaboración propia

La página de bienvenida contiene información sobre la facultad en la que fue desarrollado el sistema y su finalidad como proyecto.

Ilustración 42: Implementación - Inicio de Sesión



Fuente: Elaboración propia

La página de iniciar sesión contiene una autenticación simple y segura, el cual solicita la cédula y la clave del estudiante, para acceder a las funcionalidades del sistema.

2.5.2. Páginas de Estudiantes

Las páginas presentadas a continuación, son para los estudiantes que se inicien sesión en la plataforma, con sus respectivas credenciales.

Ilustración 43: Implementación - Menú principal del estudiante



Fuente: Elaboración propia

En la **ilustración 43**, se muestran todas las carreras que contiene la Facultad de Ingeniería Civil, y al seleccionar una de ellas se presentan las asignaturas habilitadas en el período actual, con la respectiva información del docente y

en caso de no estar matriculado en una asignatura se procede a colocar la clave de acceso y dar clic en el botón “Matricularme”.

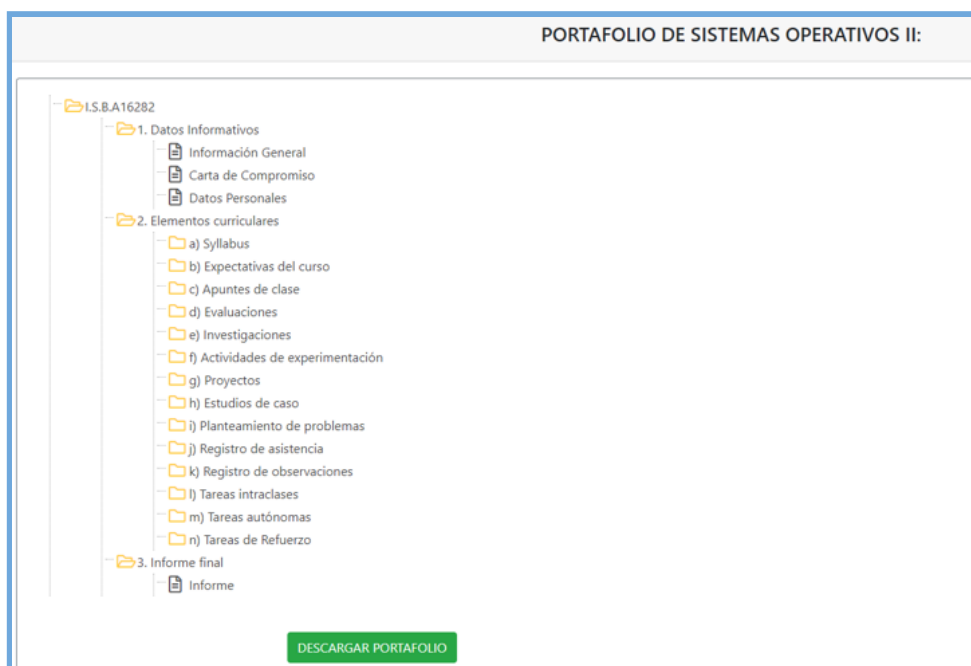
Ilustración 44: Implementación - Menú portafolios del estudiante



Fuente: Elaboración propia

En la **ilustración 44**, se presenta de forma detallada los portafolios de las asignaturas que ha sido matriculado el estudiante durante un período académico determinado.

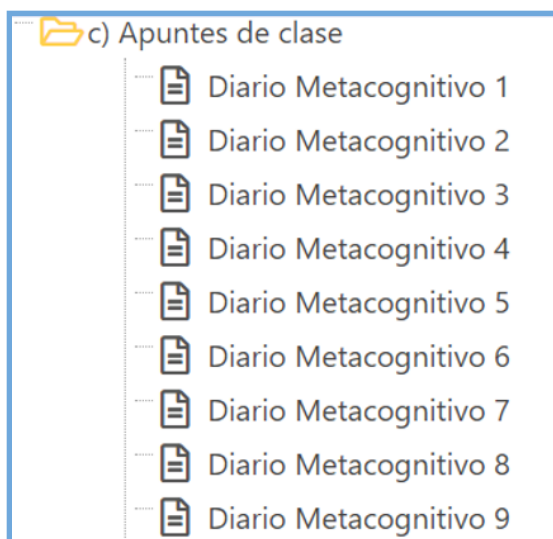
Ilustración 45: Implementación - Portafolio del estudiante



Fuente: Elaboración propia

En la **ilustración 45**, se visualiza la estructura del portafolio académico, con desglose por carpetas para poder subir los archivos y complementar la información faltante, a fin de ser descargado en caso de ser solicitado.

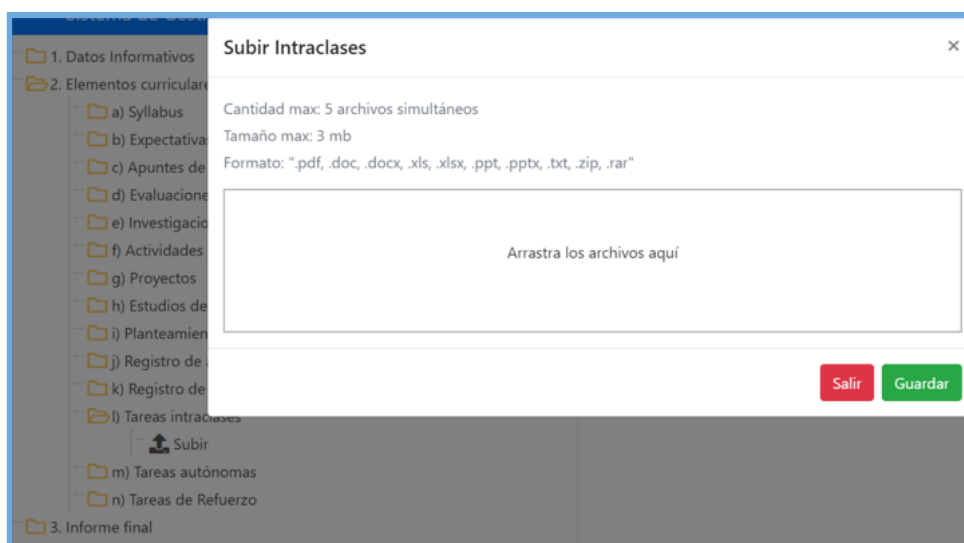
Ilustración 46: Implementación - Diarios metacognitivos



Fuente: Elaboración propia

En la **ilustración 46**, se detalla la generación de los diarios metacognitivos, basándose en el horario de clases de la asignatura y las fechas establecidas en el syllabus de clases.

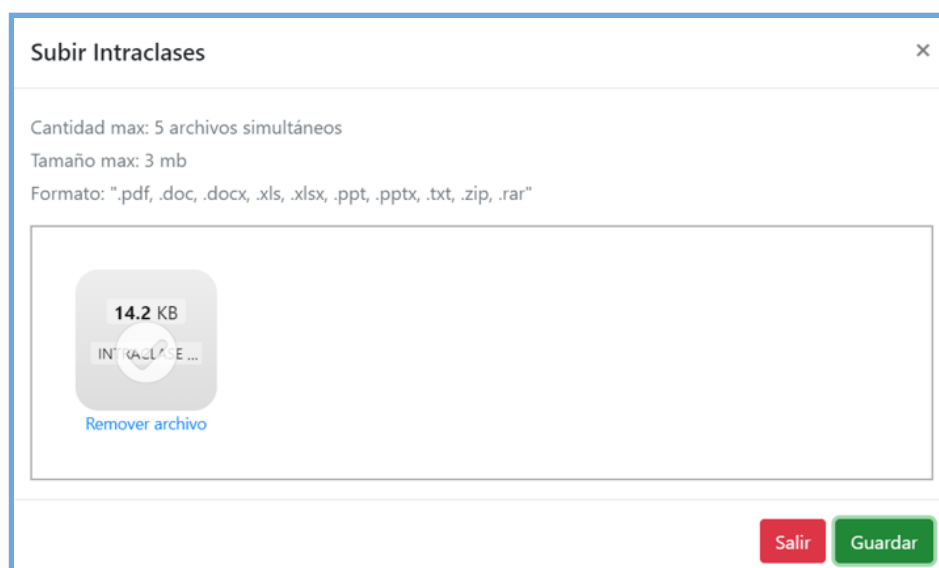
Ilustración 47: Implementación - Subir archivos al portafolio



Fuente: Elaboración propia

En la **ilustración 47**, se muestra el proceso para la subida de archivos al sistema con el detalle de las validaciones respectivas para controlar el volumen y tipo de información que ingresa al portafolio.

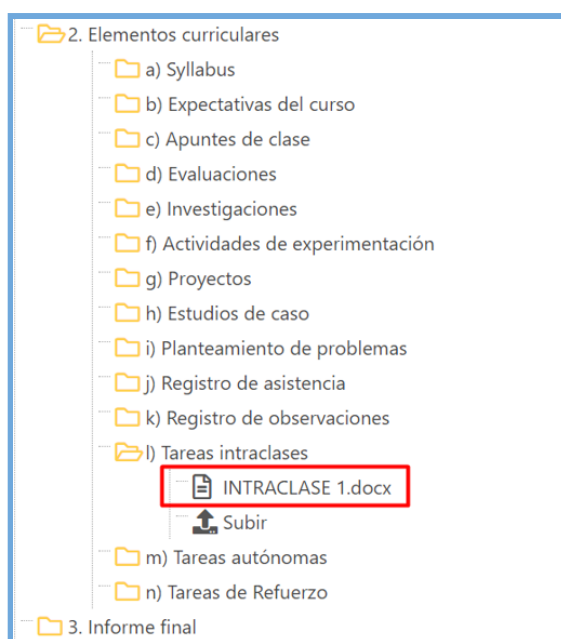
Ilustración 48: Implementación - Validación de subida de archivos



Fuente: Elaboración propia

En la **ilustración 48**, el sistema valida que el archivo subido sea el correcto, para proceder a su respectivo guardado.

Ilustración 49: Implementación - Comprobación de archivos



Fuente: Elaboración propia

En la **ilustración 49**, se comprueba que el archivo subido se encuentre visible dentro del portafolio académico.

Ilustración 50: Implementación - Perfil del estudiante



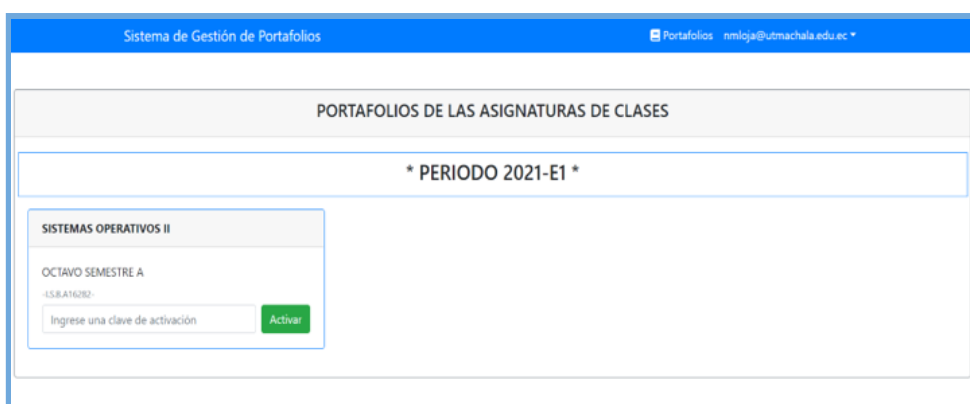
Fuente: Elaboración propia

En la **ilustración 50**, se presenta el perfil del usuario actualmente activo, en este caso es de tipo estudiante.

2.5.3. Páginas de Docentes

Las páginas presentadas a continuación, corresponden a las vistas de los docentes que inicien sesión en la plataforma, con sus respectivas credenciales.

Ilustración 51: Implementación - Menú portafolios de docentes

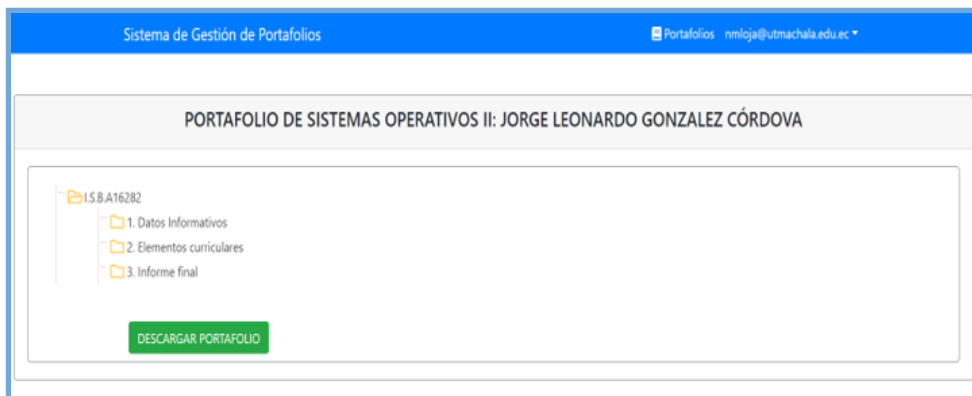


Fuente: Elaboración propia

Como se puede observar en la **ilustración 51**, los docentes solo pueden observar los portafolios de las asignaturas que ellos imparten, y en este caso

debe habilitar activar la asignatura con una clave de acceso para que los estudiantes puedan matricularse.

Ilustración 52: Implementación - Portafolio desde la vista del docente



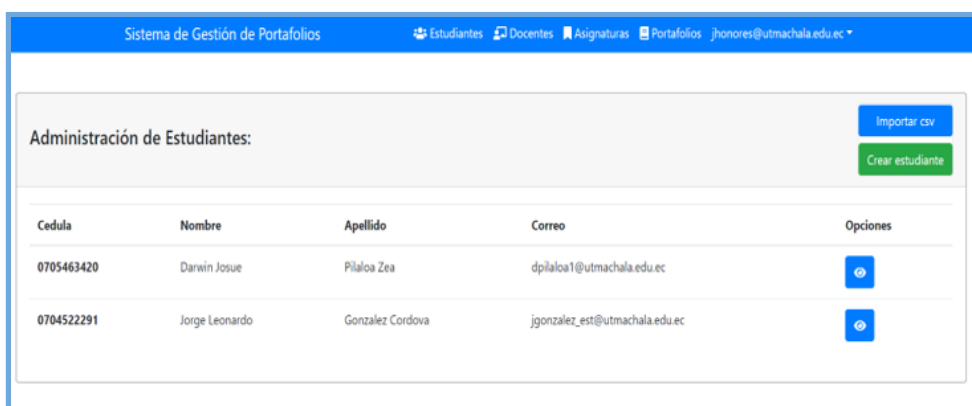
Fuente: Elaboración propia

En la **ilustración 52**, se presenta la información del portafolio de la asignatura de un estudiante en específico, el cual el docente tiene acceso al modo de lectura y descarga del mismo.

2.5.4. Páginas de Coordinador de carrera

Las páginas presentadas a continuación, corresponden a las vistas de los Coordinador de carrera de las distintas carreras que inicien sesión en la plataforma; cabe recalcar que cada Coordinador es responsable de la información de la carrera que dirige.

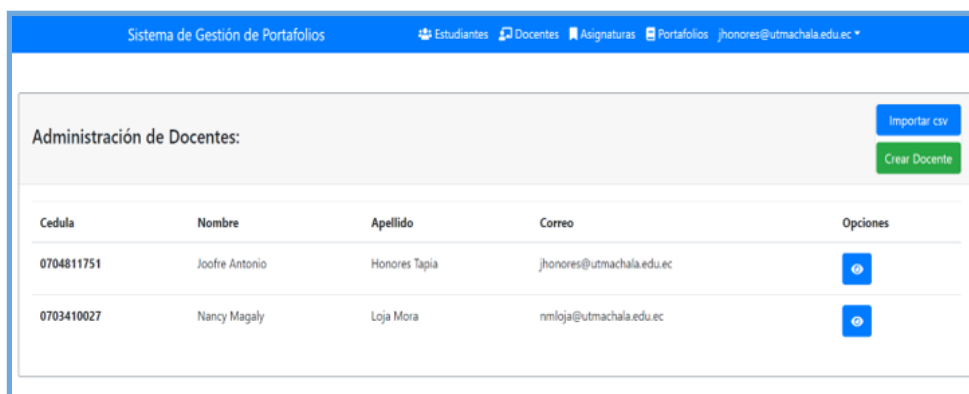
Ilustración 53: Implementación - Gestión de estudiantes





Fuente: Elaboración propia

En la **ilustración 53**, se puede gestionar la información de los estudiantes que forman parte de la carrera que dirige, con opción a subir archivos de tipo “.csv”, para la actualización de los datos.

Ilustración 54: Implementación - Gestión de docentes



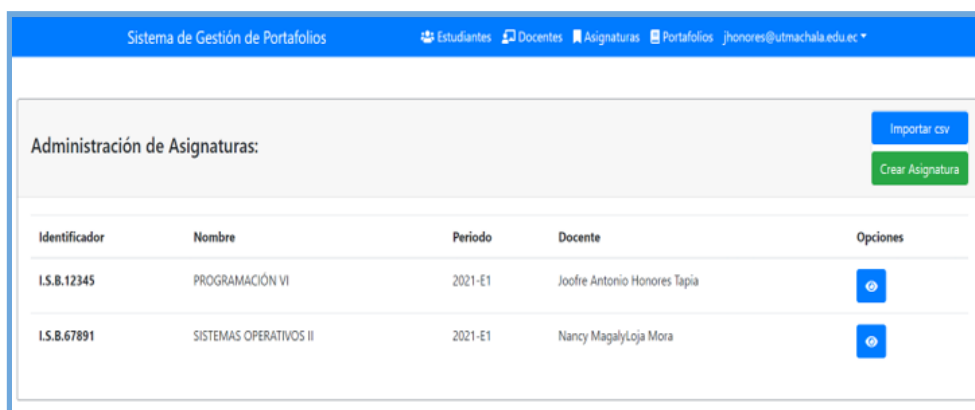
The screenshot shows a web interface for 'Administración de Docentes'. At the top, there is a navigation bar with 'Estudiantes', 'Docentes', 'Asignaturas', and 'Portafolios' menus, and a user profile 'jhonores@utmachala.edu.ec'. Below the navigation bar, there are two buttons: 'Importar csv' (blue) and 'Crear Docente' (green). The main content area contains a table with the following data:

Cedula	Nombre	Apellido	Correo	Opciones
0704811751	Joffre Antonio	Honores Tapia	jhonores@utmachala.edu.ec	
0703410027	Nancy Magaly	Loja Mora	nmloja@utmachala.edu.ec	



Fuente: Elaboración propia

En la **ilustración 54**, se puede gestionar la información respectiva a los docentes que imparten las distintas asignaturas en la carrera, con opción a subir archivos de tipo “.csv”, para la actualización de los datos.

Ilustración 55: Implementación - Gestión de asignaturas



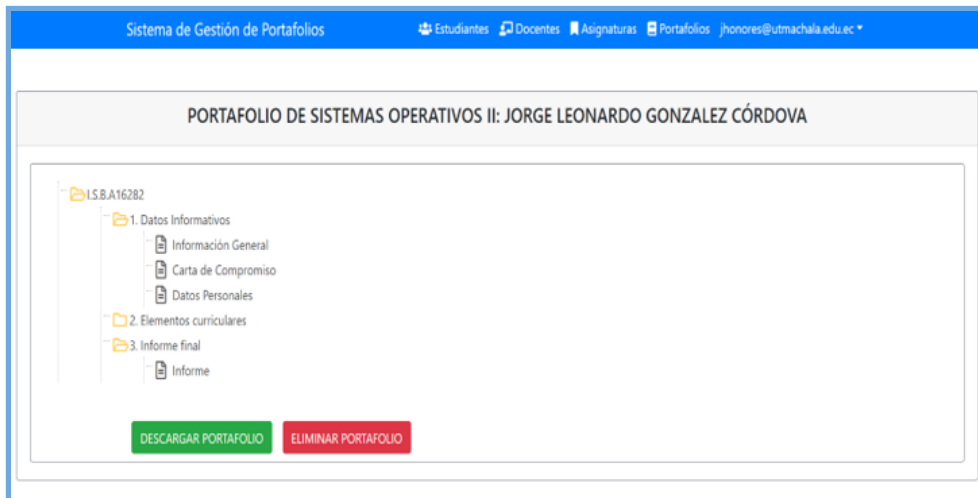
The screenshot shows a web interface for 'Administración de Asignaturas'. At the top, there is a navigation bar with 'Estudiantes', 'Docentes', 'Asignaturas', and 'Portafolios' menus, and a user profile 'jhonores@utmachala.edu.ec'. Below the navigation bar, there are two buttons: 'Importar csv' (blue) and 'Crear Asignatura' (green). The main content area contains a table with the following data:

Identificador	Nombre	Periodo	Docente	Opciones
I.S.B.12345	PROGRAMACIÓN VI	2021-E1	Joffre Antonio Honores Tapia	
I.S.B.67891	SISTEMAS OPERATIVOS II	2021-E1	Nancy MagalyLoja Mora	

Fuente: Elaboración propia

En la **ilustración 55**, se puede gestionar la información respectiva a las asignaturas impartidas en la carrera, asociándose a los docentes en un periodo determinado.

Ilustración 56: Implementación - Portafolio desde la vista del coordinador



Fuente: Elaboración propia

En la **ilustración 56**, el Coordinador de carrera puede visualizar el portafolio de un estudiante específico, y es el único con la opción de dar de baja a los portafolios académicos, en caso de ser extremadamente necesario.

3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. PLAN DE EVALUACIÓN

La metodología “SWIRL” implementada en la propuesta del proyecto, define tres formas de evaluación para determinar si el sistema desarrollado es de calidad, a través de: la verificación de métricas de calidad, evaluación de las tendencias de diseño y herramientas para medir el nivel de SEO de la aplicación.

3.2. RESULTADOS DE EVALUACIÓN

3.2.1. Evaluación de calidad de la aplicación web

Para la evaluación de calidad se utilizó como base la norma ISO/IEC 9126, el cual subdivide en categorías las características para medir el nivel de calidad del sistema, siguiendo los parámetros de funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad [43].

A continuación, se presentará una matriz de los parámetros a evaluar, con las herramientas web dedicadas al análisis de calidad, cabe destacar que la plataforma no posee un dominio, por lo que usará su IP pública para la evaluación (190.155.140.58:3000) y no todas las herramientas permiten evaluar sin un dominio propio:

Tabla 20: Evaluación de calidad ISO/IEC 9126

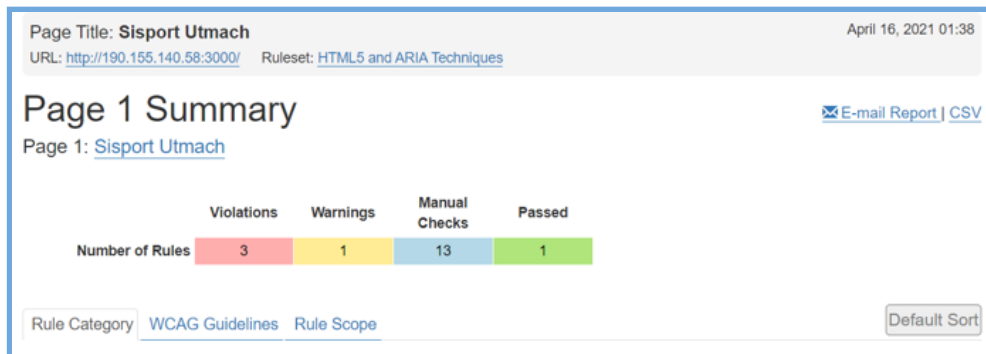
EVALUACIÓN DEL ESTÁNDAR ISO/IEC 9126	Funcionalidad	Fiabilidad	Eficiencia	Mantenibilidad
Run FAE	x			
GTmetrix		x		
W3C Validator			x	x

Fuente: Elaboración propia

3.2.1.1. Run FAE

Esta herramienta web determina el nivel de funcionalidad del sitio, encontrando posibles advertencias y errores posibles en la codificación.

Ilustración 57: Evaluación de seguridad - Run FAE



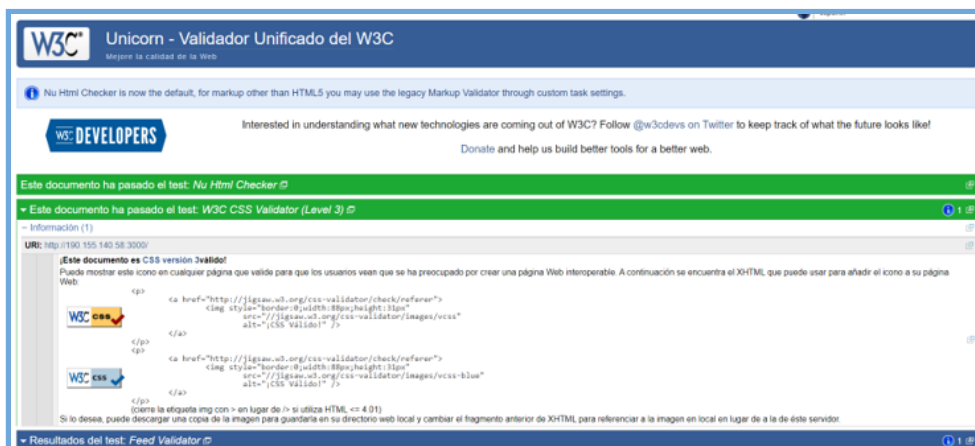
Fuente: Elaboración propia

Como observamos en la **ilustración 57**, los resultados son regulares, pero con aprobación, debido a que el sitio posee restricciones de acceso a los servicios web.

3.2.1.2. W3C Validator

Es una aplicación web dedicada a la evaluación de la estructura HTML de páginas web, a través de parámetros de desarrollo, encontrando posibles vulnerabilidades en códigos JavaScript, CSS, etc., a continuación, se presentan los resultados.

Ilustración 58: Evaluación de seguridad - W3C Validator



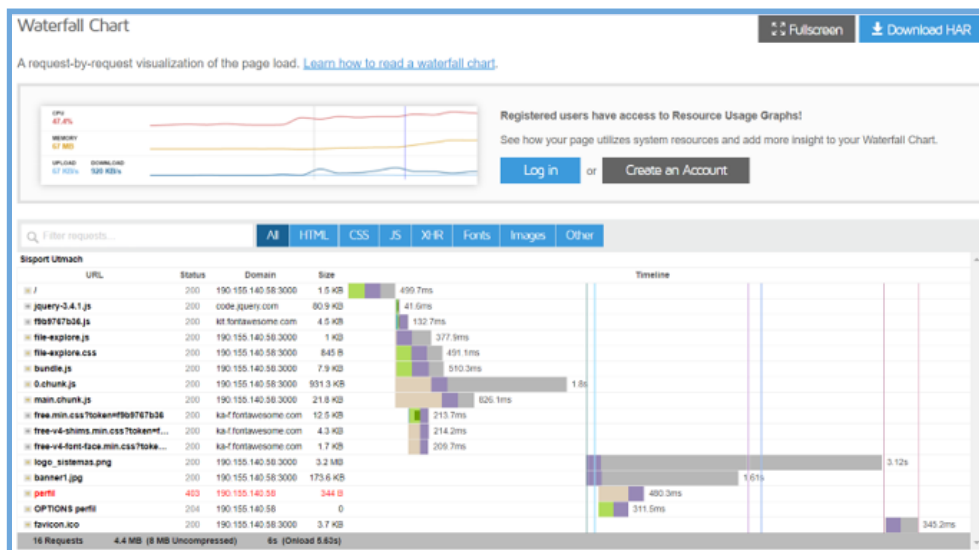
Fuente: Elaboración propia

Como observamos en la **ilustración 58**, la prueba fue exitosa, sin ningún error a nivel de estructura del sitio web.

3.2.1.3. GTmetrix

Esta aplicación determina el grado de fiabilidad del sitio web a través de los tiempos de ejecución entre el navegador y el servidor de aplicación, permitiendo tener una idea clara sobre los tiempos de velocidad.

Ilustración 59: Evaluación de seguridad - GTmetrix



Fuente: Elaboración propia

En la **ilustración 59**, se puede determinar que los tiempos de ejecución son regulares, debido a que se utiliza un servidor privado de pruebas, con limitaciones de ancho de banda y velocidad de internet.

Muchas de las herramientas no permiten evaluar índices como la usabilidad y portabilidad del sitio web por falta de un dominio privado, por lo tanto, estos parámetros se pueden analizar con base a la observación y experiencia de los usuarios con la aplicación.

A continuación, se presenta una tabla de la forma idónea para la evaluación de la usabilidad y portabilidad:

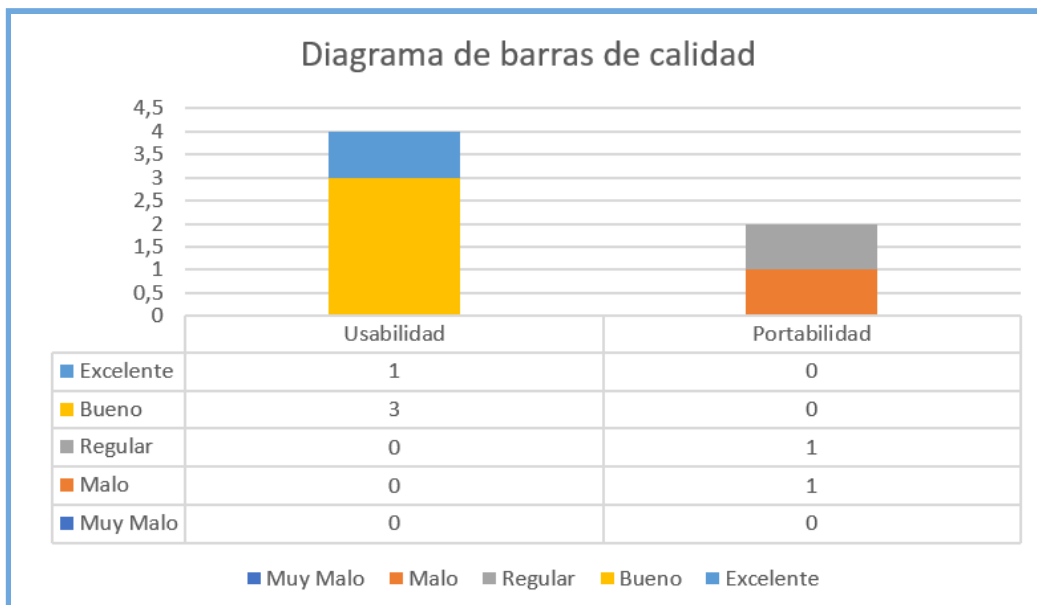
Tabla 21: Evaluación de calidad - usabilidad y portabilidad

EVALUACIÓN CON BASE A LA NORMA ISO/IEC 9126		Muy Malo	Malo	Regular	Bueno	Excelente
Característica	Módulo	1	2	3	4	5
Usabilidad	Aprendizaje					x
	Comprensión				x	
	Operatividad				x	
	Atractividad				x	
Portabilidad	Capacidad de instalación		x			
	Capacidad de reemplazamiento			x		

Fuente: Elaboración propia

Una vez realizada la evaluación se puede elaborar un gráfico de barras con las fortalezas y debilidades detectadas.

Ilustración 60: Diagrama de barras - usabilidad y portabilidad



Fuente: Elaboración propia

3.3. CONCLUSIONES

- Se desarrolló una plataforma web funcional satisfaciendo la necesidad de gestionar los portafolios académicos de los estudiantes de la Universidad Técnica de Machala, específicamente la Facultad de Ingeniería Civil, logrando la correcta implementación de una arquitectura de hardware eficiente y múltiples lenguajes de programación, usando la metodología de desarrollo ágil “SWIRL”.
- Es fundamental la iniciativa de implementación de nuevas tecnologías para el desarrollo web, por las múltiples herramientas que ofrecen para elaborar sistemas más seguros, potentes, complejos y sobre todo que soporta arquitecturas de tipo empresarial.
- “SWIRL” es una metodología que facilita el desarrollo eficiente y eficaz de los proyectos de software web, por medio de la implementación de cada una de sus fases, dando pautas para futuras iteraciones que logren realizar mejoras en cualquier módulo de la plataforma desarrollada.
- La recolección de requisitos es el proceso primordial para lograr definir la trayectoria y el alcance del proyecto, a través de las limitaciones del software.
- Las herramientas CASE forman un papel muy importante para gestionar la correcta elaboración de proyectos informáticos, permitiendo anticiparse a futuros acontecimientos dentro de la aplicación, y sobre todo posee una interfaz didáctica, fácil de entender y explicar a todos interesados del proyecto.
- La programación estructurada y complementaria permite que aplicaciones web de alto nivel de gestión de módulos, puedan ser analizados sin dificultad alguna.
- La evaluación de la plataforma web utilizando un estándar internacional de calidad, garantiza que el sistema cumple con las recomendaciones, evaluaciones y acciones necesarias para la correcta gestión de la información.

3.4. RECOMENDACIONES

Con base a la experiencia en programación, utilizando múltiples lenguajes de desarrollo web, considero oportuno recomendar lo siguiente:

- Implementar una metodología de desarrollo ágil como lo es “SWIRL”, para elaborar proyectos de software de calidad, en un tiempo relativamente corto.
- Utilizar siempre lenguajes de desarrollo de “Back-end” simples, potentes y flexibles, que permitan aprovechar correctamente los recursos del equipo o servidor, además de la importancia de que exista facilidad comunicación e interacción con otras plataformas o servicios.
- Utilizar frameworks de diseño como lo es “Bootstrap”, reduce el proceso de elaboración de plantillas personalizadas, permitiendo enfocar el tiempo en el desarrollo operacional del sistema.
- Se debe utilizar bases de datos “no relacionales” para proyectos a gran escala, debido a que pueden soportar grandes cantidades de datos y consultas sin afectar la integridad de la información
- Realizar siempre pruebas unitarias del sistema, reduciendo así el número de errores posibles por cada funcionalidad implementada dentro del sistema.

BIBLIOGRAFÍA

- [1] H. Contreras y M. A. Cortina, «La Nueva Realidad de la Educación Superior como derecho humano ante el COVID-19,» *Scielo*, vol. 8, nº 9, 2021.
- [2] M. Agudelo, E. Chomali y J. Suniaga, «LAS OPORTUNIDADES DE LA DIGITALIZACIÓN DE AMÉRICA LATINA FRENTE AL COVID-19,» 2020. [En línea]. Available: https://repositorio.cepal.org/bitstream/handle/11362/45360/4/OportDigitalizaCovid-19_es.pdf. [Último acceso: 20 Septiembre 2020].
- [3] J. Molina y M. d. I. n. Pedreira, “SWIRL”, METODOLOGÍA PARA EL DISEÑO Y DESARROLLO DE APLICACIONES WEB, España: 3Ciencias, 2019.
- [4] J. García y M. Rojas, «El portafolio de evidencias del alumno,» *Dialnet*, vol. 3, nº 6, pp. 177-190, 2018.
- [5] L. Dino, «El Portafolio de evidencias como una modalidad de titulación en las escuelas normales,» *Scielo*, vol. 8, nº 14, pp. 69-90, 2017.
- [6] J. Medina, E. Pineda y F. Reynaldo, «Requerimientos de software: prototipado, software heredado y análisis de documentos,» *Scielo*, vol. 37, nº 2, p. 6, 2019.
- [7] J. d. D. Murillo y S. Caamaño, «IMPLEMENTACIÓN DE UN SERVIDOR FTP UTILIZANDO EL MODELO CLIENTE/SERVIDOR MEDIANTE EL USO DE SOCKETS EN LENGUAJE C UNIX CON EL FIN DE MEJORAR LOS TIEMPOS DE RESPUESTA EN LA RED,» *Redalyc*, nº 24, pp. 83-89, 2010.
- [8] J. Murillo y S. Caamaño, «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.
- [9] A. Rodríguez y L. Silva, «Arquitectura de software para el sistema de visualización médica Vismedic,» *Scielo*, vol. 8, nº 1, pp. 75-86, 2016.
- [10] S. Domingo, H. Vega y L. Guerra, «Programación en N capas,» *RISI*, vol. 7, nº 2, pp. 57-67, 2010.
- [11] CentOS, «The CentOS Project,» 2021. [En línea]. Available: <https://www.centos.org/>. [Último acceso: Febero 2021].
- [12] Z. Rodriguez, L. D. P. Rodríguez y J. C. Gonzales, «Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua,» *Redalyc*, vol. 23, nº 2, p. 9, 2020.
- [13] J. Zumba, «Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software,» *Dialnet*, vol. 3, nº 10, pp. 20-33, 2018.
- [14] J. Crespo, 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.
- [15] J. Pereira y R. Russo, «Design Thinking Integrated in Agile Software Development: A Systematic Literature Review,» *ScienceDirect*, vol. 138, pp. 775-782, 2018.
- [16] M. Santos y N. Escudero, «Propuesta de metodología híbrida y base de documentación para el desarrollo de software actual,» *Dialnet*, nº 60, p. 17, 2020.

- [17] 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.,» *Redalyc*, vol. 12, nº 1, p. 10, 2018.
- [18] D. Bastidas, O. Espíndola y A. Palma, «Implementación del bootstrap como una metodología ágil en la web,» *Dialnet*, vol. 5, nº 268-286, p. 19, 2019.
- [19] J. Briede, M. Cabello, C. Pérez y A. Arriagada, «Plataforma Colaborativa para la Gestión de Proyectos de Diseño Industrial,» *Scielo*, vol. 9, nº 3, p. 7, 2016.
- [20] R. O. Source, «reactjs.org,» Facebook, Enero 2021. [En línea]. Available: <https://es.reactjs.org/>. [Último acceso: Febrero 2021].
- [21] N. Palma, «Computer solution for the selection of the web server during the migration to open source,» *Redalyc*, vol. 14, nº 2, p. 9, 2020.
- [22] N. Palma, «Solución informática para la selección del servidor web durante la migración a código abierto,» *Scielo*, vol. 14, nº 2, pp. 49-69, 2020.
- [23] A. Vera y J. Camargo, «Herramienta informática para notificación comunitaria como insumo para la generación de alertas en seguridad alimentaria y nutricional,» *Redalyc*, vol. 22, nº 45, p. 36, 2019.
- [24] Flask, «flask.palletsprojects.com,» Flask development, 2010. [En línea]. Available: <https://flask.palletsprojects.com/>. [Último acceso: Febrero 2021].
- [25] M. Zea, J. Honores y W. Rivas, Fundamentos de bases de datos, Machala: UTMACH, 2015.
- [26] D. Rodríguez, R. Reyes y B. Chang, «Implementación de una Base de Datos Relacional para la Aplicación BEHIQUE SIC,» *Scielo*, vol. 21, nº 3, pp. 78-85, 2017.
- [27] M. Zea, J. Molina y F. Redrován, «Administración de base de datos con PostgreSQL,» *Dialnet*, p. 82, 2017.
- [28] K. Hauger y M. Olivier, «NoSQL databases: forensic attribution implications,» *Scielo*, vol. 109, nº 2, pp. 119-132, 2018.
- [29] F. Moreno, J. Quintero y R. Rueda, «UNA COMPARACIÓN DE RENDIMIENTO ENTRE ORACLE Y MONGODB,» *Scielo*, vol. 26, nº 1, p. 5, 2016.
- [30] G. Benigni y J. Celada, «Moomh case. Herramienta automatizada para la ayuda al desarrollo de sistemas hipermediales,» *Scielo*, vol. 23, nº 4, p. 10, 2008.
- [31] M. Visio, «www.microsoft.com,» Microsoft, 2021. [En línea]. Available: <https://www.microsoft.com/es-ww/microsoft-365/visio/flowchart-software>. [Último acceso: Febrero 2021].
- [32] M. Project, «microsoft.com,» Microsoft, 2021. [En línea]. Available: <https://www.microsoft.com/es-ww/microsoft-365/project/project-management-software?market=ec>. [Último acceso: 2021].
- [33] S. Systems, «sparxsystems.com,» 2021. [En línea]. Available: <https://sparxsystems.com/>. [Último acceso: Febrero 2021].
- [34] Quest, «quest.com,» Quest Toad Data Modeler, 2021. [En línea]. Available: <https://www.quest.com/mx-es/products/toad-data-modeler/>. [Último acceso: Febrero 2021].
- [35] Balsamiq, «www.balsamiq.com,» Balsamiq Mockups, 2021. [En línea]. Available: <https://balsamiq.com/wireframes/>. [Último acceso: Marzo 2021].
- [36] StarUML, «staruml.io,» MkLabs, 2021. [En línea]. Available: <https://staruml.io/>. [Último acceso: 2021].

- [37] Bizagi, «bizagi.com,» Bizagi company, 2021. [En línea]. Available: <https://www.bizagi.com/es/plataforma/modeler>. [Último acceso: 2021].
- [38] A. Gregory, J. Atkins y G. Midgley, «Stakeholder identification and engagement in problem structuring interventions,» *ScienceDirect*, vol. 283, nº 1, pp. 321-340, 2020.
- [39] Y. Molina, A. Granda y A. Velázquez, «Los requisitos no funcionales de software. Una estrategia para su desarrollo en el Centro de Informática Médica,» *Scielo*, vol. 13, nº 2, pp. 77-90, 2019.
- [40] L. Guevara y D. Esmeralda, «Historias de usuario generadas a partir de un modelo orientado a objetivos basado en el framework i*,» *Dialnet*, nº 11, pp. 57-72, 2018.
- [41] J. Molina, M. Zea, M. Contento y F. García, «METODOLOGÍAS DE DESARROLLO EN APLICACIONES WEB,» *Dialnet*, vol. 6, nº 3, pp. 54-71, 2017.
- [42] A. Rodas, J. Rios y G. Solarte, «Creación de una arquitectura utilizando Lenguaje de Modelado Unificado (UML) en la implementación de un Lenguaje Específico de Dominio Interno (LEDI): construcción de un LEDI para el modelado de problemas de optimización,» *Scielo*, vol. 10, nº 20, pp. 15-23, 2016.
- [43] L. Baquero, L. Argota, O. Valdés y F. Ciudad, «Método para el modelado y prueba de Diagramas de Casos de Uso mediante redes de Petri,» *Scielo*, vol. 10, nº 2, pp. 138-149, 2016.
- [44] J. Cucunubo, J. Parra y J. Otálora, «Propuesta para la evaluación de Entornos Virtuales de Enseñanza Aprendizaje con base en estándares de usabilidad,» *Dialnet*, vol. 21, nº 41, pp. 135-147, 2017.

ANEXOS

ANEXO A: ESTRUCTURA DEL PORTAFOLIO ACADÉMICO – UTMACH

Ilustración 61: Estructura de portafolio UTMACH

1. Datos informativos:
 - Nombre Institución
 - Nombre de la Carrera
 - Misión
 - Visión de la Carrera
 - Semestre/nivel –datos personales
 - Asignatura
 - Nombre profesor
 - Horario de clases.
 - Periodo Académico.
 2. Elementos curriculares
 - a) Sílabo recuperado del SIUTMACH
 - b) Descripción sobre las expectativas al iniciar el curso
 - c) Apuntes, ejercicios y resúmenes de actividades en clase
 - d) Evaluaciones parciales calificadas
 - e) Investigaciones calificadas
 - f) Informes de actividades de experimentación, prácticas de laboratorio y campo
 - g) Proyectos individuales y grupales
 - h) Estudios de caso
 - i) Planteamiento de problemas y su resolución
 - j) Registros de asistencia recuperados del SIUTMACH y justificación de las inasistencias de darse el caso.
 - k) Registros de observaciones, simulaciones o testimonios que pueden acompañarse de fotografías, videos y audios.
 - l) Productos de talleres en clase
 - m) Trabajos autónomos
 - n) Tareas de refuerzo
 3. Informe final sobre el cumplimiento de las expectativas al finalizar el curso, realizando una crítica constructiva respecto del cambio, determinando las mejoras.
- Las evidencias de proyectos, trabajos en clase teóricos e informes de prácticas y experimentación, así como, los trabajos autónomos y otros, deberán estar revisados por el profesor.

Fuente: UTMACH

ANEXO B: FORMATO DE DISEÑO DE ENTREVISTA

Ilustración 62: Formato de entrevista

Diálogo para determinar la importancia de un sistema para la gestión de portafolios académicos en la Facultad de Ingeniería Civil.

- a. ¿Tiene conocimiento sobre la existencia de un formato estándar establecido por la universidad, para la gestión de portafolios académicos?
- b. ¿Cómo controla la recopilación de evidencias académicas y actividades dentro del portafolio académico?
- c. ¿Considera que debe existir una plataforma que facilite la creación de los portafolios académicos?

Fuente: Elaboración propia

ANEXO C: ANÁLISIS DE RIESGOS

Tabla 22: Gestión de Riesgos

N°	RIESGO	DETONANTE	PREVENCIÓN
RG-1	Recolección incompleta de requisitos.	No dialogar con los interesados del proyecto.	Mantener un continuo diálogo vía conferencia.
RG-2	Falta de conocimiento en la implementación de nuevas tecnologías.	Falta de documentación y guías de desarrollo.	Investigar herramientas en auge y con guías de soporte
RG-3	Falta de recursos tecnológicos para la implementación de la plataforma.	No definir el alcance del proyecto a nivel de arquitectura.	Utilizar alternativas de implementación del sistema.
RG-4	Daño de los equipos físicos de la plataforma.	Desastre natural	Hacer respaldos del código vía GitHub
RG-5	Incompatibilidad de herramientas de desarrollo.	Malas prácticas en el desarrollo del proyecto.	Verificar la guía de compatibilidad y documentación de herramientas.
RG-6	Los procesos de diseño y desarrollo no finalizan en el tiempo estimado.	Disponibilidad de tiempo	Seguir el cronograma de actividades estableciendo holguras.

Fuente: Elaboración propia