



UNIVERSIDAD TECNOLÓGICA ECOTEC

Título del trabajo:

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL
DE DOCUMENTOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD ECOTEC

Línea de investigación:

TECNOLOGIAS DE LA INFORMACIÓN Y COMUNICACIÓN

Modalidad de titulación

PROPUESTA TECNOLÓGICA

Carrera:

INGENIERÍA EN SISTEMAS CON ÉNFASIS EN SISTEMAS

Título a obtener:

INGENIERO EN SISTEMAS

Autor:

JUAN JOSÉ FLORES CUADROS

Tutor:

ING LUIS ESPINOZA MENDOZA

GUAYAQUIL – 2021

Dedicatoria

Dedico esta tesis a mi familia quienes siempre me acompañan en cada victoria y me impulsan a dar el siguiente paso, a mis padres por sus enseñanzas y ética con la que siempre han actuado.

Dedico este trabajo especialmente a mis hijos Juan Andrés y Juan Sebastián que son el motor de mi vida y la razón para seguir creciendo, que nunca olviden que todo sacrificio siempre tiene su recompensa y que todo logro se alcanza con una decisión.

Agradecimiento

Agradezco a Dios por mostrarme el camino, darme la serenidad y valor para aceptar su voluntad. Agradezco a mi familia que con mucha paciencia y amor siempre están listos para darme el impulso cada vez que lo necesito, sobre todo agradezco a mis padres José y María Estelia que con su amor incondicional siempre han estado para escuchar y alentarme. Agradezco a mis hermanos María Belén y Luis Fernando que siempre han puesto esa chispa de alegría en todos los momentos de mi vida.

Agradezco especialmente a la Universidad Ecotec por nutrirme con los conocimientos que hoy poseo y a sus docentes de quienes guardo muy buenos recuerdos y toda su sabiduría.

ANEXO N°15

CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS

Habiendo sido nombrado LUIS ENRIQUE ESPINOZA MENDOZA, tutor del trabajo de titulación "IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE DOCUMENTOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD ECOTEC." elaborado por FLORES CUADROS JUAN JOSÉ, con mi respectiva supervisión como requerimiento parcial para la obtención del título de INGENIERO EN SISTEMAS ENFASIS SISTEMAS

Se informa que el mismo ha resultado tener un porcentaje de coincidencias 2 (%) mismo que se puede verificar en el siguiente link: <https://secure.arkund.com/view/105051556-484878-450078#/details/findings/matches/156>. Adicional se adjunta print de pantalla de dicho resultado.

The screenshot displays the Arkund plagiarism detection interface. At the top, it shows navigation options and a status message: "VOLVER A LA VISTA GENERAL DEL ANÁLISIS" and "Todos los cambios realizados son guardados automáticamente". The document details are: REMITENTE: Lespinoza, ARCHIVO: Tesis_Sistemas_de_manejo_de_documentos.docx, and SIMILITUD: 2%. Below this, there are tabs for "COINCIDENCIAS", "FUENTES", and "DOCUMENTO COMPLETO". The "COINCIDENCIAS" tab is active, showing a "TIPO" filter set to "MOSTRAR EN EL TEXTO". There are three toggle switches: "Citas" (disabled), "Paréntesis" (disabled), and "Diferencias detalladas de texto" (enabled). A document entry is shown with a lightning bolt icon, a trash icon with "99", and a "1 / 7" indicator. The document is labeled "DOCUMENTO ENVIADO" and "INCLUIR EN EL ANÁLISIS" (checked). The similarity percentage is "80%" and the text is "SIMILITUD DE TEXTO". The text content of the match is: "FUE REVISADO, SIENDO SU CONTENIDO ORIGINAL EN SU TOTALIDAD, ASÍ COMO EL CUMPLIMIENTO DE LOS REQUERIMIENTOS QUE SE DICTAN EN LA GUÍA DE ELABORACIÓN DEL TRABAJO DE TITULACIÓN, POR LO QUE SE AUTORIZA A:".

Nombres y Apellidos del Tutor: Ing. Luis Espinoza, M.Ed.

Tesis Lista Recibidos x



LUIS ESPINOZA MENDOZA

para mí ▾

Estimado después de revisar su tesis, cumple los requisitos para su sustentación.

Gracias

--

Ing. Luis Espinoza, M.Ed.

Docente Facultad de Ingenierías

PBX: 04 3723400 Ext.: 508



ANEXO N°16

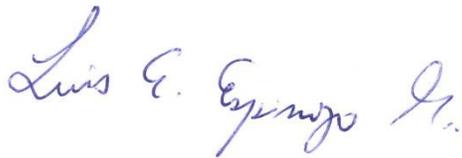
CERTIFICACION DE REVISION FINAL

QUE EL PRESENTE PROYECTO DE INVESTIGACIÓN TITULADO:

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE DOCUMENTOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD ECOTEC.

ACOGIÓ E INCORPORÓ TODAS LAS OBSERVACIONES REALIZADAS POR LOS MIEMBROS DEL TRIBUNAL ASIGNADO Y CUMPLE CON LA CALIDAD EXIGIDA PARA UN TRABAJO DE TITULACIÓN, POR LO QUE SE AUTORIZA A: **FLORES CUADROS JUAN JOSE**, QUE PROCEDA A SU PRESENTACION.

Samborondón, 5-Julio-2021



Nombres y Apellidos del Tutor: Ing. Luis Espinoza, M.Ed.

Tesis Lista Recibidos x



LUIS ESPINOZA MENDOZA

para mí ▾

Estimado después de revisar su tesis, cumple los requisitos para su sustentación.

Gracias

--

Ing. Luis Espinoza, M.Ed.

Docente Facultad de Ingenierías

PBX: 04 3723400 Ext.: 508



Resumen

El sistema de control documental digital es parte primordial en cualquier Universidad o empresa que necesite almacenar y consultar documentación. El presente proyecto pretende satisfacer esta necesidad usando buenas prácticas y metodologías que permitan servir como inicio para la automatización de este proceso usando tecnología actual. Para esto se identificaron funcionalidades en común entre los procesos investigados.

El proyecto inicia consultando con los docentes encargados de cada área y los alumnos quienes comentan la problemática que tienen y las expectativas de una herramienta que permita la automatización de este proceso.

Se incluye la generación de reporte que permite visualizar rápidamente el cumplimiento de las solicitudes de documentos para ayudar con el control en forma ágil. Es importante destacar que el software permite la estandarización de los documentos solicitados mediante restricciones de formatos y uso de plantillas.

La metodología de desarrollo utilizada es scrum apoyado por herramientas como trello y la integración continua de gitlab. Esto dió como resultado un software que mejora el control en el proceso de almacenamiento de documentos.

Abstract

The electronic document management system is an essential part of any University or company that needs to store and consult documentation. The present project aims to satisfy this need using good practices and methodologies that allow to serve as a starting point for the automation of this process using current technology. For this, common functionalities were identified among the investigated processes.

The project begins by consulting with the teachers in charge of each area and the students who comment on the problems they have and the expectations of a tool that allows the automation of this process.

Report generation is included that allows you to quickly visualize the fulfillment of document requests to help with control in an agile way. It is important to note that the software allows the standardization of the requested documents through format restrictions and the use of templates.

The development methodology used is scrum supported by tools such as trello and gitlab continuous integration. This resulted in software that improves control over the document storage process.

Índice

Dedicatoria.....	i
Agradecimiento	ii
CERTIFICACION DE REVISION FINAL.....	Error! Bookmark not defined.
Resumen.....	v
Abstract.....	vii
Índice	viii
Índice de ilustraciones.....	xi
Índice de tablas	xii
Índice de anexos	xiii
Introducción.....	xiv
Antecedente.....	xiv
Planteamiento del problema a resolver.....	xv
Objetivo general.....	xvi
Objetivos específicos	xvi
Marco teórico	1
Sistemas de manejo de documentos.....	1
Sistema de gestión de documentos electrónicos.....	2
Características comunes de EDMS	2
Lenguajes de programación	4
HTML.....	4
PHP	4
JavaScript.....	4
Base de datos.....	5
Mariadb.....	5
ORM	5
Eloquent ORM	5
Frameworks	5
Framework Laravel.....	5
Symfony.....	6
Framework Vue	6

Entornos de desarrollo integrado	6
PHPstorm	6
Visual Studio Code	7
Servidor web.....	7
Apache	7
Protocolos de comunicación	9
OpenSSL.....	9
Manejador de versiones.....	9
Git.....	9
GitLab.....	9
Depuración de programas	9
Xdebug	10
ESLint.....	10
Prettier	10
Prácticas de ingeniería de software	10
Integración continua	10
Entrega continua	11
Pruebas de desarrollo.....	11
Unit testing.....	11
Pruebas unitarias en PHPUnit.....	11
Test Driven Development	12
Librerías y bibliotecas	12
Composer	12
Node.....	13
NPM.....	13
Tailwind CSS	13
Axios.....	13
PHPDocumentor.....	14
Metodología de software.....	14
Metodología ágil	14
Framework Scrum	15
Metodología XP	15

Kanban	15
Herramientas a utilizar	15
Metodología del proceso de desarrollo de la propuesta tecnológica.....	19
Metodología de desarrollo	19
Enfoque de la investigación	19
Tipo de Investigación	19
Período y lugar donde se desarrolla la propuesta tecnológica	19
Universo y muestra	20
Definición y comportamiento de las principales variables incluidas en el estudio	20
Definición de variables.....	20
Variable independiente	20
Variable dependiente.....	20
Operacionalización:	20
Métodos empleados e instrumentos de la investigación.	21
Procesamiento y análisis de la información.....	21
Análisis e interpretación de los resultados	26
Propuesta.....	35
Diccionario de datos	35
Modelo entidad-relación.....	50
Sistema operativo	51
Framework Laravel	51
Framework Vue JS	55
Pruebas unitarias.....	57
Despliegue.....	58
Conclusiones.....	61
Recomendaciones	63
Bibliografía	64
Anexos	67

Índice de ilustraciones

Ilustración 1 Backlog	26
Ilustración 2 Ingreso del software.....	27
Ilustración 3 Pantalla del listado de usuarios	28
Ilustración 4 Mantenedor para crear / editar un usuario.....	28
Ilustración 5 Pantalla principal del módulo de procesos.....	29
Ilustración 6 Mantenedor de procesos	30
Ilustración 7 Mantenedor de requerimientos	30
Ilustración 8 Reglas/ restricciones para cada archivo	31
Ilustración 9 Reportes de cumplimiento	31
Ilustración 10 Mantenedor de formatos con versiones.....	32
Ilustración 11 Pantalla para subir documentos del usuario alumno	33
Ilustración 12 Validando formulario de ingreso	33
Ilustración 13 Validando campos requeridos	34
Ilustración 14 Modelo entidad -relación.....	50
Ilustración 15 Estructura de carpetas del proyecto	54
Ilustración 16 Modelo - vista - controlador	55
Ilustración 17 Estructura de un archivo en vuejs.....	57
Ilustración 18 Pruebas unitarias	58

Índice de tablas

Tabla 1: Variables	21
Tabla 2: Tabla abilities	35
Tabla 3: Tabla assigned_roles	36
Tabla 4: Tabla careers	36
Tabla 5: Tabla categories.....	37
Tabla 6: Tabla document_request_rules.....	37
Tabla 7: Tabla document_requests.....	38
Tabla 8: Tabla document_template_versions	39
Tabla 9: Tabla document_templates	40
Tabla 10: Tabla documents.....	41
Tabla 11: Tabla failed_jobs	42
Tabla 12: Tabla menus	42
Tabla 13: Tabla migrations.....	43
Tabla 14: Tabla password_resets	43
Tabla 15: Tabla permissions	43
Tabla 16: Tabla personal_access_tokens	44
Tabla 17: Tabla process_heads	44
Tabla 18: Tabla process_users	45
Tabla 19: Tabla processes	45
Tabla 20: Tabla roles	46
Tabla 21: Tabla schools	46
Tabla 22: Tabla sessions	47
Tabla 23: Tabla user_details	48
Tabla 24: Tabla users	49
Tabla 25: Tabla de dependencias laravel	53
Tabla 26: Tabla de dependencias vuejs.....	56

Índice de anexos

Anexo 1 67

Introducción

Antecedente

La facultad de Ingenierías de la Universidad Ecotec como parte de sus procesos documenta todas las actividades que realizan los estudiantes respaldándolas en sistemas de repositorio de documentos como Google Drive.

Al ser esta una tarea cíclica y sensible para automatización, se pretende implementar un sistema web que permita subir la documentación de forma centralizada, al mismo tiempo se pretende agilizar la revisión del cumplimiento de fechas de entregas manteniendo un flujo de comunicación eficiente y relevante sobre la gestión entre alumnos y docentes, esto se lo realizará creando un repositorio con los formatos al alcance del alumno.

Dentro de las políticas de calidad de la Universidad Ecotec una actividad prioritaria es la de aplicar mejoras continuas por lo que contar con el sistema propuesto en el presente trabajo evitará que el personal realice tareas repetitivas, optimizando su tiempo, al mismo tiempo que permitirá a los alumnos estar informados de los formatos y fechas de cada gestión.

La digitalización de los documentos permite evidenciar, controlar y proporcionar acceso a los mismos de manera inmediata, facilitando su administración, el cumplimiento de los tiempos asociados a los procesos y evitando la duplicidad de información o la demora en la búsqueda de un documento.

Un pilar base de la filosofía de la Universidad Ecotec es la responsabilidad ambiental por lo que el disminuir al máximo posible el uso de papel es un objetivo a alcanzar al mismo tiempo que se debe mantener la evidencia de todos los procesos para su posterior consulta y análisis. Es así que los estudiantes podrán descargar el formato para documentar la actividad a realizar o el trámite a solicitar y en caso de actualización del formato evitar el desperdicio de los formatos almacenados en bodega.

En la actualidad los documentos se almacenan en carpetas con apellidos y nombre de los estudiantes siendo esta forma de trabajo poco eficiente como proceso y con una alta probabilidad de error y afectación de la integridad de la información, debido a que se pueden eliminar o sobrescribir de manera involuntaria.

La justificación de este proyecto es agregar facilidad para almacenar y obtener la información para el nuevo personal y la auditoría del proceso y así de esta forma agregar calidad al proceso pudiendo incluso acceder a certificaciones internacionales. (KYOCERA Document Solutions, 2021).

Surge la pregunta: ¿Cómo mejorar los controles de cumplimiento a los documentos de los procesos en la facultad de Ingeniería de la Universidad Ecotec?

El sistema de control documental propuesto permite mantener la individualidad y privacidad de los documentos al mismo tiempo que facilita el control del cumplimiento mediante reportes que permiten realizar su consulta bajo demanda y en tiempo real.

Planteamiento del problema a resolver

Actualmente la Facultad de Ingeniería de la Universidad Ecotec tiene inconvenientes con la gestión del control de documentos, ellos están usando para este fin el servicio de Google Drive, el cual es básico y no brinda las funcionalidades precisas para las necesidades requeridas.

La Universidad Ecotec en sus inicios contaba con pocos alumnos por lo que no tenía la necesidad de un sistema de control de documentos especializado debido a su crecimiento esta actividad está generando problemas, ya que Google Drive no proporciona una forma de control de cumplimiento de los archivos solicitados, no posee control de archivos duplicados y al tener que navegar entre subcarpetas el proceso es lento, tedioso y propenso al error. Siendo este un proceso de vital importancia es mandatorio poder identificar el documento en un formato preestablecido que cuente con la aprobación de los responsables que pueda ser posteriormente almacenado y usado, manteniendo un control sobre los cambios

realizados. “La gestión documental consiste en controlar de un modo eficiente y sistemático la creación, la recepción, el mantenimiento, la utilización y la disposición de los documentos”. (ISO 15489-1: 2001).

En la actualidad la carga de los documentos es realizada por cada estudiante que debe subirlos en carpetas que llevan su apellido y nombre. Posteriormente cada alumno va almacenando los documentos bajo petición del docente encargado del proceso. Estos procesos tienen una fecha límite, momento en el cual el docente bloquea el link de acceso. El docente encargado revisa carpeta por carpeta varias ocasiones para ver el cumplimiento y dar seguimiento.

Pese a esto Google Drive al ser un sistema en donde se pueden compartir archivos y carpetas entre los alumnos, profesores y personal administrativo puede generarse el error eliminar documentos o moverlos de carpeta sin mal intención, lo cual causaría un malestar entre los participantes del proceso.

Por tal motivo es importante la implementación de un software de sistema de gestión documental creado a la medida que permita usar formatos preestablecidos y se puedan compartir con los usuarios predeterminados para que los alumnos puedan almacenar la documentación requerida por el docente encargado del proceso y que al mismo tiempo sea de fácil acceso, revisión, seguimiento y control para el docente y las personas involucradas en el proceso.

Objetivo general

Implementar un sistema de control documental sobre los procesos académicos de la Facultad de Ingeniería que permita el almacenamiento digital de los documentos en forma ágil y controlada.

Objetivos específicos

- Identificar los tres procesos principales de control documental de la Facultad de Ingeniería que sirvan de guía para determinar la funcionalidad base de la herramienta.

- Desarrollar una solución basada en web que permita almacenar documentos.
- Definir la aceptación del sistema y satisfacción de parte del usuario.

Marco Teórico

Capítulo 1

Marco teórico

Sistemas de manejo de documentos

Las empresas están cada vez más interesadas en cuidar su información privada como para aseverarse de que se maneje de manera correcta y sea más eficaz para sus labores cotidianas. A medida que tanto, el número y la diversidad de documentos hacen que la gestión y la utilización sean más desafiantes. Por consiguiente, cómo conseguir una administración eficiente de los documentos se convierte en un asunto emergente en la sociedad rica en información de hoy. El requisito puede clasificarse además en 2 puntos, que se describen luego.

Compatibilidad con el acceso ubicuo: para poder hacer una administración eficiente de los documentos, se debería considerar la función de los usuarios para entrar de forma apropiada a la información más actual en cualquier instante, en cualquier sitio y a partir de cualquier dispositivo. Además, debido al incremento y la pluralidad de dispositivos, los mecanismos de gestión e ingreso a los documentos que corresponden tienen que proporcionarse sin cargas de aprendizaje extras y operaciones complicadas de parte de los usuarios.

Apoyo para el intercambio y la participación: gracias a la globalización y la gran separación de trabajos, el intercambio y la participación en las empresas son inevitables e incrementan. Los compañeros de trabajo tienen la posibilidad de participar en una labor y compartir un grupo especial de documentos para respaldar sus trabajos. A medida que tanto, los documentos evolucionarán mientras avanza la labor. Se debería tener en cuenta el control y la administración de documentos. (Kao & Liu , 2013)

Las labores de administración de documentos ocurren en una variedad de espacios y en una diversidad de tamaños de almacenes de información y estructuras de información. Ejemplificando, las labores más comunes es la gestión sistemas de archivos, mandar mensajes de correspondencia y uso de páginas web. Las construcciones de información primordiales que puede descubrir un cliente integran listas ordenadas, jerarquías y gráficos. Comúnmente, los documentos tienen la posibilidad de pertenecer a bastante más de una de estas construcciones de información. (DOURISH, y otros, 2000)

Sistema de gestión de documentos electrónicos

El sistema de administración de documentos electrónicos (EDMS) posibilita la digitalización de documentos usando sistemas y tecnologías informáticos para saciar las necesidades corporativas. Así, muchas instituciones que usan un EDMS completo gestionan de forma sencilla toda la información producida interna y externamente.

Por consiguiente, en términos de eficiencia y productividad en la prestación de servicios, este sistema todavía funciona a un grado preeminente comparativamente con las instituciones que atienden con procedimientos clásicos. La utilización exitosa de EDMS es una necesidad para cada una de las instituciones, debido a que dichos sistemas aceleran los procesos comerciales y brindan tranquilidad a los usuarios. (Ayaz & Yanartaş, 2020)

Para guardar y acomodar diversos tipos de documentos, un sistema de programa se estima un sistema de administración de documentos electrónicos (EDMS). Esta clase de marco es un tipo de sistema de administración de documentos cada vez más específico, un tipo de sistema de almacenamiento cada vez más extenso que anima a los consumidores a ordenar y guardar informes en papel o computarizados.

EDMS alude tanto más explícitamente a un marco de producto que maneja documentos digitales, en oposición a registros en papel, pese a que, en determinados ejemplos, dichos marcos además tienen la posibilidad de intentar con versiones examinadas computarizadas de documentos originales en papel.

La administración de documentos electrónicos es muchísimo más que solo analizar y ahorrar: es un marco completo que posibilita a los trabajadores de la información acomodar y dispersar eficazmente los registros de la agrupación para un uso mejor y coordinado en las labores del día a día.

Características comunes de EDMS

Para la unión del sistema y la automatización integral de documentos, los sistemas de administración de documentos electrónicos poseen varias herramientas a partir del almacenamiento y la digitalización elemental del papel.

a) Categorización e indexación de archivos jerárquicos: Los documentos digitales se catalogan en un almacén focal con base en construcciones de sobres que mantienen los informes, de modo que, conformados por oficina, fecha de construcción u otras fronteras de gestión de datos.

b) La unión según la programación de administración de documentos de todo el sistema extrae datos de marcos de registro como CRM, ERP y bases de datos de recursos humanos para capturar, rastrear o conectar de manera natural a los datos hechos a lo extenso de la compañía.

c) Recuperación y averiguación vinculada EDMS posibilita a los representantes aprobados descubrir registros de manera cruzada en sobres de documento y almacenes, con habilidades de averiguación que principalmente integran el nombre del archivo, los metadatos, el contenido del informe y más.

d) Los seguimientos de auditoría y registros de estabilidad llegan a diseñar por trabajo representativo o separación colocando los datos en las manos necesarias. Por lo cual, las pistas de auditoría rastrean con exactitud cómo se accede a los datos para continuar en la consistencia administrativa. (Pustokhina, Rokhayati Rosa, Lydia, Shankar, & Huda, 2019)

Según (Sathiadas & Wikramanayake, 2003) las aplicaciones de EDMS se centran en el control de documentos electrónicos durante todo su periodo de vida, a partir de la construcción hasta el ser archivado. Sus funcionalidades integran la construcción, almacenamiento y recuperación de documentos, gestión, control de variantes, flujo de trabajo y diversos formatos de entrega.

Construcción de documentos: un archivo es un contenedor que centra información de una pluralidad de fuentes, en diversos formatos, alrededor de un asunto específico para saciar las necesidades de una persona u organización en especial.

Almacenamiento y recuperación: esto involucra guardar y recobrar documentos en un dispositivo de almacenamiento como, ejemplificando, un disco duro, una cinta, etcétera.

Administración: cubre una extensa área de administración de todos los documentos de forma eficiente para saciar las necesidades de la organización y los individuos.

Control de variantes: esta es una manera de hacer un seguimiento de los cambios hechos en un archivo y la función de recobrar variantes viejas de un archivo.

Flujo de trabajo: esta es una manera de rastrear el estado del archivo y quién es responsable de aquel paso.

Diversos formatos de entrega: maneras de dar el contenido del archivo en diversos formatos, como PDF, Word, Imagen, etcétera., para saciar las necesidades de los usuarios finales.

Lenguajes de programación

En esta sección se detalla los lenguajes necesarios para la construcción del software a implementarse. Esta elección está ligada a los frameworks que se han elegido y a la facilidad de aprendizaje para que cualquier nuevo integrante en el equipo pueda rápidamente integrarse a la producción de código.

HTML

Es el lenguaje en el cual permanecen escritos la mayor parte de los sitios web. HTML se usa para producir páginas y hacerlas funcionales. El código usado para hacerlos visualmente atractivos se sabe cómo CSS. (Html, 2021)

PHP

Es un lenguaje de programación del lado del servidor diseñado especialmente para la web, en una página HTML, puede incrustar código PHP que se ejecutará cada vez que se visite la página. El código PHP se interpreta en el servidor web y crea HTML u otra salida que verá el visitante. (Thomson & Welling, 2016)

JavaScript

Es una manera de añadir programas a páginas web en el navegador Netscape Navigator. A partir de entonces, el lenguaje fue adoptado por todos los otros navegadores web gráficos relevantes. Ha producido probables las aplicaciones web modernas, aplicaciones con las que puede interactuar de manera directa sin tener que recargar la página para cada acción. JavaScript además se usa en sitios web más clásicos para dar distintas maneras de interactividad e inteligencia. (Haverbeke, 2018)

Base de datos

Para poder crear y consultar la información se usa un conjunto de programas específicos que permiten consultar y generar informes, en esta sección se detalla los programas utilizados.

Mariadb

Es el sistema de gestión de bases de datos Mariadb de código abierto más distinguido. Las bases de datos Mariadb constan de varias tablas. Las tablas tienen dentro los datos. Las tablas se conforman de columnas y filas. Los usuarios a los que se les han otorgado funciones tienen la posibilidad de producir y remover tablas de dicha base de datos. (Dyer, 2015)

ORM

El mapeo relacional de objetos es una técnica de programación para convertir datos entre sistemas de tipos incompatibles en lenguajes de programación orientados a objetos. Esto crea, en efecto, una "base de datos de objetos virtuales" que se puede utilizar desde dentro del lenguaje de programación. (Roebuck, 2011)

Eloquent ORM

Incluido con Laravel, suministra una ejecución de ActiveRecord simple para trabajar con su base de datos. Cada tabla de la base de datos tiene un "Modelo" el cual se utiliza para interactuar con esa tabla. Los modelos le permiten examinar datos en sus tablas, así como insertar nuevos registros en la tabla. (Laravel, 2021)

Frameworks

Para tener un sistema organizado se utiliza frameworks o marcos de trabajos que estandarizan conceptos, prácticas y criterios sobre problemas frecuentes.

Framework Laravel

Este framework se desarrolló para diseñar sitios web o aplicaciones web. Los frameworks son sinónimos de una jaula o una red donde se procesa recursos o paquetes incluidos

en ella y brindan una característica similar a la de un pegamento. Esta característica parecida al pegamento preserva todos los paquetes unidos.

El framework Laravel está escrito en lenguaje de programación PHP. El preprocesador de hipertexto PHP es conocido por incrustar marcos como los mencionados previamente. PHP o la página de inicio personal es un lenguaje de programación con script disponible que se ha usado para incrustar servidores HTML. (Mangukiya, 2020)

Symfony

Es un entorno de desarrollo estable y reconocido internacionalmente. Un entorno de trabajo innovador y fácil de usar gracias a la integración de soluciones creadas en otros entornos como la inyección de dependencias (extraídas de Java) y soluciones desarrolladas específicamente como Web Debug Toolbar o Web Profiler. Finalmente, al adoptar estándares de facto, Symfony no lo limita a su entorno, sino que permite elegir los componentes de software que desea utilizar. (Symfony, 2021)

Framework Vue

Vue es un framework progresivo para crear interfaces de usuario. A diferencia de otros marcos monolíticos, Vue está diseñado desde cero para ser adoptable de forma incremental. La biblioteca principal se centra únicamente en la capa de visualización y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes. (Pšenák & Tibensky, 2020)

Entornos de desarrollo integrado

Para poder escribir el código se necesita un editor de código, en este caso los entornos de desarrollo integrado van más allá de ser solo editores de código. Estos además de permitir escribir el código, brindan ayuda con ciertas funcionalidades adicionales para el desarrollador como es la indentación, formateo y ayudas visuales para que el trabajo del desarrollador sea más productivo.

PHPstorm

Es un editor de código inteligente que proporciona la funcionalidad para completar inteligentemente los códigos, resaltar la sintaxis, configurar los formatos de código

extendido, verificar errores sobre la marcha, plegar códigos, verificar la compatibilidad con mezclas de idiomas. (Chaudhary & Kumar, 2014)

Visual Studio Code

No es un simple editor de código; es un entorno poderoso que pone la escritura de código como eje central. El propósito principal de Visual Studio Code no es crear binarios (como archivos .exe y .dll); Este editor facilita la escritura de código para plataformas web, móviles y en la nube para cualquier desarrollador que trabaje en diferentes sistemas operativos, como Windows, Linux y OS X, lo que lo mantiene independiente de los entornos de desarrollo propietarios. Sus funciones principales son:

Dar soporte integrado para codificar con muchos idiomas, incluidos los que normalmente se usa en escenarios de desarrollo multiplataforma, con funciones de edición avanzadas y soporte para idiomas adicionales a través de la extensibilidad.

Uso de depurador integrado para Node.js, con soporte para depuradores adicionales (como Mono) mediante extensibilidad.

Realizar control de versiones basado en el popular motor Git, que proporciona una experiencia integrada de colaboración que admite confirmaciones de código y bifurcaciones, y es la elección adecuada para una herramienta destinada a trabajar posiblemente con cualquier idioma. (Del Sole, 2017)

Servidor web

Un servidor web es un servidor que se encarga de aceptar solicitudes HTTP de clientes web y brindarles respuestas HTTP, generalmente en forma de páginas web que contienen contenido estático (texto, imágenes, etc.) y dinámico (scripts).

Apache

El servidor web Apache ha sido el servidor web más popular y ampliamente utilizado durante la última década. Es utilizado por aproximadamente el 50% de todos los sitios web. Apache es multiplataforma, ligero, robusto y se utiliza tanto en pequeñas empresas como en grandes corporaciones. Apache también es gratuito y de código abierto.

Una de las mejores características que ofrece Apache es que se ejecuta en prácticamente todas las plataformas informáticas más utilizadas. Apache no solo se ejecuta en la mayoría las versiones de Unix y también se ejecuta en versiones de Windows y muchos otros sistemas operativos de escritorio.

Apache ofrece muchas otras características, incluida la indexación de directorios, alias de directorio, negociaciones de contenido, informes de errores HTTP configurables, ejecución de SetUID de Programas CGI, gestión de recursos para procesos secundarios, mapas de imágenes del lado del servidor, reescritura de URL, revisión ortográfica de URL y manuales en línea.

Las otras características principales de Apache son:

Es totalmente compatible con el estándar HTTP 2. Un navegador web puede enviar solicitudes en paralelo, lo que ahorra ancho de banda al no transmitir encabezados HTTP en cada solicitud. Es probable que esto proporcione un aumento del rendimiento en el lado del usuario final porque los archivos solicitados en paralelo aparecerán más rápido en el navegador.

Es lo suficientemente flexible como para permitirle distribuir la configuración de su host virtual en varios archivos para que un solo httpd.conf no se vuelva demasiado engorroso de administrar con muchas configuraciones de servidor virtual.

Soporte para hosts virtuales: Apache es también uno de los primeros servidores web en admitir hosts virtuales basados en IP y con nombre.

Soporte para scripting PHP: este lenguaje de scripting es muy utilizado y Apache proporciona un gran soporte para PHP usando el módulo mod_php.

Estado del servidor y registros personalizables: Apache le brinda flexibilidad para registrar y monitorear el estado del servidor en sí. El estado del servidor se puede monitorear a través de un navegador web. También puede personalizar sus archivos de registro a su gusto.

Soporte para Secure Socket Layer (SSL): puede crear fácilmente un sitio web SSL utilizando OpenSSL y el módulo mod_ssl para Apache. (Apache, 2021)

Protocolos de comunicación

OpenSSL

Es una implementación de código abierto de los protocolos ubicuos Secure Sockets Layer (SSL) y Transport Layer Security (TLS) esenciales para proporcionar una comunicación segura a través de Internet. OpenSSL está disponible para una gran cantidad de sistemas operativos y está ampliamente distribuido e implementado en muchas plataformas. También se ha utilizado como un componente esencial en una serie de proyectos relacionados con la infraestructura de comunicación segura, por ejemplo, OpenVPN y mod_ssl. (Lawall, Laurie, Hansen, Palix, & Muller, 2010)

Manejador de versiones

Durante el proceso de desarrollo de software se genera un conjunto de código según el requerimiento y una vez concluido el requerimiento este código debe integrarse a un repositorio para ser compartido con los demás miembros del equipo, así al mismo tiempo se mantenga un respaldo del trabajo realizado.

Git

Es un sistema de control de revisiones distribuido, rápido y escalable con un conjunto de comandos que proporciona operaciones de alto nivel y acceso completo a los componentes internos. (Git scm, 2021)

GitLab

Es una plataforma DevOps abierta, que permite crear un flujo de trabajo de software optimizado que brinda libertad a la organización para realizar trabajos en forma eficiente a lo largo del ciclo de vida de DevOps. (Gitlab, 2021)

Depuración de programas

Los errores en el desarrollo son frecuentes y el proceso de corregirlos es la depuración para esto existen herramientas como las detalladas a continuación que permite encontrarlos más rápidamente.

Xdebug

La extensión PHP de Xdebug le permite ampliar la cantidad de información que registra el servidor web. Symfony integra los mensajes de Xdebug en sus propios comentarios de depuración, La instalación de la extensión depende mucho de su plataforma. (Symfony, 2021)

ESLint

Es una herramienta para identificar e informar sobre patrones encontrados en código ECMAScript / JavaScript, con el objetivo de hacer que el código sea más consistente y evitar errores. En muchos sentidos, es similar a JSLint y JSHint con algunas excepciones como que usa Espree para el análisis de JavaScript, utiliza un Abstract Syntax Tree (AST) para evaluar patrones en el código y es completamente conectable, cada regla es un complemento y puede agregar más en tiempo de ejecución. (Eslint, 2021)

Prettier

Es un formateador de código persistente. Elimina todo el estilo original y garantiza que todo el código generado se ajuste a un estilo coherente. Prettier toma su código y lo reimprime desde cero teniendo en cuenta la longitud de la línea. (Prettier, 2021)

Prácticas de ingeniería de software

Es importante involucrar al usuario durante el proceso de desarrollo para que pueda dar su punto de vista en el código entregado, para esto es importante que el requerimiento que está terminado sea incluido en el servidor de prueba y sea testeado. Para esto se realizan integraciones continuas y entregas continuas.

Integración continua

La Integración Continua (CI) es una práctica de desarrollo ampliamente establecida en la industria del desarrollo de software, en la que los miembros de un equipo integran y fusionan el trabajo de desarrollo (por ejemplo, código) con frecuencia, por ejemplo, varias veces al día. CI permite a las empresas de software tener ciclos de lanzamiento más cortos y frecuentes, mejorar la calidad del software y aumentar la productividad de sus

equipos. Esta práctica incluye la creación y prueba de software automatizado. (Mojtaba & Muhammad, 2017)

Entrega continua

La entrega continua (CDE) tiene como objetivo garantizar que una aplicación esté siempre lista para producción después de pasar con éxito las pruebas automatizadas y los controles de calidad. CDE emplea un conjunto de prácticas, por ejemplo, CI y automatización de implementación para entregar software automáticamente a un entorno similar a la producción. Según, esta práctica ofrece varios beneficios, como un menor riesgo de implementación, menores costos y obtener comentarios de los usuarios más rápidamente. (Mojtaba & Muhammad, 2017)

Pruebas de desarrollo

Para garantizar la calidad de software de cualquier desarrollo, el software debe ser probado en distintos niveles.

Unit testing

Las Unit test tienen los siguientes tres atributos importantes. Una prueba unitaria verifica un segmento pequeño de código (también conocido como unidad), lo hace rápido y lo hace de forma aislada.

Hay un consenso con los dos primeros atributos. Sin embargo, puede haber controversia sobre qué constituye exactamente una prueba unitaria rápida porque es una medida muy subjetiva. Si el tiempo de ejecución del total de pruebas es suficiente, significa que las pruebas son lo suficientemente rápidas. (KHORIKOV, 2020)

Pruebas unitarias en PHPUnit

Las pruebas unitarias es el proceso de escribir pruebas que ejercen la funcionalidad básica cohesiva más pequeña de unidad en php, una clase. También puede escribir pruebas para procedimientos y funciones, pero las pruebas de Unit funcionan mejor con clases cohesivas y desacopladas. Por tanto, la programación orientada a objetos es un requisito; este proceso se contrapone a las pruebas funcionales y de integración, que construyen gráficos medianos y grandes de objetos cuando se ejecutan. Las pruebas

unitarias prueba instancias con una o muy pocas unidades a la vez, y esto implica que las pruebas unitarias suelen ser fáciles de ejecutar en todos los entornos y no sobrecargan el sistema con operaciones pesadas. (Sironi, 2009)

Test Driven Development

TDD es una técnica simple y eficaz para desarrollar software de calidad que funcione y se pueda modificar refactorizándolo frecuentemente. Esto permite que el código tenga tasas de defectos más bajas y se pueda probar continuamente permitiendo que se eliminen errores y haciendo más productivo el desarrollo. (KOSKELA, 2008)

La implementación más común del ejemplo del semáforo, se abrevia a rojo, verde y refactorización. La luz roja representa el código de prueba defectuoso o posiblemente no compilable. La luz verde es el resultado final de escribir la cantidad mínima de código para que ese código de prueba pase. La refactorización se utiliza para eliminar cualquier duplicación de código o malas técnicas de programación que se introdujeron al volverse verde lo más rápidamente posible. (Hammond & Umphress, 2012)

Librerías y bibliotecas

Los gestores de paquete brindan acceso a código de problemas comunes que ya fueron resueltos por otros desarrolladores y han sido probados por muchos así. De esta forma se logra mayor agilidad en la construcción de nuevos desarrollos. A continuación se detalla los manejadores de estas librerías para cada lenguaje usado en el desarrollo de este software.

Composer

Es una herramienta para la gestión de dependencias en PHP. Te permite declarar las bibliotecas de las que depende un proyecto y las administrará. Principalmente se usa para declarar las bibliotecas de las que depende, descubrir qué versiones de qué paquetes pueden instalarse y actualizar todas sus dependencias con un solo comando. (Composer, 2021)

Node

Es una plataforma para desarrollar aplicaciones de red. Está construido sobre V8, el motor de tiempo de ejecución de JavaScript de Google. Sin embargo, Node no es solo un V8. Una parte importante de la plataforma Node es su biblioteca principal. Esto abarca todo, desde servidores TCP hasta administración de archivos asíncrona y síncrona. (YOUNG & HARTER, 2015)

NPM

Es un manejador que sirve para compartir códigos funcionales de otros desarrolladores. Consta de tres componentes distintos el sitio web, la interfaz de línea de comandos (CLI) y el registro. El sitio web se usa para descubrir paquetes, configurar perfiles y administrar otros aspectos de su experiencia npm. La CLI se ejecuta desde una terminal y es la forma en que la mayoría de los desarrolladores interactúan con npm. Por último, el registro es una gran base de datos pública de software JavaScript y la meta información que la rodea. (Npmjs, 2021)

Tailwind CSS

Es un "marco CSS de utilidad para crear rápidamente diseños personalizados", puede hacer que el CSS de su sitio sea más fácil de controlar y depurar. Casi todas las clases básicas de Tailwind son envoltorios finos alrededor de una única configuración de estilo CSS, como usar "m-4" para proporcionar un "margin: 1rem" o "text-lg" para cambiar el tamaño del texto a "font-size:". 125rem". (Rappin, 2021)

Axios

Es un cliente HTTP basado en promesas para node.js y el navegador. Es isomórfico (Puede ejecutarse en el navegador y en nodejs con la misma base de código). En el lado del servidor, utiliza el módulo http nativo node.js, mientras que en el cliente (navegador) utiliza XMLHttpRequests. Sus funciones principales son Hacer XMLHttpRequests desde el navegador, realizar solicitudes http desde node.js, admite la API Promise, intercepta solicita y respuesta, transformar los datos de solicitud y respuesta, cancelar solicitudes, transformaciones automáticas para datos JSON y dar soporte del lado del cliente para protegerse contra XSRF. (Axios-http, 2001)

PHPDocumentor

Es una herramienta con la que es posible generar documentación a partir de su código fuente PHP. Con esta documentación, puede proporcionar a sus consumidores más información sobre la funcionalidad incorporada en su fuente y no solo sobre lo que pueden utilizar desde su interfaz de usuario. La documentación generada por phpDocumentor no pretende ser un reemplazo de la documentación convencional, sino que es más bien documentación complementaria o de referencia. (Symfony, 2021).

Metodología de software

Metodología ágil

Los procedimientos ágiles son procesos que secundan la filosofía expedita. Sus frameworks integran Extreme Programming y Scrum. Los procedimientos ágiles consisten en recursos personales denominados prácticas. Las prácticas integran la utilización del control de variantes, el establecimiento de estándares de codificación y la entrega de demostraciones semanales a las piezas interesadas. La mayor parte de estas prácticas hay ya hace años. Los procedimientos ágiles los combinan de maneras únicas, acentuando esas piezas que secundan la filosofía diligente, descartando lo demás y mezclando varias ideas novedosas. (Shore & Warden, 2008)

Las metodologías ágiles adoptan un enfoque iterativo para el desarrollo de programa. A diferencia de un modelo de cascada lineal sencillo, los proyectos ágiles consisten en una secuencia de ciclos más pequeños: sprints. Todos ellos es un plan en miniatura: tiene un backlog y consta de fases de diseño, utilización, testing y despliegue en los alcances de trabajo predefinidos.

Finalmente de cada Sprint se entrega un aumento de producto potencialmente enviable. Por consiguiente, con cada iteración se añaden novedosas propiedades al producto, lo cual da como resultado el aumento gradual del software. Con las propiedades validadas inicialmente del desarrollo, las modalidades de dar un producto potencialmente fallido son de manera significativa menor.

Primordiales puntos ágiles:

Flexibilidad: el alcance del trabajo puede modificar según los nuevos requisitos.

Desglose del trabajo: el plan consta de pequeños ciclos (conocidos como Sprints en Scrum).

Costo del trabajo en grupo: los miembros del equipo trabajan en estrecha participación y poseen una perspectiva clara de sus responsabilidades.

Mejoras iterativas: existe una reevaluación recurrente del trabajo llevado a cabo en un periodo para mejorar el producto final.

Cooperación con un comprador: un comprador participa de cerca en el desarrollo y puede modificar los requisitos o admitir las recomendaciones del equipo. (Altexsoft, 2016)

Framework Scrum

Es un Framework de desarrollo de programa diligente incremental e iterativo para gestionar el desarrollo de productos. La herramienta se puede usar para producir productos complicados. Es "un plan de desarrollo de producto flexible. No prescribe ni una de las prácticas usuales de ingeniería, personas, administración de peligros u otras prácticas. Ya que Scrum no explica explícitamente ni una práctica de ingeniería, es viable y, algunas veces, deseable además tener en cuenta no Prácticas de Scrum que tienen la posibilidad de estar estrechamente asociadas al éxito de Scrum. (Sachdeva, 2016)

Metodología XP

Metodología creada por Kent Beck, Ward Cunningham y Ron Jeffries que propone un conjunto de prácticas técnicas. (Bahit, 2012)

Kanban

Kanban es un método de gestión de flujo de trabajo para definir, gestionar y mejorar los servicios que ofrecen trabajo de conocimiento. Su objetivo es ayudarlo a visualizar su trabajo, maximizar la eficiencia y mejorar continuamente. (Kanbanize, 2021)

Herramientas a utilizar

En el presente desarrollo se utilizó el lenguaje de marcado HTML y JavaScript dentro del framework Vue para el desarrollo front-end por su rápida curva de aprendizaje lo que

permitió un rápido desarrollo y permitirá de ser necesario la rápida inclusión de desarrolladores en futuras fases, además permite la ampliación de funciones mediante el uso de su manejador de librerías llamado NPM.

Para el desarrollo back-end se utilizó el framework Laravel puesto que utiliza lenguaje gratuito e incluye la posibilidad de ampliar sus funciones mediante el manejador de paquetes composer e incluye un ORM que permite la integración con la base de datos.

Para la base de datos se usó MySQL que incluye integración directa con el framework Laravel.

Para la depuración del software se utiliza Xdebug para el lenguaje PHP que viene integrado en el IDE PHPstorm.

Se utiliza el IDE Visual Studio Code por su capacidad de instalar plugins que permite formatear código del framework Vue.

Para dar estilos se incorpora el framework Tailwind css por ser muy liviano y flexible.

Metodología del proceso de desarrollo de la propuesta tecnológica

Capítulo 2

Metodología del proceso de desarrollo de la propuesta tecnológica

Metodología de desarrollo

Para el desarrollo se usó el framework scrum que permite el desarrollo incremental e iterativo en el desarrollo del producto. La herramienta se puede usar para producir productos complicados. Es "un plan de desarrollo de producto flexible. No prescribe ni una de las prácticas usuales de ingeniería, personas, administración de peligros u otras prácticas. Ya que Scrum no explica explícitamente ni una práctica de ingeniería, es viable y, algunas veces, deseable además tener en cuenta no Prácticas de Scrum que tienen la posibilidad de estar estrechamente asociadas al éxito de Scrum. (Sachdeva, 2016)

Enfoque de la investigación

La metodología a emplearse para el siguiente proyecto es el método de investigación cualitativa, ya que la información obtenida para realizarlo no se basa en datos estadísticos sino más bien ha sido tomada de entrevistas que se ha hecho a personal académico y administrativo de la universidad Ecotec.

Tipo de Investigación

La investigación a desarrollarse en primera fase es de ámbito exploratorio, ya que se está examinando un problema que ha sido expresado por las personas involucradas con la universidad en las entrevistas realizadas, determinando las necesidades que se desean satisfacer para un mejor control de sus archivos. Como segunda instancia se realizará un método descriptivo el cual permita con las indicaciones dadas desarrollar un software que cumpla con las expectativas deseadas para poder mejorar en manejo y control de ciertas áreas de la universidad Ecotec.

Período y lugar donde se desarrolla la propuesta tecnológica

El lugar para el cual se desarrollará este proyecto es la facultad de ingeniería de la universidad Ecotec que empezó el 15 de marzo del 2021 y que terminará el 11 de junio del 2021.

Universo y muestra

- El universo de este estudio son los profesores y la decana encargada de la Facultad de Ingeniería de la Universidad Ecotec.
- La muestra de ese proyecto son 4 profesores de la facultad de ingeniería y la decana encargada de la misma.

Definición y comportamiento de las principales variables incluidas en el estudio

Conceptualización:

Definición de variables

Es una propiedad que puede cambiar y cuya variación está dispuesta a adoptar varios valores, los cuales pueden medirse u observarse. Las variables cuando se relacionan con otras es que adquieren un valor, es decir cuando forman parte de una teoría o hipótesis.

Variable independiente

Es la que se modifica o es intervenida para analizar sus efectos. Esta variable condiciona a la variable dependiente, se la nombra como causal o experimental ya que es operada por el investigador.

Variable dependiente

El eje del ensayo es la variable independiente está aislada y manejada por el investigador. El resultado medible de esta manipulación es la variable dependiente. Esta variable es la que se puede medir e investigar. (Villasís-Keever, & Miranda-Novales, 2016)

Operacionalización:

Tabla 1: Variables

VARIABLE	Dimensión	Definición	Indicadores	Instrumento
Independiente: Proceso de control y almacenamiento de documentos	Docentes	Documentos solicitados por encargados de los procesos de titulación, prácticas pre profesional, vinculación con la sociedad.	Documentos solicitados en procesos.	Observación directa
Dependiente: Creación de Software de control de archivos	Cronograma	Actividades semanales a realizar	Reporte de cronograma	Entrevistas
	Pruebas funcionales	Pruebas realizadas por los encargados de cada área	Observaciones	Entrevistas
	Almacenamientos de documentos	Carga y descarga de archivos	Existencia de archivos en el servidor	Observación directa

Fuente: Elaboración propia

Métodos empleados e instrumentos de la investigación.

Para la elaboración de este proyecto se utilizó el método empírico el cual permite obtener y elaborar datos con los cuales darán pie para la creación del proyecto a desarrollar. Se utilizó como herramienta primordial las encuestas las cuales se realizaron a personal docente de la facultad de ingeniería con el fin de conocer los inconvenientes que a ellos se les presentaban y cuáles serían las necesidades que necesitarían cubrir.

Procesamiento y análisis de la información.

Para desarrollar el Proyecto se realizó una entrevista inicial en la que participaron representantes de los procesos más importantes en la que intervienen uno o más

docentes, con uno o más alumnos. El alumno es el que está obligado a cumplir con una cantidad de requerimientos de documentos como constancia del cumplimiento del proceso.

En esta primera reunión se determinó quien necesita esta herramienta. Se evaluó el problema general que tenían los docentes y la herramienta que se usa para resolver este problema. Se determinó los usuarios de la herramienta actual. Se socializó las limitantes que presenta el software actual para controlar los requerimientos de cada proceso. Se habló sobre la capacitación recibida para usar el software y si existía algún requerimiento para poder usarlo. Las plataformas en donde se usa actualmente el software y la expectativa de usabilidad en diferentes plataformas del nuevo software. Cuál es la capacitación esperada para aprender sobre el nuevo software. Finalmente se determinó si existían requisitos legales o administrativos para la implementación del software.

Como resultado de esta entrevista y posterior a una actividad de lluvia de ideas se determinó las siguientes historias de usuario que se convierten en requerimiento para el nuevo software:

- Como encargado del proceso quiero que los requerimientos de archivos tengan una fecha de vencimiento para que el usuario alumno pueda subir el archivo solicitado dentro de los plazos establecidos.
- Como encargado del proceso necesito limitar los tipos de archivos permitidos en caso de ser necesario para poder controlar los archivos esperados.
- Como encargado del proceso necesito que los alumnos usen una plantilla predefinida para mantener uniformidad con la información requerida, esta plantilla debe poder ser descargada por el alumno según el requerimiento solicitado.
- Como docente encargado es importante conocer rápidamente la cantidad de archivos cumplidos por cada alumno, cuales alumnos no han subido el documento y cuales les falta para así poder retroalimentarlos de ser el caso.

- Como docente encargado del proceso necesito poder acceder desde mi computadora, tableta o celular para poder revisar incluso si no estoy cerca de mi computador.
- Como docente quiero que archivo solicitado sea único y no existan duplicados para evitar confusión en la revisión de un archivo que subió el alumno pero se revisó otro.
- Como docente quiero que el alumno reciba una alerta cuando se haya quedado en recuperación para que esté atento a subir el documento y no se retrase.
- Como docente dueño del proceso quiero que los alumnos no puedan acceder a la documentación de otros alumnos para evitar que se eliminen carpetas o archivos por accidente, para esto cada archivo debe poder ser accedido solo por el usuario dueño y el docente encargado del proceso.
- Como creador de procesos deseo que llegue un mail al correo institucional del alumno en donde se le informe que está participando en el proceso y cuando se inicie un requerimiento de documentos, de esta forma el alumno estará informado de sus obligaciones.
- Como docente necesito poder buscar archivos según una categoría o palabra clave.
- Como administrador necesito que existan roles tanto de docente como de alumno para limitar las opciones a las que puede acceder.
- Como administrador del proceso necesito que los alumnos accedan usando el correo institucional. Para cumplir con este requerimiento el registro de los usuario debe estar limitado únicamente a los roles de docentes y no es de libre registro.
- Como administrador necesito que el alumno pueda descargar solicitudes para realizar trámites como retiro de materia, exámenes atrasados par que tenga disponible el formato en todo momento.

- Como docente necesito editar formatos por ejemplo el Excel de las clases y subirlo nuevamente.
- Como docente necesito saber que alumno se descargó un documento para mantener control.
- Como docente necesito que los syllabus puedan ser descargados en Word para poder actualizar los campos.
- Como revisor de los documentos necesito saber el usuario responsable de subir el documento.
- Como docente necesito que se pueda validar los usuarios contra la BD de la universidad.
- Como tutor de tesis necesito poder acceder a un repositorio de tesis aprobados, clasificados por carrera y línea de investigación para ser usados como guía o referencia de nuevos estudiantes.
- Como revisor necesito visualizar el cumplimiento de los requerimientos de cada alumno y los documentos faltantes.

El requerimiento de notificación cuando el alumno se quede en recuperación se descarta puesto que el sistema pretende almacenar documentos y no registrar en que materias está inscrito el alumno condición necesaria para conocer el estado de su materia.

El vínculo con la base de datos de la universidad y el software se realiza haciendo un match con el número de matrícula.

En una segunda reunión en este caso con usuarios alumnos que ya pasaron por estos procesos se solicitó retroalimentación sobre mejoras de la experiencia.

Para ellos es importante:

- Como alumno quiero subir la documentación desde cualquier dispositivo sea computadora o celular.

- Como alumno quiero acceder a plantillas y formatos guías que me permitan cumplir con el requerimiento.
- Como alumno quiero que mis documentos sensibles como cedula de identidad se manejen con mayor privacidad.
- Como alumno quiero conocer las fechas límites para cumplir con el requerimiento.
- Como alumno quiero poder subir, editar y eliminar mi requerimiento mientras este dentro del tiempo para así no pedirle al docente que elimine el archivo y poder subirlo nuevamente.
- Como alumno quiero poder recuperar mi contraseña si no la recuerdo, para así no depender del área de sistemas para recuperar mi cuenta.

En una tercera reunión con tutor de tesis para recopilar información sobre la documentación necesaria y pruebas en vivo de los módulos desarrollados para la implementación del proyecto. Se realiza un alcance en el proyecto en donde se amplía el uso para que los docentes también puedan subir archivos en los procesos.

En la cuarta reunión en esa ocasión de forma individual con cada uno de los encargados de las áreas y la decana para revisión, realizar las pruebas de cumplimientos de características y aceptación del producto. En esta reunión se afinaron temas para mejorar la estética y experiencia de usuario. Se acepta funcionalidad del software.

Para realizar el software se usó una de las metodologías ágiles. En este caso scrum, se crearon las historias de usuario producto de las reuniones y que sirven como requerimiento.

Análisis e interpretación de los resultados

Capítulo 3

Análisis e interpretación de los resultados

Después de realizar el levantamiento de la información se procedió a crear el backlog con los módulos a realizar. Para esto se utilizó la herramienta web trello separando tres tableros. Módulo por desarrollar, en proceso y terminado. Como se observa en la Ilustración 1.

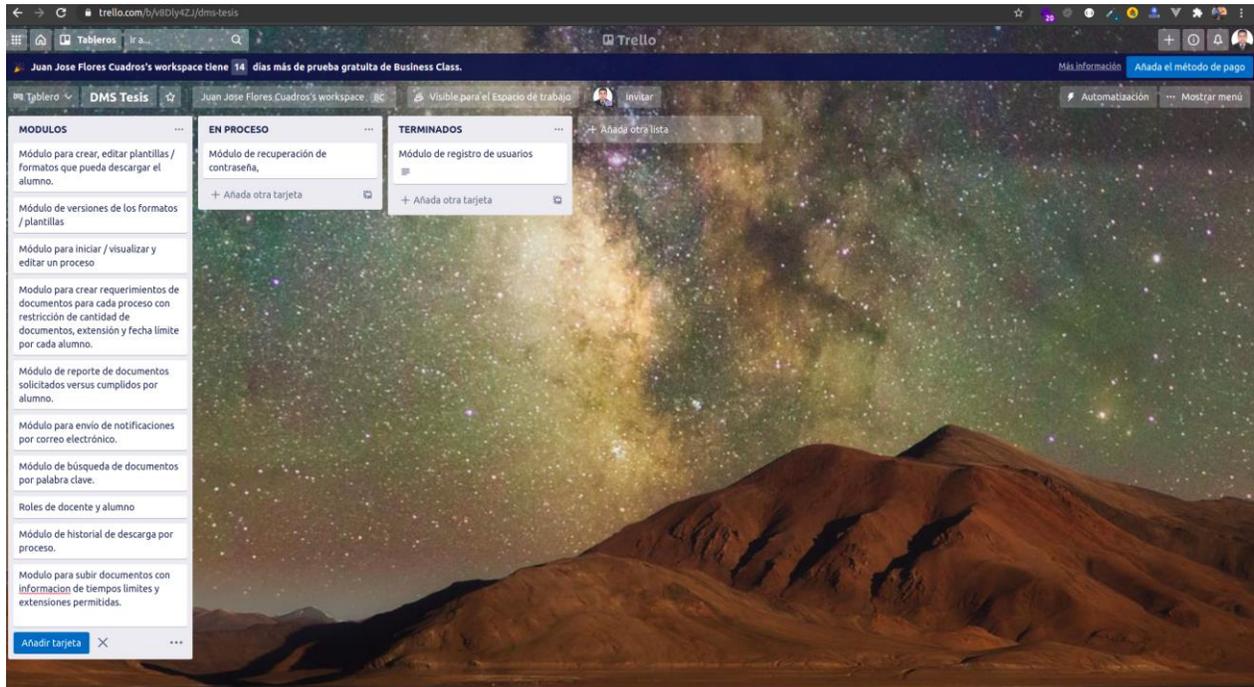


Ilustración 1 Backlog

Fuente: Elaboración propia

Para el módulo de ingreso se usó el correo electrónico como nombre de usuario. El mismo módulo de ingreso permite recuperar la contraseña en caso de pérdida tal como se puede observar en la ilustración 2.

Correo

uflores@est.ecotec.edu.ec

Contraseña

.....

Recordame en este dispositivo

[¿Olvidaste la contraseña?](#)

INGRESAR

Ilustración 2 Ingreso del software

Fuente: Elaboración propia

Por cada módulo se muestra una pantalla principal que contiene el total de los registros en base de datos y un campo que permite filtrar las búsquedas como se observa en la ilustración 3. Cada pantalla posee un paginado para acelerar la respuesta de las peticiones del total de registros.

El módulo de usuario posee un mantenedor que permite crear y editar usuario y sus roles como se observa en la ilustración 4.

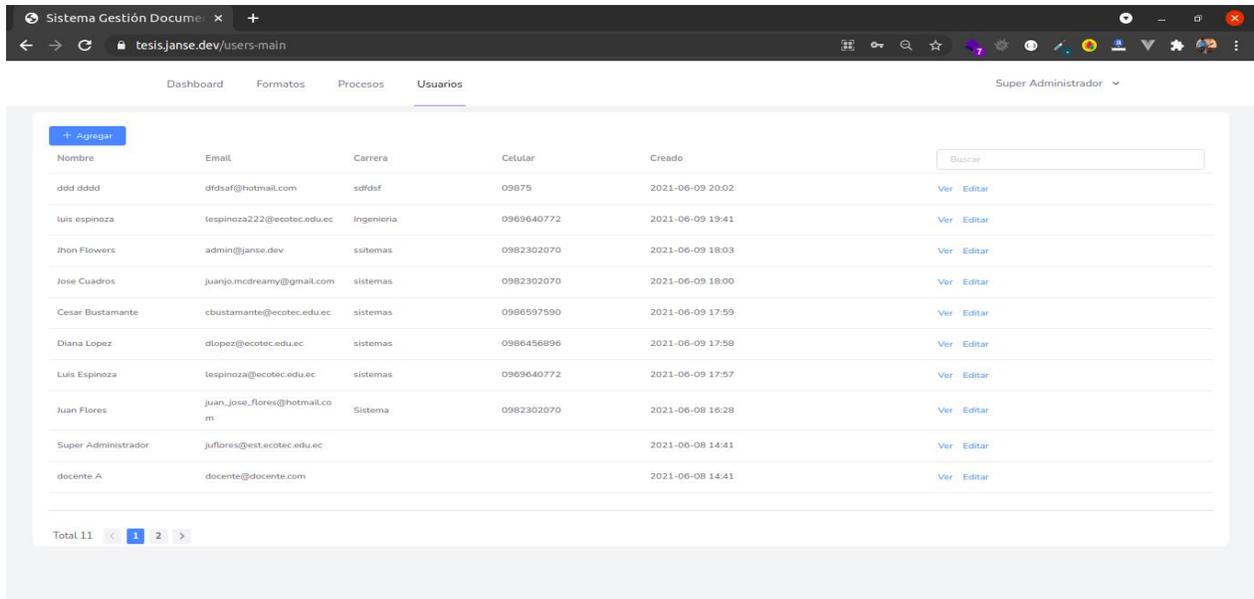


Ilustración 3 Pantalla del listado de usuarios

Fuente: Elaboración propia

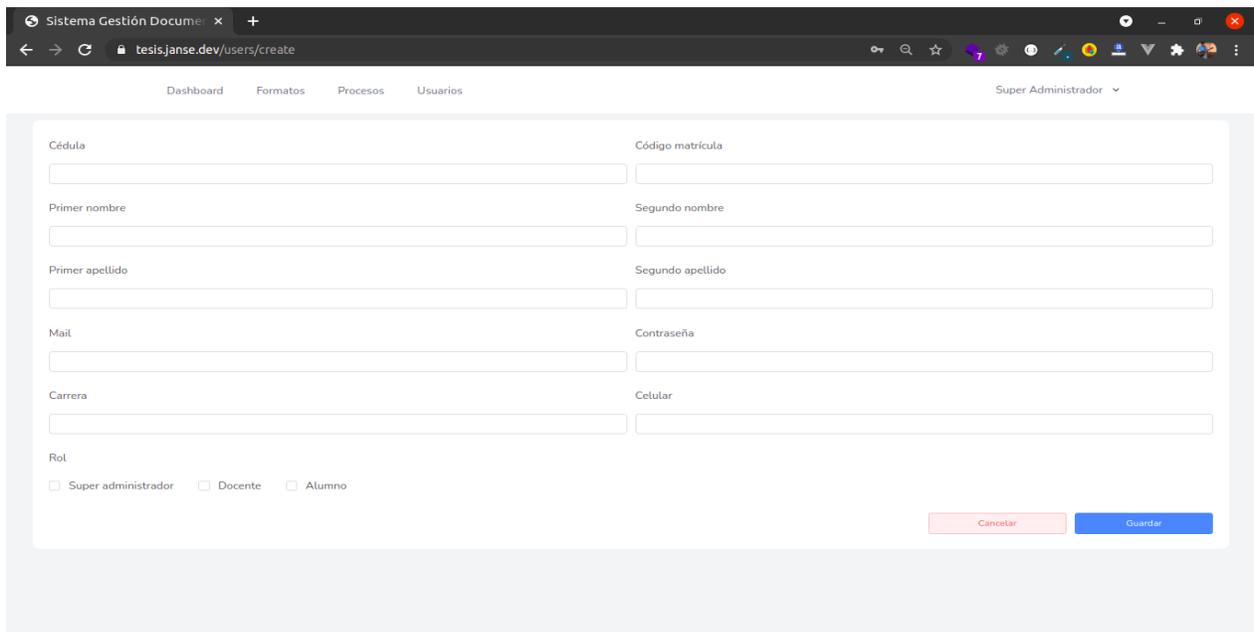


Ilustración 4 Mantenedor para crear / editar un usuario

Fuente: Elaboración propia

El módulo de procesos en su pantalla principal muestra el total de procesos paginados por cada 10 registros como se observa en la ilustración 5. Permite crear los procesos en donde se configuran los participantes, representantes de proceso (generalmente docentes) como los usuarios invitados al proceso (generalmente alumnos) detallados en la ilustración 6. Permite crear los requerimientos de documentos y sus respectivas restricciones por documento solicitado, acceso a reportes y el mantenedor como se ve en la ilustración 7.

En la ilustración 8 se puede observar la sección donde se definen las reglas como son el poder limitar las extensiones que el alumno puede subir el documento, las fechas límite de cumplimiento y la inclusión de un formato guía en caso de necesitarlo.

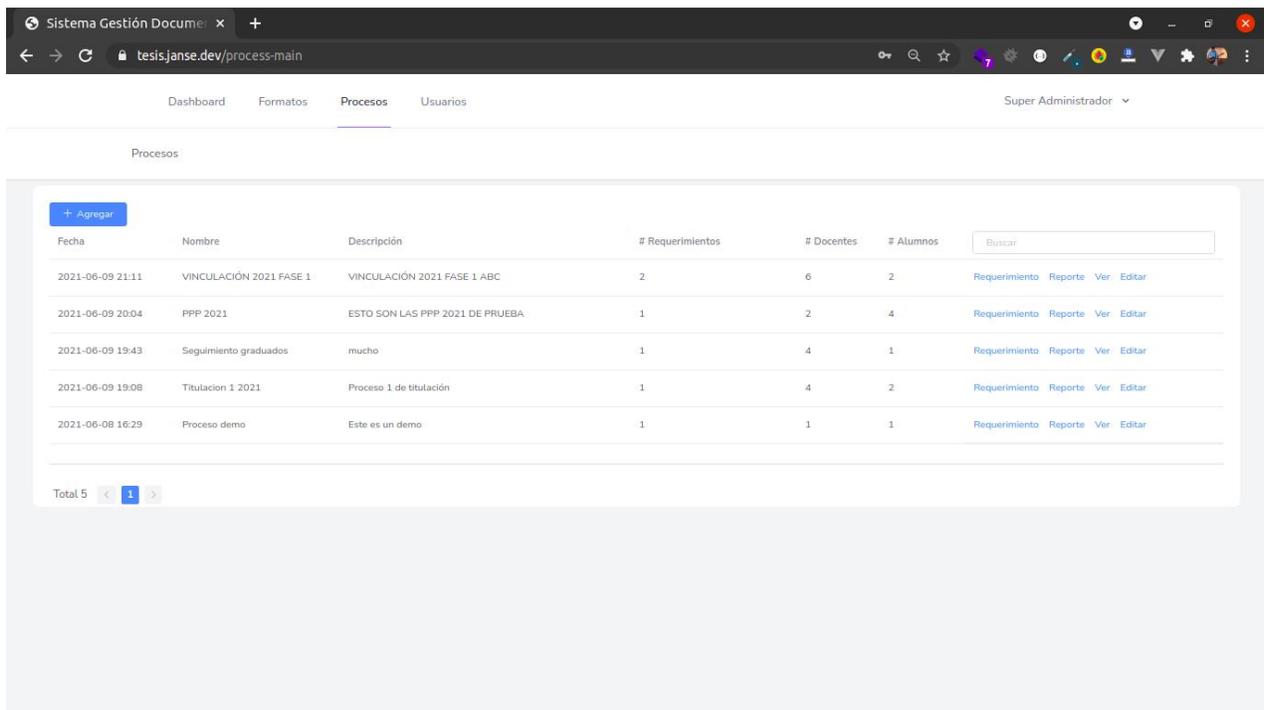


Ilustración 5 Pantalla principal del módulo de procesos

Fuente: Elaboración propia

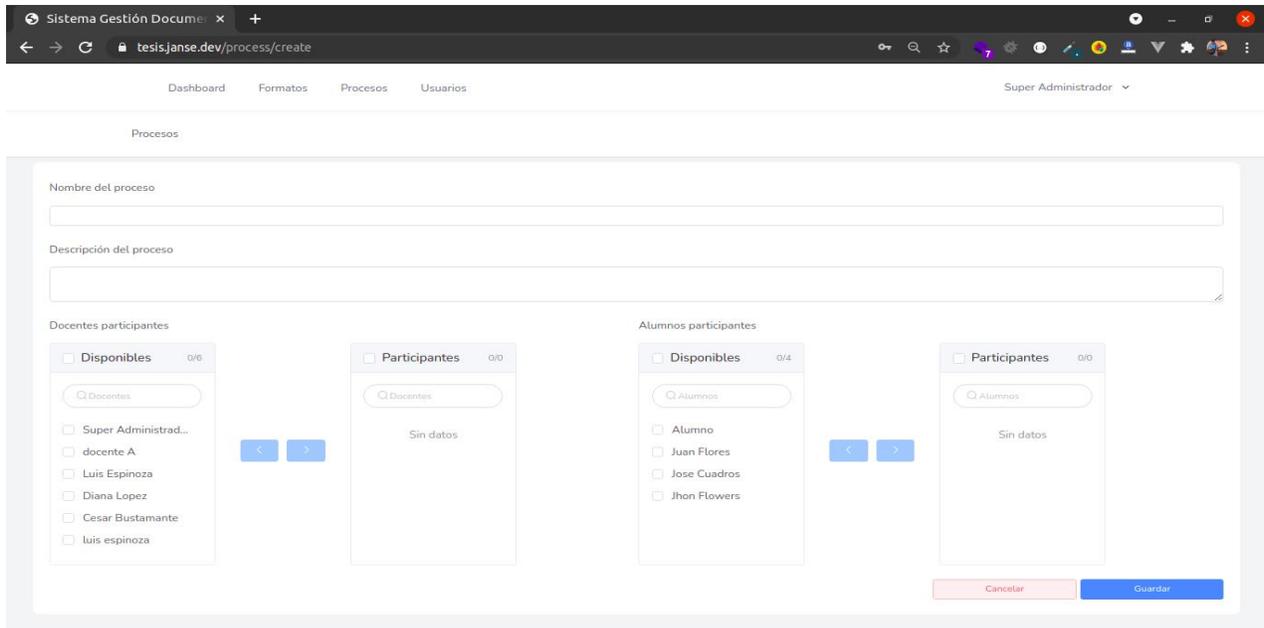


Ilustración 6 Mantenedor de procesos

Fuente: Elaboración propia

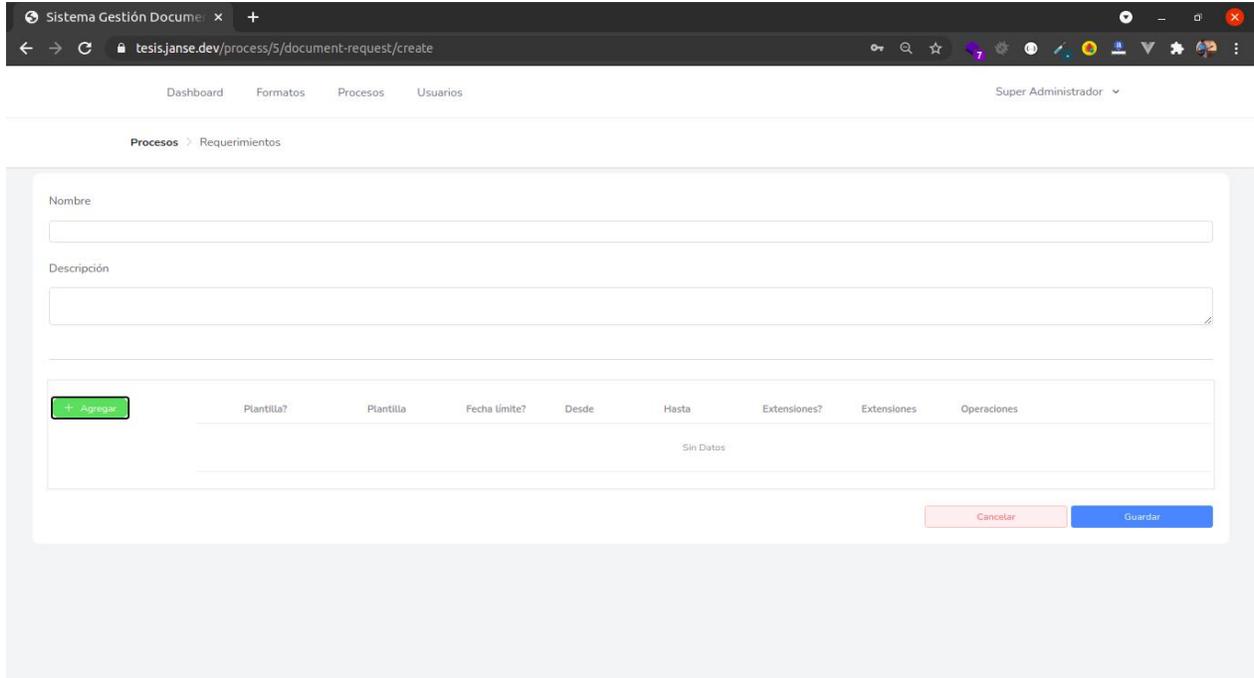


Ilustración 7 Mantenedor de requerimientos

Fuente: Elaboración propia

Definir reglas para subir documento ✕

Descripción

Tiene fecha límite?

Si tiene.

Tiene plantilla?

Si tiene.

Requiere formato?

Solo las siguientes extensiones.

Cancelar solicitud
Agregar solicitud

Ilustración 8 Reglas/ restricciones para cada archivo

Fuente: Elaboración propia

Dentro del módulo procesos también se encuentra el reporte que permite ver el cumplimiento de archivos subidos en cada requerimiento. Este módulo se puede desplegar para ver rápidamente cual es el documento faltante y los cumplidos tal como se detalla en la ilustración 9.

Nombre	Correo	Documentos solicitados	Documentos subidos
Alumno	alumno@alumno.com	2	0
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>Pendiente</p> <p>Pendiente</p> </div> <div style="width: 15%;"> <p>Descargar</p> <p>Descargar</p> </div> <div style="width: 40%;"> <p>PDF.</p> <p>CERTIFICADO 2</p> </div> </div>			
Jose Cuadros	juanjomcdreamy@gmail.com	2	0
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>Pendiente</p> <p>Pendiente</p> </div> <div style="width: 15%;"> <p>Descargar</p> <p>Descargar</p> </div> <div style="width: 40%;"> <p>PDF.</p> <p>CERTIFICADO 2</p> </div> </div>			
Jhon Flowers	admin@janse.dev	2	0
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>Pendiente</p> <p>Pendiente</p> </div> <div style="width: 15%;"> <p>Descargar</p> <p>Descargar</p> </div> <div style="width: 40%;"> <p>PDF.</p> <p>CERTIFICADO 2</p> </div> </div>			
Juan Flores	juan_jose_flores@hotmail.com	2	1
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>Ok</p> <p>Pendiente</p> </div> <div style="width: 15%;"> <p>Descargar</p> <p>Descargar</p> </div> <div style="width: 40%;"> <p>PDF.</p> <p>CERTIFICADO 2</p> </div> </div>			

Ilustración 9 Reportes de cumplimiento

Fuente: Elaboración propia

El módulo de formatos permite subir plantillas para que el usuario pueda descargar y usarlo como guía, estos formatos pueden ser de acceso público o solo para usuarios registrados, según lo configure el dueño del documento. Adicionalmente cuenta con la opción de tener versionados de los archivos y la opción de seleccionar el que esté vigente como se detalla en la ilustración 10.

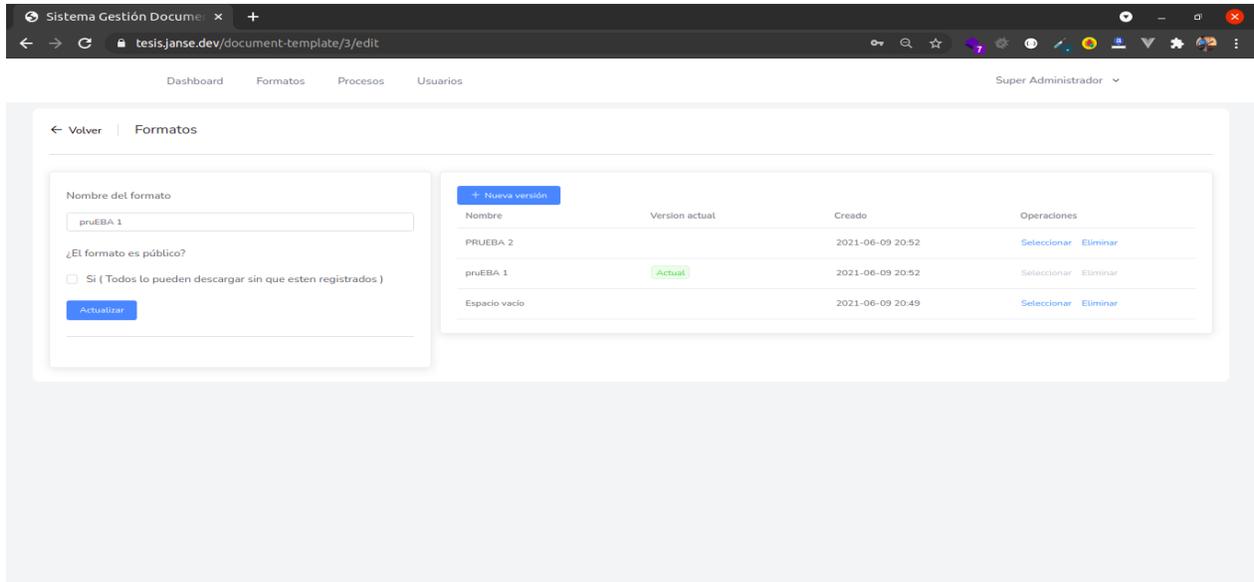


Ilustración 10 Mantenedor de formatos con versiones

Fuente: Elaboración propia

Por parte del usuario que sube documentos, solo tiene acceso a sus procesos y acceso únicamente a subir, eliminar y descargar sus archivos como se detalla en la ilustración 11.

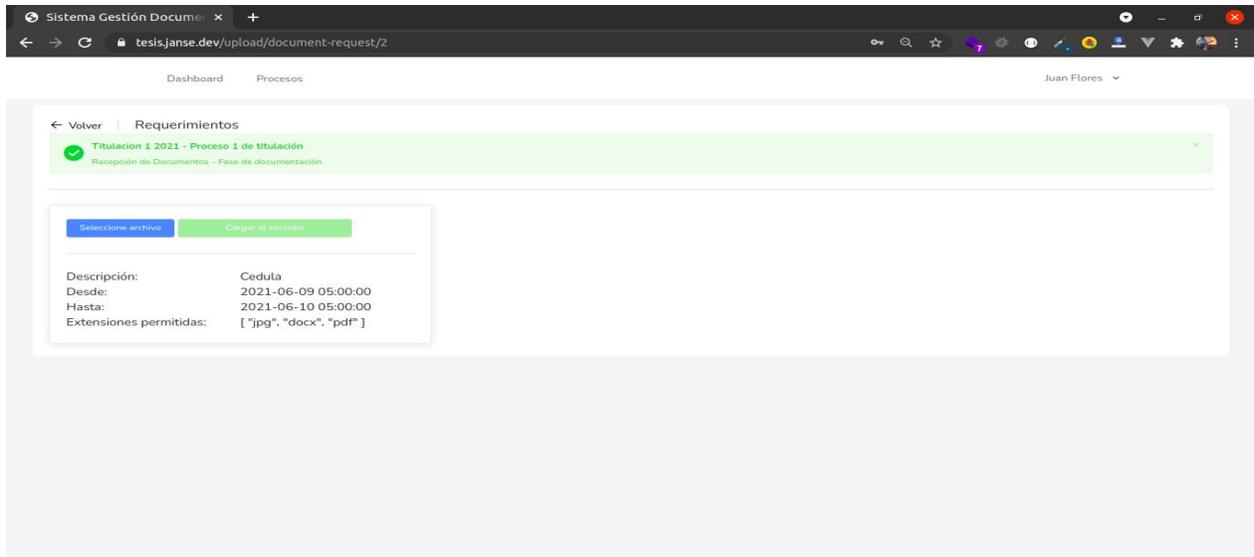


Ilustración 11 Pantalla para subir documentos del usuario alumno

Fuente: Elaboración propia

Para las pruebas de validación se pensó en una experiencia de usuario que permita identificar el error de validación cometido detallándolo de forma clara como se observa en la ilustración 12 y cercano al campo en el cual se cometió el error como se detalla en la ilustración 13.

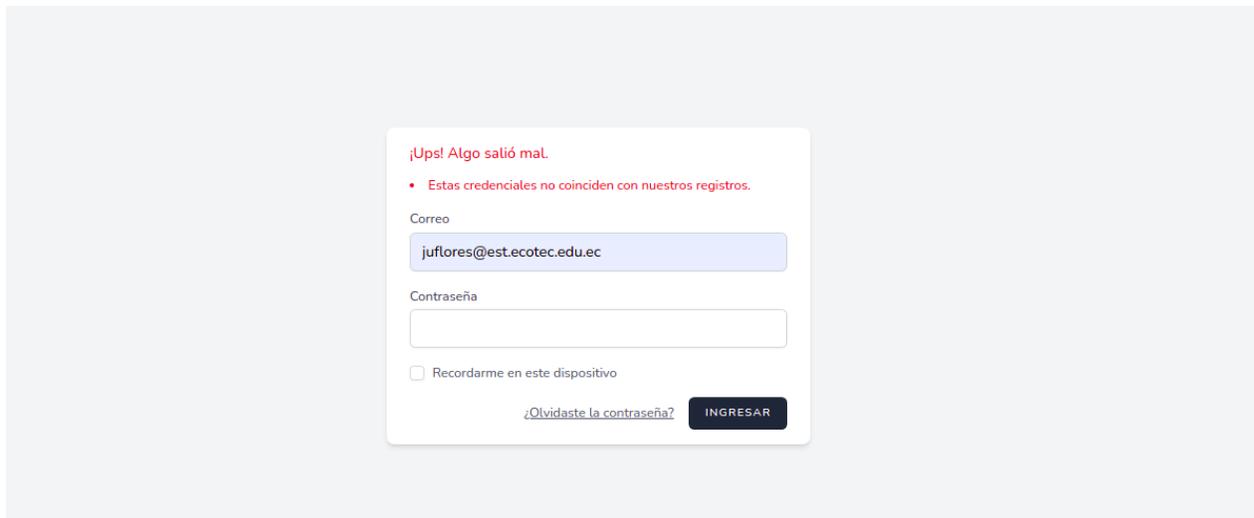


Ilustración 12 Validando formulario de ingreso

Fuente: Elaboración propia

Dashboard Formatos Procesos Usuarios Super Administrador ▾

Cédula	Código matrícula
<input type="text"/>	<input type="text"/>
	<small>["El campo Código matrícula es obligatorio."]</small>
Primer nombre	Segundo nombre
<input type="text"/>	<input type="text"/>
<small>["El campo Primer nombre es obligatorio."]</small>	
Primer apellido	Segundo apellido
<input type="text"/>	<input type="text"/>
<small>["El campo Primer apellido es obligatorio."]</small>	
Mail	Contraseña
<input type="text"/>	<input type="text"/>
<small>["El campo email es obligatorio."]</small>	<small>["El campo Contraseña es obligatorio."]</small>
Carrera	Celular
<input type="text"/>	<input type="text"/>
<small>["El campo Carrera es obligatorio."]</small>	<small>["El campo Celular es obligatorio."]</small>
Rol	
<input type="checkbox"/> Super administrador <input type="checkbox"/> Docente <input type="checkbox"/> Alumno	
	<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>

Ilustración 13 Validando campos requeridos

Fuente: Elaboración propia

Para las pruebas de aceptación se adjunta el anexo 1 en donde se detallan las pruebas realizadas y la respectiva firma de aceptación de parte del usuario final.

Propuesta

Capítulo 4

Propuesta

Diccionario de datos

Tabla 2: Tabla abilities

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
Title	varchar(255)		
entity_id	bigint(20) unsigned		
entity_type	varchar(255)		
only_owned	tinyint(1)		
Options	Longtext		
Scope	int(11)	MUL	
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 3: Tabla assigned_roles

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
role_id	bigint(20) unsigned	MUL	
entity_id	bigint(20) unsigned	MUL	
entity_type	varchar(255)		
restricted_to_id	bigint(20) unsigned		
restricted_to_type	varchar(255)		
Scope	int(11)	MUL	

Fuente: Elaboración propia

Tabla 4: Tabla careers

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
school_id	bigint(20) unsigned	MUL	
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 5: Tabla categories

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 6: Tabla document_request_rules

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
process_id	bigint(20) unsigned	MUL	
document_request_id	bigint(20) unsigned	MUL	
Description	varchar(255)		
has_model	tinyint(1)		
document_template_id	bigint(20) unsigned	MUL	
has_deadline	tinyint(1)		
Since	datetime		
Until	datetime		
has_ext	tinyint(1)		
Ext	longtext		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 7: Tabla document_requests

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
process_id	bigint(20) unsigned	MUL	
Name	varchar(255)		
Description	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 8: Tabla document_template_versions

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
document_template_id	bigint(20) unsigned	MUL	
Name	varchar(255)		
Path	varchar(255)		
user_id	bigint(20) unsigned	MUL	
is_public	tinyint(1)		
Ext	varchar(255)		
Size	int(11)		
category_id	bigint(20) unsigned		
Keywords	varchar(255)		
Current	tinyint(1)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 9: Tabla document_templates

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
Path	varchar(255)		
user_id	bigint(20) unsigned	MUL	
is_public	tinyint(1)		
Ext	varchar(255)		
Size	int(11)		
category_id	bigint(20) unsigned		
Keywords	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 10: Tabla documents

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
Path	varchar(255)		
user_id	bigint(20) unsigned	MUL	
process_id	bigint(20) unsigned	MUL	
document_request_id	bigint(20) unsigned	MUL	
document_request_rule	int(11)		
public_level	int(11)		
category_id	bigint(20) unsigned		
Keywords	varchar(255)		
Ext	varchar(255)		
Size	int(11)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 11: Tabla failed_jobs

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Uuid	varchar(255)	UNI	
Connection	text		
Queue	text		
Payload	longtext		
Exception	longtext		
failed_at	timestamp		

Fuente: Elaboración propia

Tabla 12: Tabla menus

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
rol_id	bigint(20) unsigned	MUL	
url	varchar(255)		
Order	varchar(255)		
created_at	timestamp		
updated_at	timestamp		
failed_at	timestamp		

Fuente: Elaboración propia

Tabla 13: Tabla migrations

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	int(10) unsigned	PRI	auto_increment
Migration	varchar(255)		
Batch	int(11)		

Fuente: Elaboración propia

Tabla 14: Tabla password_resets

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Email	varchar(255)	MUL	
Token	varchar(255)		
created_at	timestamp		

Fuente: Elaboración propia

Tabla 15: Tabla permissions

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
ability_id	bigint(20) unsigned	MUL	
entity_id	bigint(20) unsigned	MUL	
entity_type	varchar(255)		
Forbidden	tinyint(1)		
Scope	int(11)	MUL	

Fuente: Elaboración propia

Tabla 16: Tabla personal_access_tokens

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
tokenable_type	varchar(255)	MUL	
tokenable_id	bigint(20) unsigned		
Name	varchar(255)		
Token	varchar(64)	UNI	
Abilities	text		
last_used_at	timestamp		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 17: Tabla process_heads

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
process_id	bigint(20) unsigned	MUL	
user_id	bigint(20) unsigned	MUL	
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 18: Tabla process_users

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
process_id	bigint(20) unsigned	MUL	
user_id	bigint(20) unsigned	MUL	
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 19: Tabla processes

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
Description	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 20: Tabla roles

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)	MUL	
Title	varchar(255)		
Level	int(10) unsigned		
Scope	int(11)	MUL	
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 21: Tabla schools

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 22: Tabla sessions

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	varchar(255)	PRI	
user_id	bigint(20)	MUL	
	unsigned		
ip_address	varchar(45)		
user_agent	text		
Payload	text		
last_activity	int(11)	MUL	

Fuente: Elaboración propia

Tabla 23: Tabla user_details

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
user_id	bigint(20) unsigned	MUL	
identification	varchar(255)		
Code	varchar(255)		
first_name	varchar(255)		
second_name	varchar(255)		
Surname	varchar(255)		
second_surname	varchar(255)		
Career	varchar(255)		
is_head	tinyint(1)		
Cellphone	varchar(255)		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Tabla 24: Tabla users

ATRIBUTO	TIPO DE DATO	PK	EXTRA
Id	bigint(20) unsigned	PRI	auto_increment
Name	varchar(255)		
Email	varchar(255)	UNI	
email_verified_at	timestamp		
Password	varchar(255)		
two_factor_secret	Text		
two_factor_recovery_codes	Text		
remember_token	varchar(100)		
current_team_id	bigint(20) unsigned		
profile_photo_path	Text		
created_at	timestamp		
updated_at	timestamp		

Fuente: Elaboración propia

Modelo entidad-relación

En la ilustración 14 se observa las tablas de la base de datos con sus relaciones.

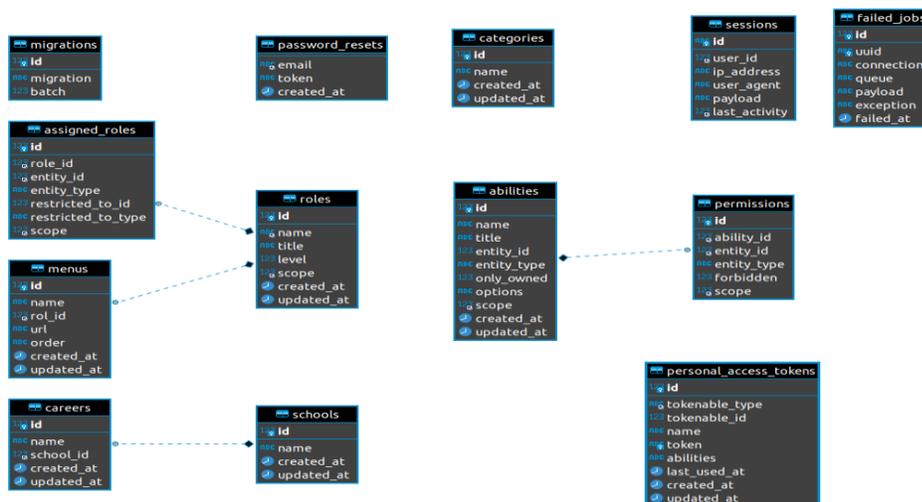
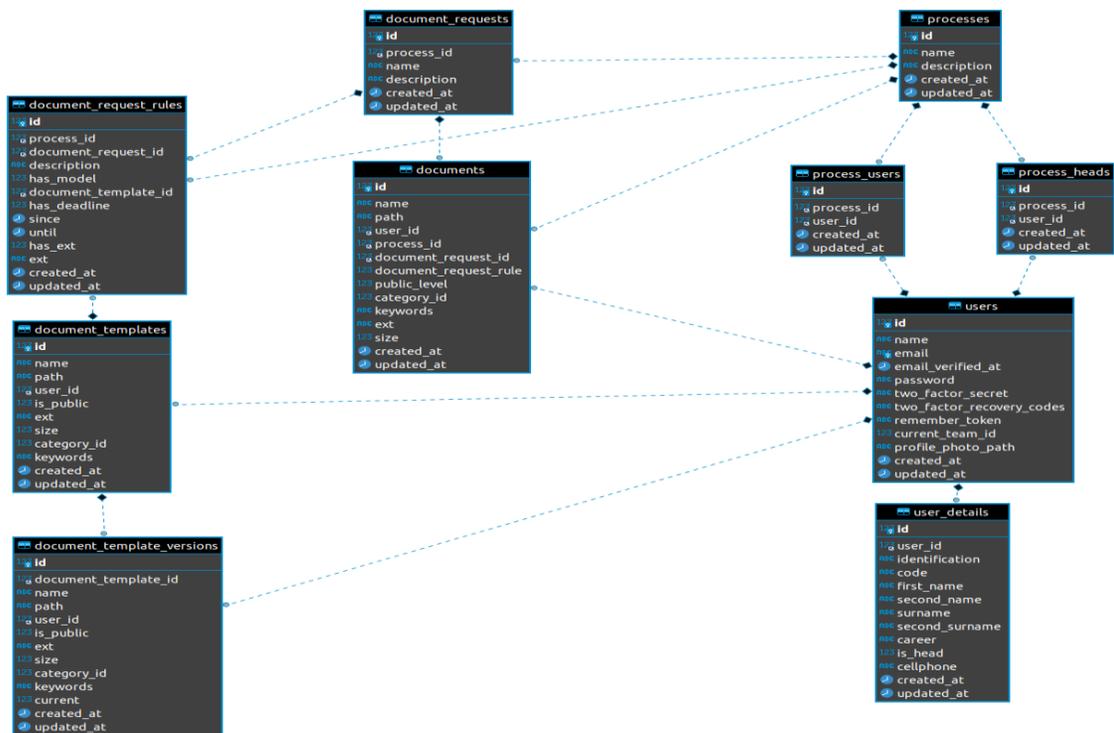


Ilustración 14 Modelo entidad -relación

Fuente: Elaboración propia

Sistema operativo

El proyecto necesita un servidor web, se recomienda usar apache server, el cual puede ser instalado en cualquier sistema operativo Linux / Windows. En el servidor en donde se implementó se utilizó Linux Ubuntu 20.04. Posteriormente se instaló dos dependencias que necesita el framework laravel para poder ser ejecutado. Estos son composer versión 2 y node.js el cual también instala el programa npm el cual es un manejador de paquetes.

Framework Laravel

Para el desarrollo del software se usó el framework Laravel que utiliza el lenguaje PHP, el cual ya trae una estructura guía de carpetas como se observa en la ilustración 13. Está basado en el patrón de diseño de arquitectura modelo vista controlador como capas básicas. Esto permite tener una estructura muy organizada y que permite una fácil adaptación para el trabajo en equipo por cuanto usa la especificación de PHP que es una recomendación estándar llamada PSR.

En la carpeta app es en donde se colocaron los controladores. Cada archivo es un módulo en donde se encuentra la lógica de negocio en caso de ser necesario se conecta a un modelo que tiene las instrucciones para conectarse con la base de datos y regresar una vista al usuario. Otras carpetas como mailable corresponden a patrones de diseño orientado a objetos el cual permite instanciar una clase para enviar una notificación. Adicionalmente se le agrega una estructura de controladores llamados middlewares que sirven de filtro para las peticiones http. Este controlador permite el manejo de roles y permisos en cada petición del usuario.

En la carpeta database se alojaron las migraciones que utiliza la metodología código primero (first code). Esta metodología permite crear las tablas con todas sus columnas utilizando el comando “php artisan make:migrations create_nombre_de_la_tabla_table” de esta forma todo el proyecto está integrado y no es necesario ejecutar scripts por separado, aunque de ser necesario el framework lo permite. Adicionalmente se puede crear datos semillas que generan datos iniciales y por medio del uso del patrón Factory se puede crear datos falsos muy útiles durante la etapa de prueba para las pruebas funcionales y pruebas de estrés.

En la carpeta resources se almacena todos los archivos para la interfaz gráfica del usuario y las traducciones de los mensajes de validación ya que por defecto laravel muestra cualquier mensaje de la aplicación en idioma inglés. Para la parte de frontend se usó el framework vuejs que permite la interacción y reactividad con los usuarios. Vuejs se encuentra dentro de los 3 frameworks de JavaScript con mayor aceptación entre los desarrolladores junto a svelte y react.

En la carpeta storage están almacenados todos los documentos subidos usando el software, el acceso a los archivos no son de acceso público sino más bien requiere de una petición a un controlador y este es quien valida si quien solicita puede acceder a descargar el documento. De ser aprobado se envía una respuesta de tipo arraybuffer que contiene el archivo a ser descargado. La estructura interna que tiene es de una carpeta con la clave primaria del proceso, una subcarpeta con la clave primaria del requerimiento y finalmente una última carpeta llamada con la clave primaria del usuario que creo el documento, esa estructura es muy parecida a la que actualmente se usa.

En la carpeta vendor están todas las dependencias y librerías del proyecto que necesita laravel para poder ejecutarse. Para el desarrollo de la aplicación se usaron las siguientes dependencias con sus respectivas versiones:

Tabla 25: Tabla de dependencias laravel

LIBRERÍA	VERSION
Php	7.3 8.0
fideloper/proxy	4.4.0
fruitcake/laravel-cors	2.0.0
guzzlehttp/guzzle	7.0.1
inertiajs/inertia-laravel	0.3.5
laravel/framework	8.12.0
laravel/jetstream	2.3.0
laravel/sanctum	2.6.0
laravel/tinker	2.5.0
laraveles/spanish	1.5.0
silber/bouncer	v1.0.0-rc.10
tightenco/ziggy	1.0.0

Fuente: Elaboración propia

El código fuente está alojado en el repositorio GitLab: <https://gitlab.com/juanjo1307/dms> lo cual permite instalarlo usando internet. Para esto digitamos el comando “git clone <https://gitlab.com/juanjo1307/dms>” desde la terminal de comandos.

Una vez que se clonó el proyecto ingresamos a la carpeta y copiamos el archivo de variables globales el cual tiene una plantilla de ejemplo en el directorio raíz. Para esto usamos el comando “cp .env.example .env” en este archivo están las configuraciones generales del proyecto como son el nombre de la base de datos y el correo que usaremos para enviar las notificaciones.

Generamos la clave de la aplicación la cual sirve para generar un token que se enviará junto a todos los formularios para evitar ataques de tipo XSS. Para esto ejecutamos el comando “php artisan key:generate”.

Después de colocar el nombre de la base de datos y la contraseña en el archivo `.env` es necesario ejecutar las migraciones con el comando `php artisan migrate --seed`, este comando junto a el atributo `seed` creará todas la tablas y columnas necesarias para el funcionamiento del proyecto y adicionalmente creará datos semillas como es el caso del menú dinámico, un primer usuario; el cual será el súper administrador que tiene la clave por defecto `password` y será lo primero a cambiar después de la instalación. Además creará tres roles predefinidos que son usados para el súper administrador, docente y alumno. En la ilustración 15 se puede observar la estructura de carpetas del proyecto.

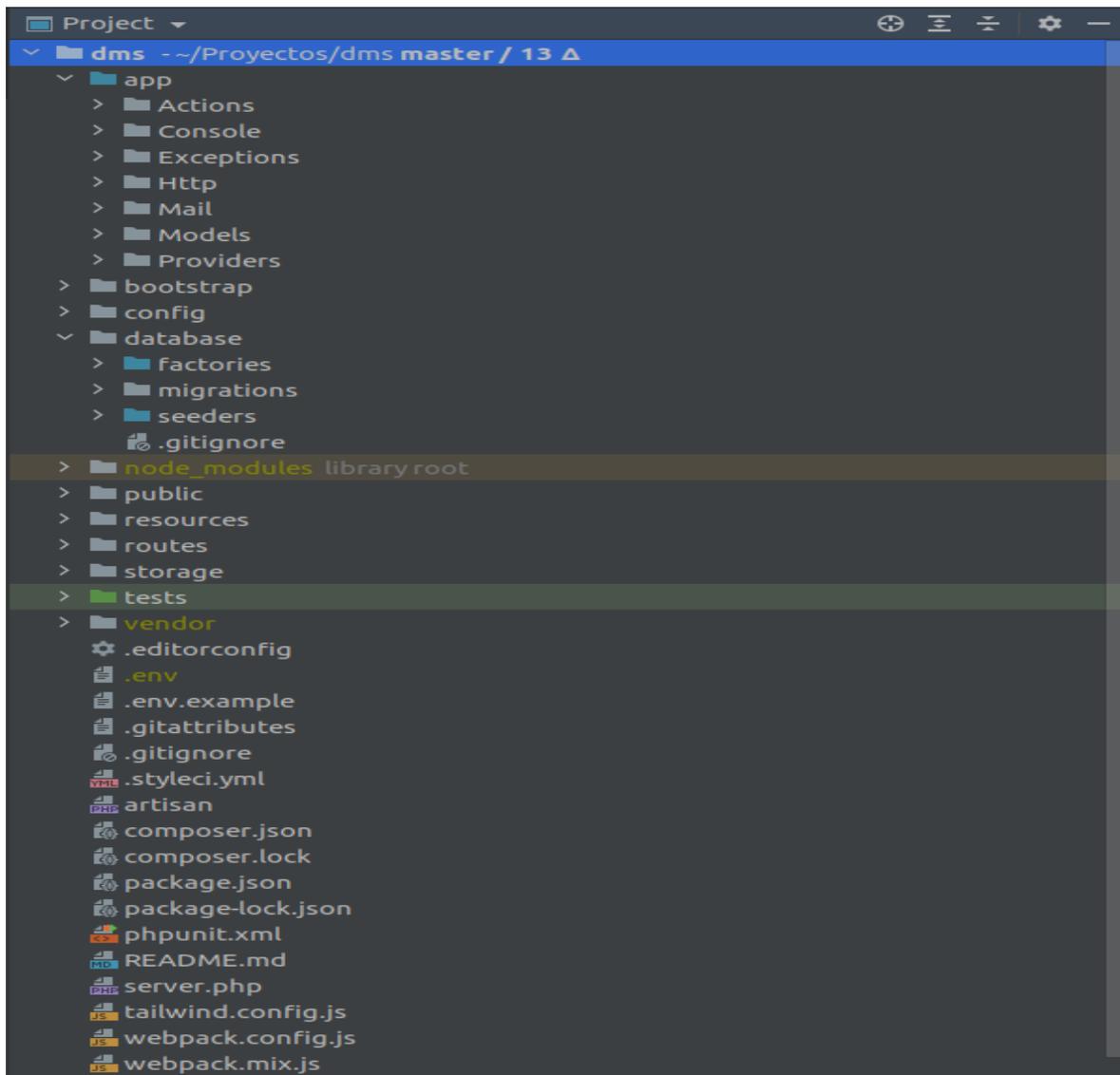


Ilustración 15 Estructura de carpetas del proyecto

Fuente: Elaboración propia

El proyecto utiliza el patrón “front controller” el cual obliga a que toda petición inicie en el archivo index.php de la carpeta public y según la url solicitada realizará la petición al controlador. El flujo se puede observar en la ilustración 16.

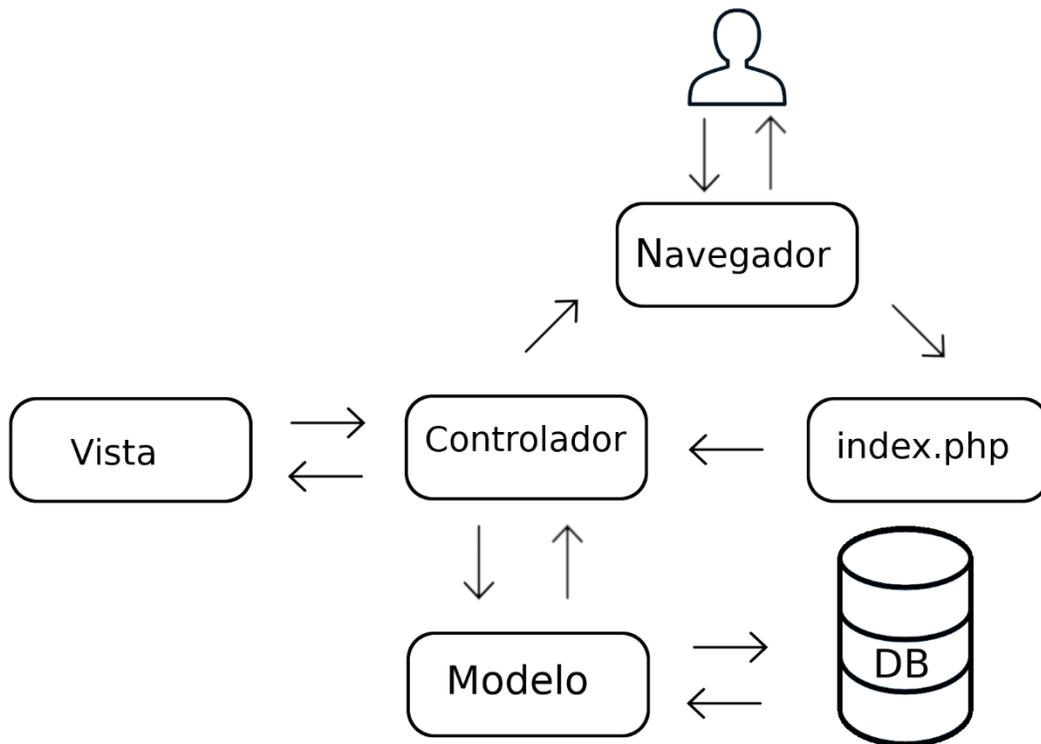


Ilustración 16 Modelo - vista - controlador

Fuente: Elaboración propia

Framework Vue JS

Después de instalar laravel es necesario instalar las dependencias de vuejs que se encuentran en la carpeta node_modules para esto se ejecuta el comando “npm install”. Esto descargará desde internet las librerías necesarias. Vue permite “hot reload” esto permite transpilar el código rápidamente mientras se está desarrollando y se realizan cambios pero una vez el software pase a producción se lo realiza con el comando “npm run production”. Realizar este paso se reduce el tamaño de los archivos de javascript y

de css para la descarga desde el servidor por parte del usuario. Para el desarrollo de la aplicación se usaron las siguientes dependencias con sus respectivas versiones:

Tabla 26: Tabla de dependencias vuejs

LIBRERÍA	VERSION
inertiajs/inertia-vue3	0.3.5
inertiajs/progress	0.2.4
tailwindcss/forms	0.2.1
tailwindcss/typography	0.3.0
vue/compiler-sfc	3.0.5
Axios	0.2.1
Laravel-mix	6.0.6
Lodash	4.17.19
Postcss	8.1.14
Postcss-import	12.0.1
Tailwindcss	2.0.1
Vue	3.0.5
Vue-loader	16.1.2

Fuente: Elaboración propia

El framework vue permitió tener componentes en cada vista de tal forma que se puede programar cada componente una vez y reutilizarlo en cualquier vista. Los componentes constan en su estructura de tres secciones como se observa en la ilustración 15 : una sección template que contiene etiquetas HTML, una sección script que contiene la lógica para experiencia de usuario y permite el uso de servicios, finalmente la última sección es para colocar estilos específicos para ese componente en caso de necesitarlo.

```
TheUpload.vue x
1  <template>
2    <div...>
115 </template>
116
117 <script>
118   import utc from 'dayjs/plugin/utc'
119
120   export default {name: "TheUpload"..}
309 </script>
310
311 <style scoped> Cuadros, 2/6/21 02:57 • modulo upload
312
313 </style>
314
```

Ilustración 17 Estructura de un archivo en vuejs

Fuente: Elaboración propia

Pruebas unitarias

Previo a subir el código fuente al repositorio en git se realizaron pruebas unitarias detalladas en la ilustración 18 las cuales aseguran que los cambios realizados no provoquen errores en el código. Las pruebas unitarias es parte de la nueva metodología de desarrollo guiada por pruebas. En esta metodología se crean primero las pruebas del código a desarrollar esperando como resultado inicial un error puesto que el código no existe, una vez se desarrolle el código según las especificaciones del requerimiento las pruebas pasan a estar aprobadas para posteriormente re factorizar el código en caso de ser necesario. Esta metodología permite asegurarnos que cualquier cambio realizado en el código probado o nuevas características desarrolladas no provoque errores.

```
WARN Tests\Feature\EmailVerificationTest
- email verification screen can be rendered →Email verification not enabled.
- email can be verified →Email verification not enabled.
- email can not verified with invalid hash →Email verification not enabled.

PASS Tests\Feature\ExampleTest
✓ basic test

PASS Tests\Feature>PasswordConfirmationTest
✓ password can be confirmed
✓ password is not confirmed with invalid password

PASS Tests\Feature>PasswordResetTest
✓ reset password link screen can be rendered
✓ reset password link can be requested
✓ reset password screen can be rendered
✓ password can be reset with valid token

PASS Tests\Feature\ProfileInformationTest
✓ profile information can be updated

PASS Tests\Feature\RegistrationTest
✓ registration screen can be rendered
✓ new users can register

PASS Tests\Feature\TwoFactorAuthenticationSettingsTest
✓ two factor authentication can be enabled
✓ recovery codes can be regenerated
✓ two factor authentication can be disabled

PASS Tests\Feature\UpdatePasswordTest
✓ password can be updated
✓ current password must be correct
✓ new passwords must match

Tests: 6 skipped, 23 passed
Time: 5.11s
```

Ilustración 18 Pruebas unitarias

Fuente: Elaboración propia

Despliegue

Para el despliegue de la aplicación se utilizó un servidor de prueba en la nube contratado con el servicio de upcloud, en este servidor se instaló el sistema operativo Linux Ubuntu versión 20.04. Se creó un usuario para evitar el uso del superusuario root y así minimizar el riesgo de ingreso de un atacante como privilegios. El acceso remoto se realiza mediante el protocolo ssh en el cual se aplicaron técnicas de hardening como son:

- Eliminar el acceso como usuario root (PermitRootLogin no).
- Se limita el número máximo de intentos de autenticación para una sesión (MaxAuthTries 3).
- Se estableció un periodo de gracia de inicio de sesión reducido (LoginGraceTime 20).
- Se deshabilitaron los accesos por contraseña (PasswordAuthentication no).
- Se deshabilitó el acceso con contraseñas vacías (PermitEmptyPasswords no).
- Se restringió el acceso únicamente por medio de la clave pública de ssh, para reducir el riesgo de posibles ataques, esto se realizó deshabilitando las otras opciones por defecto (ChallengeResponseAuthentication no, KerberosAuthentication no, GSSAPIAuthentication no).
- Se deshabilitó el inicio de aplicaciones gráficas remotas (X11Forwarding no).
- Se deshabilitó el banner textual SSH para evitar dar información a un posible atacante (DebianBanner no).

Se instaló y configuró el servicio firewall limitando el acceso únicamente a los siguientes puertos:

- 143, 993 (puertos imap)
- 110, 995 (puertos pop3)
- 25, 465, 587 (smtp)
- 53 (dns)
- 80, 443 (web)
- 22(ssh)

Como parte de las medidas de seguridad de acceso al servidor se habilitó el servicio de FAIL2Ban limitando a máximo 3 intentos de conexión al servicio ssh, mysqld. En caso de

intentos de ataque de fuerza bruta la ip del atacante queda registrada y estará inhabilitada durante 864000 segundos (10 días).

Conclusiones

Cualquier proceso que no se mide no se puede controlar y es de vital importancia un control permanente, en este caso el del cumplimiento de los documentos solicitados en cada proceso. Más importante aún es que el sistema de control sea ágil y oportuno, que muestre resultados en el momento solicitado y esté actualizado. En el caso de los sistemas de control documental también deben estar de acuerdo a la filosofía de la institución, es este caso manteniendo tecnología de punta que optimice el tiempo del representante del proceso (docente).

El software posee algunas pruebas unitarias automáticas que tratan de abarcar algunos de los procesos más importantes. Tiene la posibilidad de extender estas pruebas, incluso de crear pruebas funcionales para evitar errores cuando se necesite ampliar el desarrollo sin la necesidad de repetir pruebas con un equipo de testing sobre lo ya probado.

Se identificaron los tres procesos principales en donde se requiere un control de documentos. Estos son:

- Proceso de vinculación con la sociedad
- Proceso de tesis
- Procesos de prácticas de pregrado

Para cumplir con el segundo objetivo se construyó un software usando frameworks que permiten ser el punto de inicio para todo proceso que requiera el almacenamiento de documentos. Es por este motivo que se ha usado múltiples patrones de diseño, metodología y frameworks que manejan un estándar para darle continuidad a cualquier proyecto y sobre todo el trabajo en equipo. Los frameworks usados fueron:

- Laravel que es un framework con una curva de aprendizaje relativamente baja, se adapta al patrón de diseño de arquitectura MVC (modelo vista controlador), posee una amplia documentación construida a lo largo de sus 10 años de existencia, permite la modularidad y posee una amplia gama de paquetes y drivers que permiten la extensibilidad de forma fácil y segura. Permite la interacción con muchas bases de datos como es MySQL o postgresSQL incluso MsSQL por medio

de sus drivers privados, esto es posible gracias a la implementación de su ORM basado en el patrón active record.

- En el caso del framework vue tiene una curva de aprendizaje mucho más sencilla que otros frameworks javascript. Es posible construir código bien estructurado. Es extremadamente ligero y flexible. Brinda la posibilidad de extender sus funciones. Los templates que usa son declarativos lo cual lo hace extremadamente amigable para cualquier desarrollador con conocimientos en HTML. Su implementación del DOM virtual tiene un rendimiento muy alto. La flexibilidad de usar typescript o vainilla javascript da la facilidad para que el desarrollador escriba código rápidamente.

Se realizaron pruebas de aceptación detalladas en la anexo 1. Para esto se instala el software en un servidor de prueba en donde el usuario final realiza pruebas del sistema obteniendo aceptación del mismo.

Recomendaciones

Considerando que el software desarrollado puede ser aplicado a todo proceso que comparta la misma funcionalidad, se recomienda verificarlos e incluirlos en el desarrollo propuesto.

Instalarlo en un servidor propio de la institución para poder tener control y seguridad sobre los documentos que están alojados.

Se recomienda realizar copias de seguridad de la carpeta storage del software puesto que es en esta carpeta en donde está todo el conocimiento creado por los usuarios del software.

Se recomienda capacitar al personal sobre el uso correcto de la herramienta y sus beneficios para una transición más sencilla.

Bibliografía

- Altexsoft. (2016). *Agile Project Management: Best Practices and Methodologies*.
Obtenido de <https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>
- Apache. (2021). *Apache.org*. Obtenido de <https://httpd.apache.org/docs/current/>
- Axios-http. (2001). *Axios-http.com*. Obtenido de <https://axios-http.com/docs/intro>
- Ayaz, A., & Yanartaş, M. (2020). An analysis on the unified theory of acceptance and use of technology theory (UTAUT): Acceptance of electronic document management system (EDMS). *Computers in Human Behavior Reports*.
- Bahit, E. (2012). *Scrum y extreme programming para programadores*. Buenos aires.
- Chaudhary, M., & Kumar, A. (2014). *PhpStorm Cookbook*. 254: Packt Publishing Ltd.
- Composer. (2021). *Getcomposer.org*. Obtenido de <https://getcomposer.org/doc/00-intro.md>
- Del Sole, A. (2017). *Visual Studio Code Succinctly*. CreateSpace Independent Publishing Platform.
- DOURISH, P., EDWARDS, W., LAMARCA, A., LAMPING, J., PETERSEN, K., SALISBURY, M., . . . THORNTON, J. (2000). Extending Document Management. *ACM Transactions on Information Systems*, 140-170.
- Dyer, R. (2015). *Learning MySQL and MariaDB*. O'Reilly Media. Obtenido de <https://mariadb.com/kb/en/books/>
- Eslint. (2021). *Eslint.org*. Obtenido de <https://eslint.org/docs/user-guide/getting-started>
- Git scm. (15 de 03 de 2021). *Index git scm*. Obtenido de <https://git-scm.com/docs/git>
- Gitlab. (2021). *gitlab.com*. Obtenido de <https://about.gitlab.com/what-is-gitlab/>
- Hammond , S., & Umphress, D. (2012). Test Driven Development: The State of the Practice. *Proceedings of the Annual Southeast Conference*.
- Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press.
- Html. (2021). *Html.com*. Obtenido de https://html.com/#What_is_HTML
- Kanbanize. (2021). *kanbanize.com*. Obtenido de <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>
- Kao, C., & Liu , S. (2013). Development of a Document Management System for Private. *Procedia - Social and Behavioral*, 424-429.

- KHORIKOV, V. (2020). *Unit Testing: Principles, Practices, and Patterns*. Shelter Island: Manning Publications.
- KOSKELA, L. (2008). *Test Driven PRACTICAL TDD AND ACCEPTANCE TDD*. Greenwich: Manning Publications.
- KYOCERA Document Solutions. (2021). *KYOCERA Document Solutions*. Obtenido de <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/paperless/importancia-del-control-documental-para-el-cumplimiento-de-las-normas-de-calidad.html>
- Laravel. (2021). *Laravel.com*. Obtenido de <https://laravel.com/docs/5.5/eloquent>
- Lawall, J., Laurie, B., Hansen, R., Palix, N., & Muller, G. (2010). Finding Error Handling Bugs in OpenSSL using Coccinelle. *European Dependable Computing Conference*.
- Mangukiya, A. (2020). *Laravel: A framework for building PHP Apps*. Bournemouth University.
- Mojtaba, S., & Muhammad, A. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*.
- Npmjs. (2021). *Npmjs.com*. Obtenido de <https://docs.npmjs.com/about-npm#use-npm-to--->
- Prettier. (2021). *Prettier.io*. Obtenido de <https://prettier.io/docs/en/index.html>
- Pšenák, P., & Tibensky, M. (2020). The usage of Vue JS framework for web application creation. *Mesterséges intelligencia*, 61-72.
- Pustokhina, I., Rokhayati Rosa, A., Lydia, E., Shankar, K., & Huda, M. (2019). Concept of Electronic Document Management System (EDMS) as an Efficient Tool for Storing Document. *Journal of Critical Reviews*.
- Rappin, N. (2021). *Modern CSS with Tailwind Flexible Styling Without the Fuss*. Raleigh: The Pragmatic Bookshelf.
- Roebuck, K. (2011). *Object-Relational Mapping (Orm) : High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Samford: Tebbo.
- Sachdeva, S. (2016). Scrum Methodology. *international journal of engineering and computer science*.
- Sathiadas, J., & Wikramanayake, G. (2003). Document Management Techniques & Technologies. *Conference: 5th International Information Technology Conference*.

- Shore , J., & Warden, S. (2008). *The Art of Agile Development*. California: O'Reilly Media.
- Sironi, G. (2 de 12 de 2009). *giorgiosironi.com*. Obtenido de <https://www.giorgiosironi.com/2009/12/practical-php-testing-is-here.html>
- Symfony. (2021). *Symfony.com*. Obtenido de https://symfony.com/legacy/doc/book/1_0/en/16-Application-Management-Tools
- Symfony. (2021). *Symfony.com*. Obtenido de <https://symfony.com/projects/phpdocumentor>
- Symfony. (2021). *Symfony.com*. Obtenido de <https://symfony.com/>
- Thomson, L., & Welling, L. (2016). *PHP and MySQL Web Development*. Addison-Wesley Professional; 5th edition.
- Villasís-Keever,, M., & Miranda-Novales, M. (2016). El protocolo de investigación IV: las variables de estudio. *Alerg Mex*, 303-310.
- YOUNG, A., & HARTER, M. (2015). *Node.js in Practice*. Shelter Island: Manning Publications.

Anexos

Anexo 1

Samborondón, 16 de mayo 2021

Aceptación de software

Por la presente se hace constar que el señor **Juan José Flores Cuadros**, con número de cédula de ciudadanía **0920183126**, terminó satisfactoriamente el desarrollo del proyecto: **"SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE DOCUMENTOS ACADÉMICOS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD ECOTEC"**, cumpliendo con los requerimientos solicitados en cuanto a funcionalidad.

El detalle de la funcionalidad requerida y aceptada es el siguiente:

Funcionalidad	Conforme
Crear, ver, editar y buscar usuario	x
Cambio de contraseña, recuperación de contraseña	x
Crear, ver, editar y buscar proceso rol docente	x
Acceso a procesos propios rol usuario	x
Notificación por correo de inclusión en proceso	x
Crear, ver, editar requerimiento de documentos con restricciones de extensión y tiempo límite	x
Subir documentos bajo reglas preestablecidas en el requerimiento y funcionamiento de restricciones	x
Reporte de cumplimiento de requerimientos	x
Descarga de requerimientos	x
Crear, editar, descargar y buscar formatos y plantillas	x
Versionado de formatos y plantillas	x
Mostrar plantillas públicas dinámicamente y permitir descarga.	x

Atentamente.


Mgtr. Erika Ascencio Jordán
Decana de la Facultad de Ingenierías
COORDINADORA ACADÉMICA
DE LA FACULTAD DE INGENIERÍAS