

**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y
GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE
TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS
DE LA ESFOT**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

ANIBAL ANDRÉS TAIPE GUAMÁN

anibal.taipe@epn.edu.ec

DIRECTOR: Ing. Luis Aníbal Almeida Molina, MSc

luis.almeidal@epn.edu.ec

CODIRECTOR: PhD. María Gabriela Pérez Hernández

maria.perez@epn.edu.ec

Quito, Octubre 2019

Declaración

Yo Anibal Andrés Taipe Guamán declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la ley de propiedad intelectual, por su reglamento y por la normatividad institucional vigente.

Anibal Andrés Taipe Guamán

Certificación

Certifico que el presente trabajo fue desarrollado por el señor Anibal Andrés Taipe Guamán, bajo nuestra supervisión.

Ing. Luis Aníbal Almeida Molina, MSc
DIRECTOR DE PROYECTO

PhD. María Gabriela Pérez Hernández
CODIRECTORA DE PROYECTO

Agradecimiento

En primer lugar, agradezco a Jehová Dios por darme la vida, las fuerzas y la inteligencia para realizar todas las actividades que me he propuesto, ya que sin su guía y ayuda no podría haber alcanzado ningún logro.

En especial quiero dar las gracias a mis padres Anibal y Leticia que fueron los que siempre han estado apoyándome en sentido físico, económico, emocional y sobre todo espiritual. Agradezco su paciencia, por estar pendiente de mi, por quererme tanto y apoyarme en todas las decisiones que he tomado, por sus consejos y amor que siempre me han mostrado.

También quiero dar las gracias por tener tres hermanos Daniel, Evelyn y Raquel que nunca me dejaron solo y que me estuvieron animando para que logre cumplir mis objetivos y pueda seguir adelante con mis metas.

Quiero agradecer también al Ingeniero Luis Almeida por su tiempo, paciencia y todo el apoyo que me brindó en la realización de este proyecto, que a pesar de todas sus actividades siempre estuvo dispuesto a guiarme y darme las sugerencias que me serán útiles a lo largo de mi carrera profesional.

Finalmente quiero agradecer a la Escuela de Formación de Tecnólogos y a los pocos profesores que de verdad se desvivían por enseñarnos y buscaban los mejores métodos para que podamos aprender, a esos profesores que, aunque la mayoría los consideraba estrictos, difíciles, pero fue con los que más aprendí y lograron ayudarme en mi formación como un profesional.

Dedicatoria

Este proyecto va dedicado a mi madre Leticia que es la que ha creído en mi, en mis capacidades y estuvo cuidándome siempre en todos estos años que me dediqué a la universidad y en las otras metas que me propuse.

A mi padre Anibal que estuvo para ayudarme económicamente, con sus consejos con sus apoyo moral y espiritual permitieron que pueda mantenerme firme hasta culminar mis estudios.

En especial va dedicado a esa persona especial que estuvo en los últimos años de mi carrera que creyó en mí y no permitió que me de por vencido en la realización de este proyecto. Aunque ya no está conmigo este logro va dedicado especialmente para ti.

ÍNDICE DE CONTENIDO

1.	INTRODUCCIÓN.....	1
1.1	<i>Planteamiento del problema</i>	1
1.2	<i>Objetivo general</i>	2
1.3	<i>Objetivos específicos</i>	3
1.4	<i>Alcance</i>	3
1.5	<i>Estructura del documento</i>	4
2.	METODOLOGÍA.....	5
2.1	<i>Metodología de desarrollo ágil Scrum</i>	5
	Componentes de Scrum	5
	Elementos de Scrum.....	7
2.2	<i>Diseño de la base de datos y prototipos de interfaces</i>	7
	PostgreSQL	7
	Moqups	8
2.3	<i>Herramientas de desarrollo web</i>	8
	<i>Hyper Text Markup Languaje (HTML)</i>	8
	<i>Cascading Style Sheets (CSS)</i>	8
	Sass	8
	Bootstrap.....	9
	Angular.....	9
	Node.js.....	9
2.4	<i>Recursos estimados</i>	9
	Costos de <i>hardware</i>	10
	Costos de <i>software</i> para desarrollo	10
	Costos de licencia para el desarrollo	10
	Costos de desarrollo	10
3.	RESULTADOS Y DISCUSIÓN	11
3.1	<i>Requerimientos generales del sistema web.</i>	11
	Interfaces de usuario para el sistema web	11
	Interfaces de software para el sistema web.....	11
	Interfaces de comunicación para el sistema web	12
	Restricciones para el sistema web	13
	Funciones del sistema web.....	13
	Descripción de roles del sistema	13
	Características del usuario	14

3.2	<i>Requerimientos específicos del sistema web.</i>	14
	Funcionalidades del sistema web.	14
	Identificación y autenticación de usuarios	14
	Administración de facultades	15
	Administración de carreras	15
	Administración de usuarios	15
	Administración de planes de titulación	15
	Administración de estudiantes	15
	Creación de convocatorias	15
	Creación de categorías e ítems para el acta de reunión de la comisión	15
	Evaluación de los planes de titulación	16
	Finalización de la reunión	16
3.3	<i>Conformación del equipo Scrum.</i>	16
3.4	<i>Actividades.</i>	16
	Historias de usuario	16
	<i>Product backlog</i>	17
	<i>Sprints</i>	18
3.5	<i>Creación de la base de datos.</i>	20
3.6	<i>Creación de la página web.</i>	23
3.7	<i>Desarrollo de la aplicación web para la gestión de los planes de unidad de titulación.</i>	24
	Conexión del <i>backend</i> con la base de datos	24
	Conexión del <i>frontend</i> con las APIs del backend	25
	Creación del módulo de autenticación	27
	Creación del módulo de restaurar contraseña	28
	Creación del módulo de administración	29
	Creación del módulo planes	32
	Creación del módulo convocatoria	34
	Creación del módulo estudiantes	35
	Creación del módulo comisión	36
	Creación del módulo perfil de usuario	42
	Creación del <i>dashboard</i>	43
3.8	<i>Pruebas de software del sistema web.</i>	44
	Pruebas de compatibilidad	44
	Pruebas de seguridad	45
	Pruebas unitarias	46
	Pruebas de carga	47

Pruebas de calidad.	49
Pruebas de usabilidad.	50
Preguntas SUS.	50
4. CONCLUSIONES Y RECOMENDACIONES.....	51
4.1 Conclusiones	51
4.2 Recomendaciones	52
5. REFERENCIAS BIBLIOGRÁFICAS	53
6. ANEXOS:.....	55
6.1 Anexo I: Historias de usuario.....	55
6.2 Anexo II: Prototipos de Interfaz.....	63
6.3 Anexo III: Manual de usuario	66

ÍNDICE DE FIGURAS

Figura 1 Ciclo de desarrollo ágil	5
Figura 2 Diagrama de acceso a módulos generales del sistema	12
Figura 3 Diagrama de comunicación entre tecnologías	12
Figura 4 Panel de administración ZenHub	19
Figura 5 Diseño físico de la base de datos.....	21
Figura 6 Consola de PostgreSQL.....	22
Figura 7 Creación de base de datos según configuración regional.....	22
Figura 8 Código de ingreso a la base de datos creada	23
Figura 9 Creación del schema epn.....	23
Figura 10 Comandos de instalación y creación de proyecto en Angular.....	23
Figura 11 Configuración ambientes de desarrollo backend	24
Figura 12 Pool de conexión con la base de datos	24
Figura 13 Configuración ambientes de desarrollo frontend	25
Figura 14 Servicios para conexión con el backend	26
Figura 15 Ejemplo de conexión con API REST en backend	26
Figura 16 Rutas de comunicación con el backend	27
Figura 17 Inicio de Sesión	28
Figura 18 Correo con las credenciales de acceso.....	28
Figura 19 Restaurar contraseña	29
Figura 20 Menú administración	29
Figura 21 Lista de carreras	30
Figura 22 Formulario para crear nueva carrera	30
Figura 23 Lista de usuarios	31
Figura 24 Formulario para crear nuevo usuario	31
Figura 25 Menú planes	32
Figura 26 Formulario con datos del plan a crear	33
Figura 27 Formulario de creación de estudiantes	33
Figura 28 Resumen del plan a crear	33
Figura 29 Lista de planes ingresados.....	34
Figura 30 Menú convocatoria	34
Figura 31 Lista de convocatorias activas.....	35
Figura 32 Lista de planes para revisar	35
Figura 33 Detalles de convocatorias activas	35
Figura 34 Lista de estudiantes.....	36

Figura 35 Menú estudiantes	36
Figura 36 Menú comisión	37
Figura 37 Acta de reunión para editar	37
Figura 38 Administración del acta de reunión	38
Figura 39 Lista de convocatorias y actas a revisar.....	38
Figura 40 Formulario de observaciones de cada plan.....	39
Figura 41 Acta de reunión y calificación del plan	39
Figura 42 Acta de reunión final de la comisión.....	40
Figura 43 Ventana de impresión del acta de la reunión	40
Figura 44 Historial de planes en módulo comisión	41
Figura 45 Lista de estudiantes en modulo comisión.....	41
Figura 46 Lista de estudiantes en modulo comisión.....	41
Figura 47 Botón editar perfil	42
Figura 48 Formulario de edición del perfil	42
Figura 49 Menú dashboard.....	43
Figura 50 Dashboard	43
Figura 51 Pruebas en el navegador safari.....	45
Figura 51 Pruebas en el navegador Firefox	45
Figura 53 Pruebas de seguridad en Codacy	46
Figura 54 Pruebas unitarias en Codacy	47
Figura 55 Jmeter grupo de pruebas	48
Figura 56 Jmeter http request.....	48
Figura 57 Jmeter resultados de la prueba	49
Figura 58 Auditando la aplicación web	49
Figura 59 Resultado de las pruebas de calidad	50
Figura 60 Prototipo iniciar sesión	63
Figura 61 Prototipo restaurar contraseña	64
Figura 62 Prototipo menú administración	64
Figura 63 Prototipo tablas.....	65
Figura 64 Prototipo formularios	65

ÍNDICE DE TABLAS

Tabla 1 Costo estimado del desarrollo de la aplicación web.....	10
Tabla 2 Roles de usuario en el sistema.....	13
Tabla 3 Equipo de Trabajo y Roles	16
Tabla 4 Modelo de historia de usuario.....	17
Tabla 5 Product backlog	17
Tabla 6 Sprints.....	18
Tabla 7 Bases de datos según ambientes de desarrollo.....	21
Tabla 8 Cantidad permitida de usuarios a crear.....	32
Tabla 9 Estados de los planes de titulación	34
Tabla 10 Resultados de las pruebas de compatibilidad.	45
Tabla 11 Resultados de las pruebas unitarias.....	47
Tabla 12 Matriz de evaluación de resultados SUS.....	51

Resumen

El presente proyecto de titulación tiene como objetivo desarrollar e implementar una aplicación web, que permita gestionar la evaluación de los planes de titulación de la Escuela de Formación de Tecnólogos (ESFOT). La aplicación web permite una administración digital de los planes de titulación presentados por los estudiantes, requisito necesario previo a la obtención de su título profesional.

La aplicación web esta desarrollada con tres tecnologías principales, para la interacción con los usuarios se utiliza Angular, para el acceso a los datos y la lógica de programación se usa Node.js y los datos se almacenan en PostgreSQL. El uso de estas tecnologías se deriva del avance exponencial de JavaScript como lenguaje de programación en los últimos años.

El proceso de desarrollo de este proyecto se encuentra detallado en este documento, para lo cual fue necesario apoyarse en una metodología de desarrollo ágil de software *Scrum*. La metodología *Scrum* nos ayuda a aplicar un conjunto de buenas prácticas, a trabajar en equipo para obtener los mejores resultados en el desarrollo del proyecto.

El resultado final es una aplicación web que permite la administración de usuarios, estudiantes y planes de titulación. La evaluación de los planes de titulación, la generación de actas de las reuniones realizadas por la comisión de evaluación se puede obtener automáticamente, además de la digitalización de la información hace de esta aplicación web una herramienta adicional para usarla dentro de la ESFOT e implementarla en otras facultades.

Teniendo en cuenta que la aplicación web usa un lenguaje de programación actual, una metodología de desarrollo ágil y puede ser implementada en otras facultades, sería de gran utilidad mediante otros proyectos de titulación implementar en el sistema, módulos adicionales que abarque todo el ciclo de vida de los proyectos de titulación.

Palabras claves: aplicación web, framework, planes, titulación, cliente, servidor.

Abstract

The purpose of this qualification project is to develop and implement a web application, which allows to manage the evaluation of the qualification plans of the Escuela de Formación de Tecnólogos (ESFOT). The web application allows a digital administration of the qualification plans presented by the students, necessary requirement prior to obtaining their professional degree.

The web application is developed with three main technologies, for interaction with users Angular is used, for data access and programming logic Node.js is used and the data is stored in PostgreSQL. The use of these technologies' stems from the exponential advancement of JavaScript as a programming language in recent years.

The process of developing this project is detailed in this document, for which it was necessary to rely on an agile Scrum software development methodology. The Scrum methodology helps us to apply a set of good practices, to work as a team to obtain the best results in the development of the project.

The end result is a web application that allows the administration of users, students and degree plans. The evaluation of the qualification plans, the generation of minutes of the meetings held by the evaluation committee can be obtained automatically, in addition to the digitization of the information makes this web application an additional tool to use it ESFOT and implement it in other faculties.

Considering that the web application uses a current programming language, an agile development methodology and can be implemented in other faculties, it would be very useful through other titling projects implemented in the system, modules covering the entire life cycle of the degree projects.

Keywords: web application, framework, plans, qualification, client, server.

1. INTRODUCCIÓN

1.1 Planteamiento del problema

Uno de los requerimientos que los estudiantes de la ESFOT deben desarrollar como requerimiento para obtener su título universitario es el plan de trabajo de titulación. Cuando el director del proyecto de titulación ha revisado y aprobado el plan, el estudiante lo presenta a la subdirección, para que la Comisión de la Unidad de Titulación de la carrera a la que pertenece el estudiante lo revise y lo apruebe (Comisión Unidad de Titulación ESFOT, 2017).

La entrega deberá realizarse en formato digital y en físico a la secretaría de la subdirección (Comisión Unidad de Titulación ESFOT, 2017) El personal de secretaría enviará un correo electrónico con los archivos adjuntos correspondientes a la Comisión de la Unidad de Titulación para una revisión individual previa del plan de trabajo.

La comisión conformada por el subdirector y tres docentes deberán volver a revisar los planes, esta revisión es conjunta, uno de los miembros tomará nota de las observaciones de cada uno de los miembros y las registrará en el Acta de Observaciones, que es una plantilla en Microsoft Word. La mayoría de los planes de trabajo de titulación presentados tiene observaciones, y en muy pocas ocasiones es aprobado en la primera revisión.

En la revisión conjunta de los planes, los miembros hacen referencia a sus observaciones individuales, anotadas en las copias físicas y digitales del plan, de esta manera los otros miembros de la comisión no pueden identificar con facilidad en que parte del documento se hace referencia. Adicionalmente, el miembro encargado de realizar las anotaciones debe prestar mucha atención a cada uno de los comentarios y observaciones de los miembros para poder registrar cada una de ellas. Esto hace que el miembro encargado de registrar las observaciones no participe y aporte continuamente con sus propias observaciones y algunas de ellas se pasen por alto.

Las Actas de Aprobación y de Observaciones se llenan a mano con información que se repite en cada una de las ellas. No existen campos en los cuales se pueda realizar recomendaciones dirigidas al director del proyecto.

Una vez que la comisión ha revisado los planes, el personal de secretaría de la subdirección es el encargado de entregar las actas al estudiante y transmitir de forma verbal ciertas indicaciones realizadas por la comisión. Actualmente, este proceso de revisión y aprobación de los Planes de Trabajo de Titulación conlleva mucho esfuerzo y tiempo en la forma que se realiza.

Por otro lado, existe información técnica que la comisión le transmite al personal administrativo, para que estos lo transmitan en forma verbal a los estudiantes que presentaron su plan. Esta información en algunos casos llega incompleta, debido a que la mayoría de los estudiantes regresan por las observaciones de su plan, pasando varias semanas de su revisión.

Los estudiantes deben acudir a la secretaría de la Subdirección a solicitar los resultados de la presentación de sus planes, en caso de que han sido revisados los planes, el personal de secretaria imprime los documentos sobre los cuales se han realizado las observaciones, el estudiante debe sacar copias de ellos y solicitar copias de los documentos digitales. Si su plan no ha sido revisado el estudiante deberá regresar nuevamente en otra ocasión, generando en ellos cierto grado de frustración por no obtener información oportuna. Esto produce un gasto innecesario de dinero, tiempo y papel.

Existen muchos planes que se presentan a la comisión con las correcciones sobre las observaciones realizadas por la comisión anteriormente, algunos de ellos pasados algunos meses. Resulta complicado recordar con exactitud para los miembros y personal de secretaría las observaciones por parte de la comisión, y recuperar los documentos sobre los que se trabajó anteriormente para realizar la nueva revisión y aprobación de estos, ocasionando pérdida de tiempo adicional.

Por todo lo anteriormente expuesto, se propone desarrollar una aplicación web para la gestión de la información y generación de reportes relacionados con el proceso de revisión y aprobación de los planes de trabajo de titulación para todas las carreras de la Escuela de Formación de Tecnólogos.

1.2 Objetivo general

Desarrollar una aplicación web para la gestión y generación de reportes de los planes de trabajo de titulación revisados por las comisiones académicas de la ESFOT.

1.3 Objetivos específicos

- Determinar los requerimientos para la aplicación web.
- Diseñar la estructura del sistema.
- Diseñar la base de datos e interfaz de la aplicación.
- Desarrollar la aplicación web cumpliendo con los requerimientos establecidos.
- Implementar la aplicación web en un ordenador que funcione con la red de la Escuela Politécnica Nacional.

1.4 Alcance

Llevar a cabo el proceso logístico para la revisión y aprobación de los planes de trabajo de titulación implica mucho esfuerzo por parte de todos los implicados en el mismo. El proceso como tal es poco eficiente por el gran número de horas que se invierte para crear cada una de las versiones de la revisión.

Por ello, la aplicación web ayudará a la comisión en la evaluación de planes, para que no exista información separada o repetida, las observaciones realizadas por cada miembro de la comisión estará disponible para todos ellos. Se evitará doble trabajo en la realización de reportes, ya que automáticamente se cargará la información en las actas.

Además, los estudiantes ahorrarán tiempo y recursos económicos ya que los resultados de la revisión de sus planes de trabajo de titulación serán de forma más inmediata, puesto que la obtendrán en sus correos electrónicos adjunto con la información y observaciones de parte de la comisión, evitando el uso innecesario de papel.

Al personal de secretaría de la subdirección facilitará el almacenamiento y recuperación de los archivos generados en las revisiones de los planes. El envío de correos e información a los estudiantes y miembros de las comisiones se realizarán automáticamente, evitando la pérdida de tiempo en la digitación innecesaria de datos que pueden ser ingresados en una sola ocasión.

En cuanto a la seguridad, el sistema presentará un acceso controlado para el personal involucrado en el proceso. La aplicación estará basada en módulos, los cuales podrán ser

visualizados por los miembros de las comisiones y el personal de secretaría de acuerdo con el rol que posean, para que puedan obtener y manipular la información generada. El acceso a la aplicación se realizará mediante la red interna de la Escuela Politécnica Nacional (EPN).

1.5 Estructura del documento

En la sección 1, se describe el planteamiento del problema, objetivo general, los objetivos específicos y el alcance del proyecto.

En la sección 2, se detalla la metodología ágil de desarrollo Scrum y su implementación en el proyecto. Adicionalmente se definen los Sprint y las herramientas que se utilizaron para el desarrollo de la aplicación web.

En la sección 3, se especifica cómo se llevaron a cabo las tareas definidas en cada uno de los Sprint, también se presenta los resultados obtenidos al culminar cada una de las etapas de desarrollo.

Finalmente, en la sección 4 se presenta las respectivas conclusiones y recomendaciones obtenidas durante el desarrollo de todo el proyecto

2. METODOLOGÍA

2.1 Metodología de desarrollo ágil Scrum

Scrum es una metodología de desarrollo ágil que nos permite obtener un software en menos tiempo, de buena calidad y menos costes. Tiene como base la idea de creación de ciclos breves para el desarrollo que comúnmente se llaman iteraciones y que en Scrum se llamarán “Sprints” (Gallego, 2015). Un ciclo de desarrollo se divide en 5 fases como se observa en la figura 1.

- **Concepto:** Se define las características del producto de forma general.
- **Especulación:** Se desarrolla y revisa los requisitos generales manteniendo las funcionalidades que se esperan, además, se crea un plan de entrega.
- **Exploración:** Se añade las funcionalidades de la fase de especulación.
- **Revisión:** Se revisa lo que se ha construido y se contrasta con el objetivo deseado.
- **Cierre:** Se entregará en la fecha acordada una versión del software.



Figura 1 Ciclo de desarrollo ágil

Componentes de Scrum

Para la metodología Scrum es importante entender las fases del proceso de desarrollo y los roles que poseen los miembros del equipo (Scrum Study, 2017). Se puede dividir a Scrum en 3 fases principales, que se describen como reuniones (Gallego, 2015).

- **Planificación del *backlog***

Es un documento donde se definen los requisitos del sistema por prioridades. En esta reunión se obtendrá una lista de tareas que es conocido como el *Sprint Backlog* (Gallego, 2015).

- **Seguimiento del *sprint***

Se realizan reuniones diarias para evaluar el avance de las tareas, se realizan tres preguntas: ¿Qué trabajo se realizó desde la reunión anterior?, ¿Qué trabajo se hará hasta la nueva reunión?, ¿Qué inconvenientes hay que solucionar para poder continuar? (Gallego, 2015).

- **Revisión del *sprint***

Al finalizar el *sprint* se realiza una revisión del incremento que se ha generado y se presenta al cliente los resultados finales en una versión del software.

Los roles se dividen en 2 grupos, el primero son asignados a las personas que estarán completamente comprometidas con el proyecto y son requeridos de forma obligatoria (Scrum Study, 2017).

- ***Product owner***

Es la persona que toma las decisiones, conoce el negocio del cliente y la visión del producto. Se encarga de escribir las ideas del cliente y las establece por prioridades (Gallego, 2015).

- ***Scrum master***

Es la persona encargada en comprobar que el modelo y la metodología funcionen, cumple un rol de líder ayudando a eliminar los impedimentos que puedan tener durante el desarrollo del proyecto y se asegura de que estén siguiendo los procesos de Scrum. (Scrum Study, 2017).

- ***Development Team***

Está formado por un pequeño grupo de personas que tienen autoridad para organizar y tomar decisiones para cumplir su objetivo (Gallego, 2015). Son responsables de entender los requerimientos especificados por el *Product owner* para el desarrollo del producto (Scrum Study, 2017).

El segundo grupo de los roles en Scrum son de las personas que no son parte del proceso, pero, son necesarios para la retroalimentación al finalizar cada *sprint* (Gallego, 2015).

- **Usuarios**
Destinatario final del producto
- **Stakeholders**
Participan en las revisiones del *sprint*.
- **Managers**
Toman las decisiones finales participando en la selección de los objetivos y de los requisitos.

Elementos de Scrum

Los elementos que forman a Scrum se los divide en 3 grupos. Estos elementos ayudan a planificar y revisar los *sprint* que conforman nuestro proyecto (Bahit, 2012).

- **Product backlog**
Es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada. Estos requisitos son los que tendrá el software o los que irán adquiriendo en sucesivas iteraciones (Gallego, 2015).
- **Sprint backlog**
Es un subconjunto que nace de los elementos previamente seleccionados del *Product backlog*, una lista de tareas que se realizan en un *sprint* que producen incrementos de funcionalidad en el software (Bahit, 2012).
- **Incremento del producto**
Al finalizar cada *sprint* se evidencia un incremento en el producto, es una parte terminada y totalmente operativa para presentarlo al usuario final, de esta manera el cliente puede verificar la calidad de nuestro producto desde el inicio del desarrollo.

2.2 Diseño de la base de datos y prototipos de interfaces

PostgreSQL

Es un sistema de base de datos relacional de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación de confiabilidad, solidez de

funciones y rendimiento, además, de la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento. Se ejecuta en la mayoría de los sistemas operativos, por esta razón, es una de las bases de datos relacionales de código abierto elegida por personas y organizaciones (PostgreSQL, 2019).

Moqups

Es una aplicación web que nos permite crear *wireframes*, maquetas, diagramas y prototipos de aplicaciones web o móviles. Permittiéndonos tener una base para el desarrollo del frontend de nuestra aplicación. Podemos visualizar los prototipos en el Anexo 2.

2.3 Herramientas de desarrollo web

Hyper Text Markup Languaje (HTML)

Hyper Text Markup Languaje (HTML, por sus siglas en inglés), lenguaje de marcas de hipertexto, define la estructura básica del contenido de una página web. Este nos da la posibilidad de crear documentos que serán interpretados por los navegadores que convertirán ese texto con marcas en una página web. (Pérez & Altadiill, 2016)

Cascading Style Sheets (CSS)

CSS, hojas de estilo en cascada, lenguaje que sirve exclusivamente para trabajar la presentación de la información que esta en la estructura descrita por HTML. Las hojas de estilo tienen una estructura simple, flexible y potente, debido a estas características y su fácil estructuración es posible definir la apariencia de cada elemento o grupo con suma facilidad cambiando posteriormente su diseño si es necesario. (Orós, 2014)

Sass

Como encontramos descrito en su página web de Sass, es un CSS potente, una extensión de este lenguaje maduro y estable. Permite usar variables, reglas anidadas, combinaciones, funciones y más, todo con una sintaxis totalmente compatible con CSS. Sass ayuda a mantener las hojas de estilo grandes bien organizadas y facilita el compartir el diseño dentro y entre los proyectos. (Sass, 2019)

Bootstrap

Framework de código abierto que combina de forma integral HTML5, CSS Y JS, este nos permite diseñar las páginas utilizando un sistema de *grids* o rejillas, el cual divide a la pantalla en filas, donde cada fila puede dividirse en doce espacios de anchura relativa (Pérez & Altadill, 2016). Se caracteriza principalmente por ser adaptativo, lo que quiere decir que la información de una página web elaborada con este framework puede visualizarse con facilidad en cualquier tipo de dispositivo electrónico con capacidad de ejecutar un navegador.

Angular

Angular es un *framework* y una plataforma para la creación de aplicaciones cliente en HTML y está escrito en Typescript, implementa la funcionalidad central y opcional como un conjunto de bibliotecas de Typescript que importa a sus aplicaciones.

Node.js

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para construir aplicaciones en red escalables. Esto contrasta con el modelo de concurrencia más común hoy en día, donde se usan hilos del Sistema Operativo. Las operaciones de redes basadas en hilos son relativamente ineficientes y son muy difíciles de usar (NodeJs, 2019).

Además, los usuarios de Node.js están libres de preocupaciones sobre el bloqueo del proceso, ya que no existe. Casi ninguna función en Node.js realiza I/O directamente, así que el proceso nunca se bloquea. Debido a que no hay bloqueo es muy razonable desarrollar sistemas escalables en Node.js (NodeJs, 2019).

2.4 Recursos estimados

La aplicación web debe ser desarrollada por un analista de software, para esto, se va a utilizar herramientas de licencia libre como Angular, Node.js, PostgreSQL las cuales no

generarán un gasto en el desarrollo a nivel de *software*. El servidor fue proporcionado por la ESFOT, lo que tampoco representará un gasto a nivel de *hardware*. En la Tabla 1 se presenta el detalle de los principales costos del desarrollo, donde se incluyen los siguientes costos directos:

Costos de *hardware*

Hace referencia al equipo de computación requerido para desarrollar y presentar el aplicativo (Castillo, 2015). Estos equipos son proporcionados por la ESFOT.

Costos de *software* para desarrollo

Hace referencia al trabajo, creación de código y programación con herramientas de software licenciado (Castillo, 2015). El software utilizado es libre.

Costos de licencia para el desarrollo

Implica si el trabajo se realizó sobre una tecnología con licencia (Castillo, 2015). Se usan tecnologías bajo licenciamiento libre.

Costos de desarrollo

Es el tiempo invertido por el desarrollador en la investigación del conocimiento necesario para implementar el software, incluyendo todo el análisis respectivo y la captación de datos. Además, la implementación del software con sus respectivas pruebas (Castillo, 2015). Se toma como referencia para el costo de horas hombre en desarrollo, de la tabla de sueldos mínimos por sector presentados por el Ministerio de Trabajo del Ecuador (Ministerio del Trabajo del Ecuador, 2019).

Tabla 1 Costo estimado del desarrollo de la aplicación web

Recursos estimados			
Descripción	Costo por hora (\$)	Total horas	Costo Total (\$)
<i>Hardware</i>	0,00	0	0,00
<i>Software</i>	0,00	0	0,00
Licencias	0,00	0	0,00
Desarrollo (horas/hombre)	6,00	720	4320,00
TOTAL			4320,00

Elaborado por: Andrés Taipe

3. RESULTADOS Y DISCUSIÓN

3.1 Requerimientos generales del sistema web.

Interfaces de usuario para el sistema web

Los usuarios de la aplicación web interactuarán con el sistema por medio de un navegador web que se encuentra instalado en su estación de trabajo, el sistema web funciona correctamente con los siguientes navegadores:

- Google Chrome en su versión 54.0 y posteriores
- Firefox en su versión 54.0 y posteriores
- Safari en su versión 10 y posteriores

Interfaces de software para el sistema web

La interfaz de software brindará la información necesaria a los usuarios de acuerdo con el rol asignado. Como se detalla en la Figura 2, al administrador permitirá la gestión de las carreras y usuarios. El personal de secretaría podrá gestionar el ingreso de los planes de titulación, la creación de las convocatorias para la comisión y la administración de la información de cada estudiante. El subdirector y los miembros de la comisión podrán realizar la calificación de los planes de titulación. El director y subdirector al ser autoridades de la facultad tienen acceso a todos los módulos del sistema.

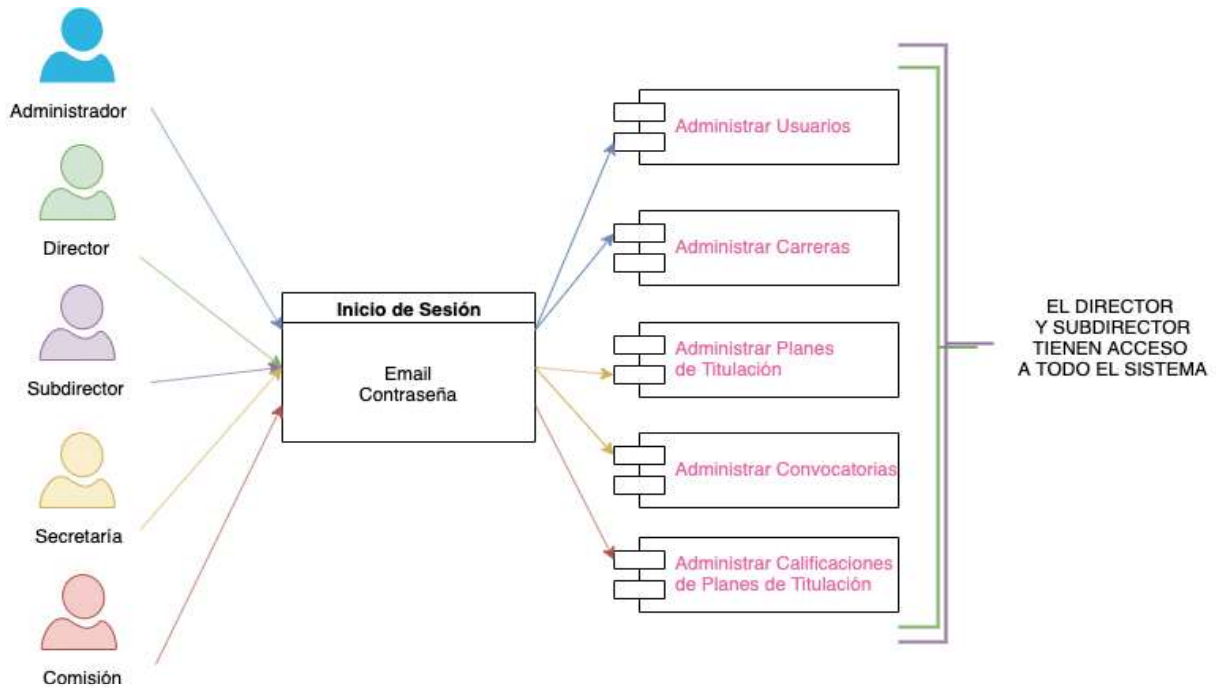


Figura 2 Diagrama de acceso a módulos generales del sistema

Interfaces de comunicación para el sistema web

El *frontend*, que es la tecnología de diseño y desarrollo web que se ejecuta en el navegador y que se encargan de la interactividad con los usuarios (Chapaval, 2017), desarrollado en Angular se comunica con el *backend*, que es la capa de acceso a datos de un software, que no es directamente accesible por los usuarios y contiene la lógica de la aplicación que maneja dichos datos (Chapaval, 2017), desarrollado en Node.js mediante API REST, que es una interfaz hecha para desarrolladores que expone nuestros recursos de una forma entendible y accesible para otros (Trinidad, 2015), y este a su vez obtiene los datos del gestor de base de datos PostgreSQL como lo apreciamos en la Figura 3.

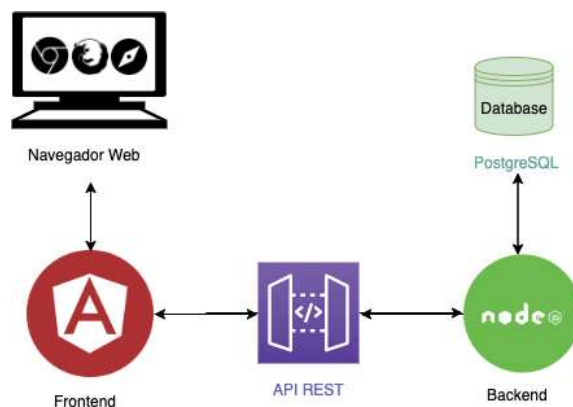


Figura 3 Diagrama de comunicación entre tecnologías

Restricciones para el sistema web

- Al ser un sistema web es necesario contar con un computador que tenga instalado un navegador web.
- El computador necesita conectarse a la red informática de la Escuela Politécnica Nacional.
- El acceso es controlado por el administrador, el cual crea los usuarios de acuerdo con el rol que sea asignado.
- Cada usuario debe ingresar obligatoriamente las credenciales recibidas en el correo electrónico para hacer uso del sistema web.

Funciones del sistema web

Mediante las entrevistas se identificaron los roles que tienen los usuarios de la aplicación como se describe en la Tabla 2.

Tabla 2 Roles de usuario en el sistema.

Roles	Descripción
SuperAdmin (súper administrador)	Tiene acceso a toda la aplicación, pero su actividad principal es la creación de facultades y un usuario con rol de Administrador por cada facultad
Administrador	Cuenta con las facultades para administrar las carreras y los usuarios que se describen a continuación por cada facultad.
Director	Posee el permiso de acceso a todas las funcionalidades, desde un administrador hasta miembro del comité
Subdirector	Se enfoca en las funcionalidades de un miembro de la comisión evaluadora (Comité), sin embargo, posee los permisos como un director.
Secretaría	Posee los permisos específicos para la creación de nuevos planes de titulación y la asignación de estos a una convocatoria o reunión de la comisión evaluadora.
Comité	Tienen la capacidad para evaluar los planes de titulación y generar las actas correspondientes.

Elaborado por: Andrés Taipe

Descripción de roles del sistema

En el sistema web el usuario administrador podrá realizar las siguientes funciones:

- Iniciar sesión como administrador.
- Editar su contraseña y perfil.
- Administrar las facultades y carreras.

- Administrar los usuarios y asignación de roles.

El usuario de secretaría podrá realizar las siguientes funciones:

- Iniciar sesión como personal de secretaría.
- Editar su contraseña y perfil.
- Ingresar la información de los planes de titulación y de los estudiantes que lo proponen.
- Crear la convocatoria de cada carrera y asignar los planes que ingresan a la convocatoria.
- Visualizar la información de los estudiantes y de los planes creados de toda la facultad.

Los miembros de la comisión podrán realizar las siguientes funciones:

- Iniciar sesión como subdirector o miembro de la comisión.
- Editar su contraseña y perfil.
- Visualizar las convocatorias en conjunto con los planes que han sido asignados.
- Crear las observaciones a cada plan.
- Crear las actas de la convocatoria e imprimirlas.
- Finalizar la convocatoria.

El subdirector, director y súper administrador poseen acceso a todo el sistema incluido las funciones de los roles, anteriormente descritos.

Características del usuario

Los usuarios deberán tener conocimientos básicos acerca de la utilización de un computador, y de los procesos respectivos al área en el cual se desempeñan.

3.2 Requerimientos específicos del sistema web.

Funcionalidades del sistema web.

Identificación y autenticación de usuarios

Para los distintos roles descritos en la Tabla 2, se debe identificar con un correo y contraseñas creadas por el súper administrador o administrador las cuales serán autenticadas. Cuando se crea un usuario nuevo se envía un correo con las credenciales

para que pueda ingresar al sistema, la contraseña dada deberá ser cambiada la primera vez que ingresa al sistema.

Administración de facultades

Las facultades son creadas por el usuario súper administrador.

Administración de carreras

Las carreras son creadas por el administrador de cada facultad, la cual fue creada con anterioridad por el súper administrador. Existen dos tipos de carreras a crear, de grado y postgrado.

Administración de usuarios

Para la creación de los usuarios deben estar registradas con anticipación las facultades y carreras.¹ Posee todas funcionalidades para habilitar, inhabilitar, modificar o eliminar a un usuario.

Administración de planes de titulación

Los planes de titulación pueden ser administrados por el personal de secretaría y se necesitará el plan de titulación en formato pdf para ingresarlo al sistema.

Administración de estudiantes

Los estudiantes son ingresados conforme se crean los planes de titulación con la información necesaria para el sistema, sin embargo, se puede añadir o editar sus datos.

Creación de convocatorias

Las convocatorias son creadas por el personal de secretaría para que la Comisión pueda revisar y evaluar los planes de titulación.

Creación de categorías e ítems para el acta de reunión de la comisión

El acta que posee la rúbrica de cada plan con el cual se aprueba o no, contiene varias categorías y en cada una de ellas varios ítems. Cada categoría e ítems son editables para poderlos modificar o eliminar si ya no son necesarios.

¹ El Administrador que es creado por el Súper Administrador solo necesita de la creación previa de la facultad al que va a ser asignado.

Evaluación de los planes de titulación

Los miembros de la comisión de la unidad de titulación pueden crear observaciones por cada plan y aprobar, negar o dejar como pendiente de modificaciones los planes. Al final la comisión puede crear e imprimir un acta final de toda la reunión.

Finalización de la reunión

Cuando todos los planes han sido calificados, el sistema posee la opción para terminar la reunión.

3.3 Conformación del equipo Scrum

En la Tabla 3 se describe la conformación del equipo Scrum con sus respectivos roles.

Tabla 3 Equipo de Trabajo y Roles

Personas/Entidades	Rol
Escuela Politécnica Nacional	<i>Product Owner</i>
Ing. Luis Almeida	<i>Scrum Master</i>
Andrés Taipe	<i>Development Team</i>

Elaborado por: Andrés Taipe

3.4 Actividades

Historias de usuario

Las historias de usuario ayudan a definir la funcionalidad del sistema web para lo cual se ha escogido el modelo detallado en la Tabla 4 con las siguientes características:

- Número: Número de la historia de usuario
- Usuario: Usuario o rol que tiene la persona
- Nombre historia: Nombre de la historia de usuario
- Prioridad: (alta, media, baja)
- Riesgo: (alta, media, baja)
- Puntos (1 a 5) de menor a mayor en nivel de dificultad
- Sprint: Número de sprint
- Programador responsable
- Descripción: Descripción de las acciones a realizar
- Observaciones

Tabla 4 Modelo de historia de usuario

Historia de Usuario	
Número: 1	Usuario: Súper Administrador
Nombre historia: Creación de Facultades y el Administrador por cada una de ellas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 5	Sprint: 1
Programador responsable: Andrés Taípe	
Descripción:	
Observaciones:	

La lista de las historias de usuario las puede encontrar en el Anexo 1.

Product backlog

Para que el sistema pueda llegar a entregarse en los tiempos requeridos y conforme a la metodología Scrum, es necesario desarrollar el *Product backlog*, el cual es creado a partir de las historias de usuario las mismas que se agruparon en categorías para hacer más simple la visualización como lo observamos en la Tabla 5, la cual contiene los siguientes campos:

- Descripción
- Prioridad (1 a 5) ordenado de menor a mayor urgencia
- Complejidad (1 a 5) ordenado de menor a mayor complejidad
- Sprint
- Duración (días)

Tabla 5 *Product backlog*

Descripción	Prioridad	Complejidad	Sprint	Duración (días)
Configuración inicial del proyecto	5	5	1	15

Desarrollo del módulo de autenticación y usuarios	4	5	2	20
Desarrollo del módulo carreras y planes	4	4	3	20
Desarrollo del módulo convocatoria y comisión	4	4	4	20
Pruebas, documentación e implementación	5	3	5	20
Total				95

Elaborado por: Andrés Taipe

Sprints

La Tabla 6 describe las tareas del Sprint backlog de acuerdo con el Product backlog generado en la Tabla 5.

Tabla 6 Sprints

Sprint	Rol Scrum	Product backlog	Sprint backlog	Esfuerzo (días)
1	<i>Development team</i>	Configuración inicial del proyecto	Creación de la base de datos	5
			Configuración del ambiente de desarrollo	5
			Creación del proyecto	3
			Integración del <i>frontend</i> , <i>backend</i> y base de datos y pruebas de conexión	2
2	<i>Development team</i>	Desarrollar un módulo de autenticación y gestión de usuarios	Crear CRUD ² de usuarios (<i>backend</i>)	5
			Crear <i>API REST</i> para autenticación y gestión de usuarios (<i>backend</i>)	2
			Creación de la página de inicio de sesión (<i>frontend</i>)	3
			Configuración para módulo admin en Angular (<i>frontend</i>)	3
			Creación de página para restaurar contraseña (<i>frontend</i>)	2
			Creación del módulo de administración de usuarios (<i>frontend</i>)	5
3	<i>Development team</i>	Desarrollar un módulo de administración de carreras y planes.	Crear CRUD de carreras (<i>backend</i>)	5
			Crear CRUD de planes (<i>backend</i>)	5
			Crear <i>API REST</i> para gestión de carreras y planes (<i>backend</i>)	2

² **CRUD** (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, pero también puede aplicarse a funciones de un nivel superior de una aplicación como soft deletes donde la información no es realmente eliminada, sino es marcada como eliminada a través de un estatus (MDN web docs, 2019).

			Creación del módulo de administración de carreras (<i>frontend</i>)	4
			Creación del módulo de creación de planes (<i>frontend</i>)	4
4	<i>Development team</i>	Desarrollar un módulo para la administración de convocatorias y reuniones de la comisión.	Crear CRUD de convocatorias (<i>backend</i>)	5
			Crear CRUD de reuniones (<i>backend</i>)	5
			Crear <i>API REST</i> para gestión de convocatorias y reuniones (<i>backend</i>)	2
			Creación del módulo de administración de convocatorias (<i>frontend</i>)	4
			Creación del módulo de creación de reuniones (<i>frontend</i>)	4
5	<i>Development team</i>	Pruebas del sistema, realización de la documentación e implementación	Pruebas de funcionalidad	5
			Pruebas de rendimiento	1
			Creación del manual de usuario	2
			Creación del documento del proyecto de titulación	11
			Implementación del sistema	1
Total				95

Elaborado por: Andrés Taipe

Para cumplir con esta planificación se utilizó la herramienta llamada ZenHub la cual es implementada dentro de GitHub en el repositorio del proyecto creado para versionar el desarrollo del sistema como se muestra en la Figura 4.

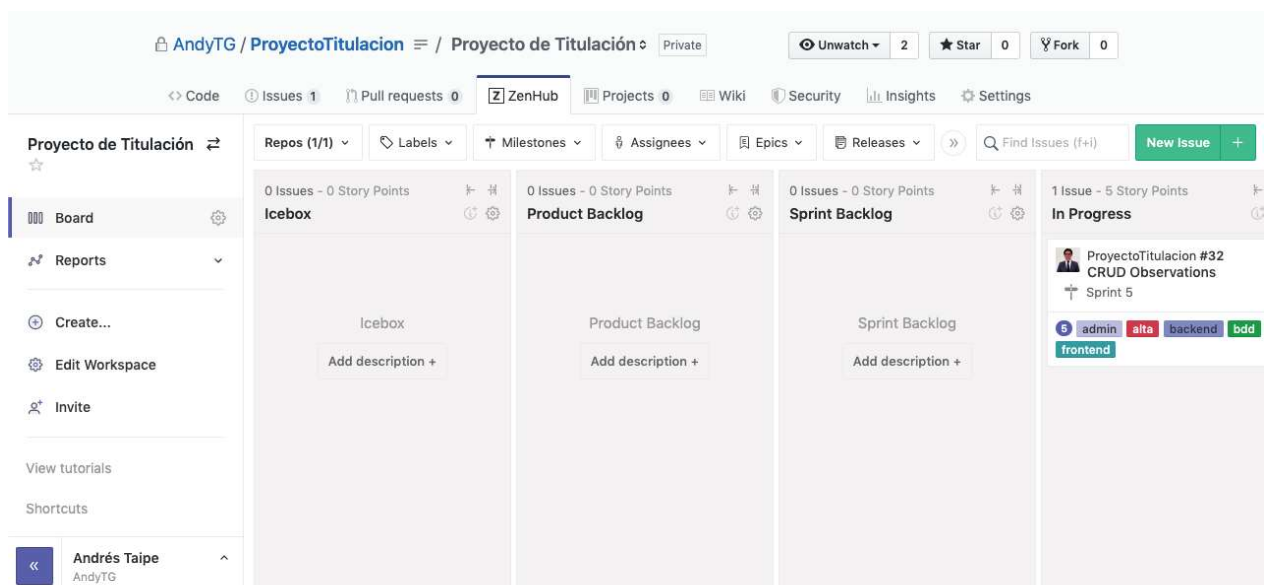


Figura 4 Panel de administración ZenHub

3.5 Creación de la base de datos

Para la creación de la base de datos se utiliza PostgreSQL (PostgreSQL, 2019), el cual ofrece la facilidad de instalarlo en cualquier sistema operativo. Como visualizamos en la Figura 5, la tabla *user* describe a los usuarios de nuestra aplicación web los cuales tendrán un rol y pertenecen a una facultad (*faculty*), los usuarios con el rol Comité, descrito en la Tabla 2, necesitamos saber la carrera a la que pertenece, por tal razón se usa la clave foránea de *career* en *user*.

En la tabla *plan* tiene una relación de uno a muchos con *file*, para guardar todas las versiones de los planes de titulación que se guarden, es decir, cuando un plan no es aprobado y se envía a modificar se puede añadir al sistema una nueva versión del archivo. La clave foránea de la tabla *plan* se guarda en la tabla *student* ya que un plan puede pertenecer hasta tres estudiantes de diferentes carreras.

La tabla *observation* permite mantener la información de los comentarios que realicen todos los miembros de la comisión al revisar el plan de titulación, y las tablas asociadas a *rubric* presentarán las observaciones finales, las cuales dan como resultado que un plan sea aprobado, negado o enviado a realizar modificaciones.

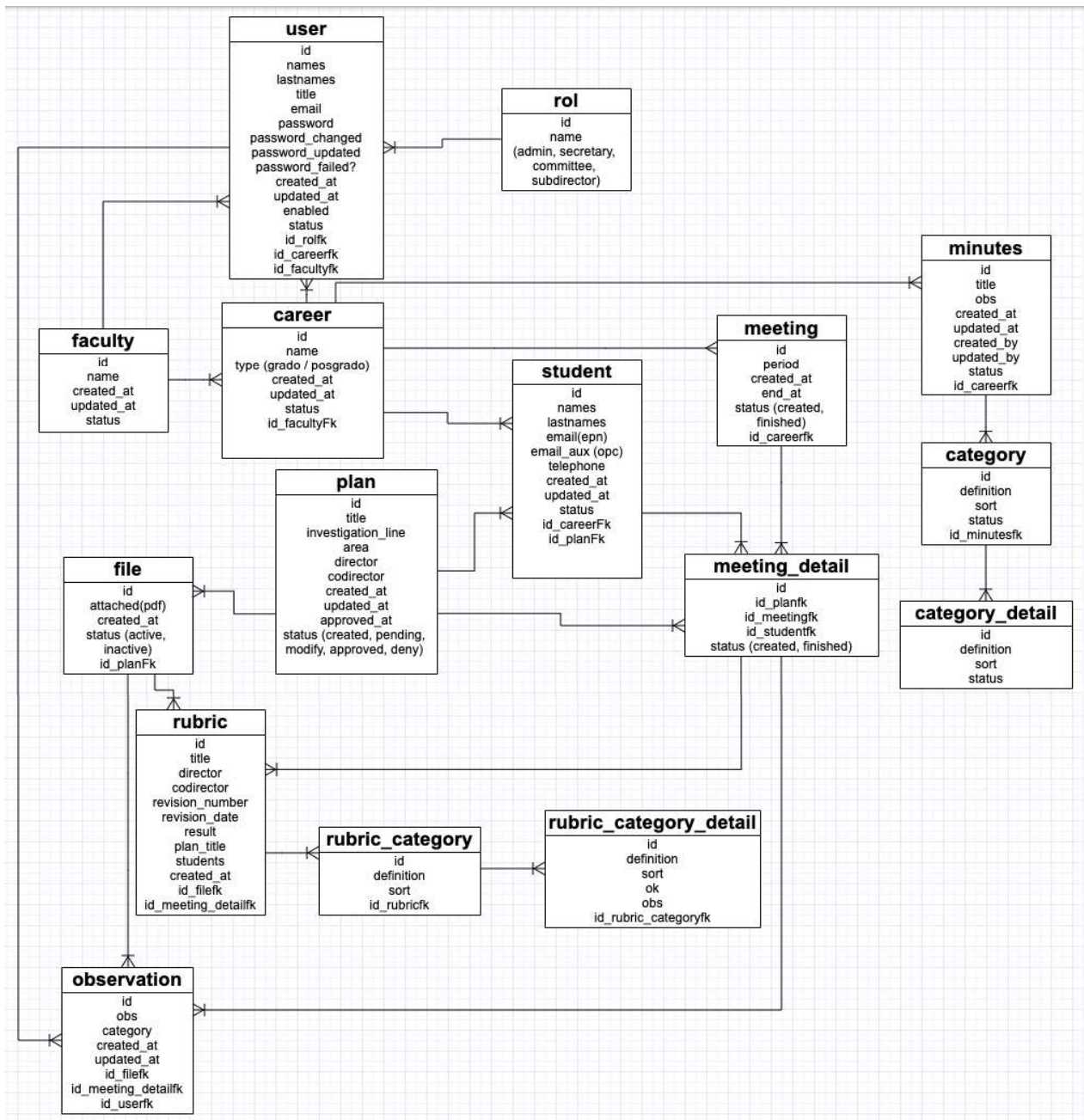


Figura 5 Diseño físico de la base de datos

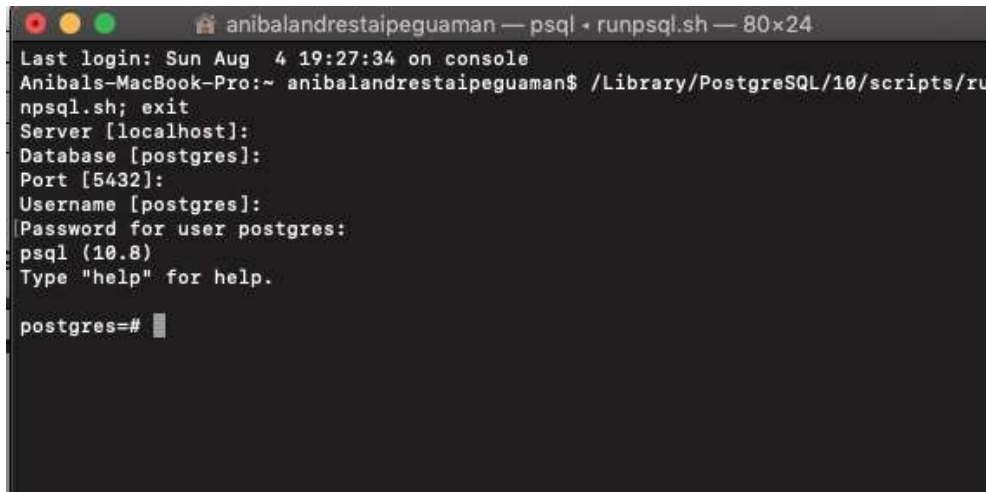
Para nuestro proyecto hemos definido 3 ambientes de desarrollo para las cuales utilizaremos diferentes bases de datos descritos en la Tabla 7

Tabla 7 Bases de datos según ambientes de desarrollo

Base de Datos	Ambiente	Descripción
devplantesis	dev	Desarrollo
stageplanteis	stage	Pruebas
prodplantesis	production	Producción

Elaborado por: Andrés Taipe

Los pasos de instalación de PostgreSQL se describen a detalle en su página oficial, para crear la base de datos mediante nuestra terminal o consola se utiliza la propia del gestor de base de datos que se instala en conjunto como un solo paquete. Se ingresan las configuraciones que se necesite por defecto como lo visualizamos en la Figura 6.

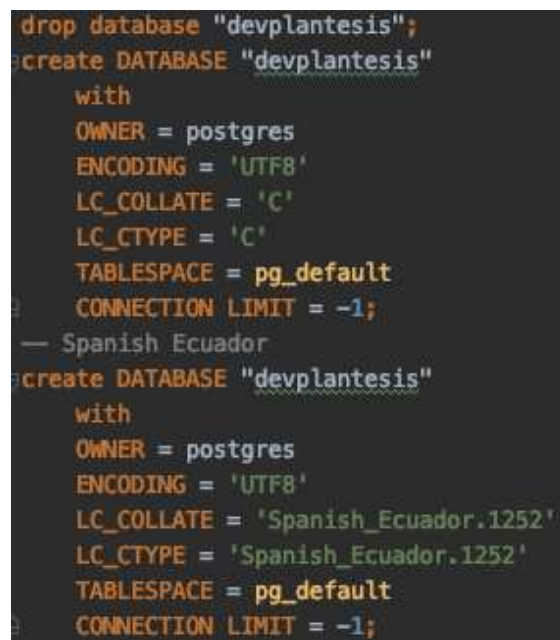


```
anibalandrestaiteguaman — psql · runpsql.sh — 80x24
Last login: Sun Aug  4 19:27:34 on console
Anibals-MacBook-Pro:~ anibalandrestaiteguaman$ /Library/PostgreSQL/10/scripts/runpsql.sh; exit
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (10.8)
Type "help" for help.

postgres=#
```

Figura 6 Consola de PostgreSQL

Para crear la base de datos se debe en cuenta la configuración regional de la computadora o el servidor que se va a utilizar. Los casos más comunes son el inglés y el español y se puede configurar de acuerdo con lo que se muestra en la Figura 7.



```
drop database "devplantesis";
create DATABASE "devplantesis"
with
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'C'
  LC_CTYPE = 'C'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;
— Spanish Ecuador
create DATABASE "devplantesis"
with
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Spanish_Ecuador.1252'
  LC_CTYPE = 'Spanish_Ecuador.1252'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;
```

Figura 7 Creación de base de datos según configuración regional

Se crea las tablas y los campos en la base de datos. Para ello se ingresa a la base de datos con el comando mostrado en la Figura 8 y se crea un nuevo *schema* llamado epn con el comando de la Figura 9.

```
— Hacer modificaciones dentro de la base creada  
c devplantesis;
```

Figura 8 Código de ingreso a la base de datos creada

```
— DROP SCHEMA epn ;  
create SCHEMA epn  
AUTHORIZATION postgres;
```

Figura 9 Creación del schema epn

3.6 Creación de la página web.

En la Figura 3 se describió las interfaces de comunicación entre tecnologías que se utiliza en el desarrollo, se desarrolló con dos *frameworks*, para el desarrollo de *backend* se instaló Node.js desde su página oficial (NodeJs, 2019) y para el *frontend* se utilizó Angular con los comandos que se muestran en la Figura 10. Lo importante de estos *frameworks* es que usan el mismo lenguaje de programación JavaScript.

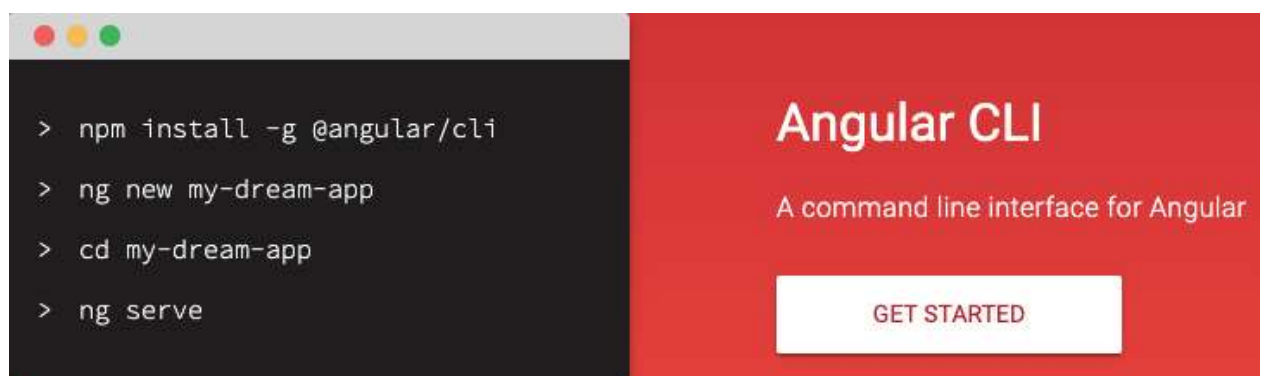
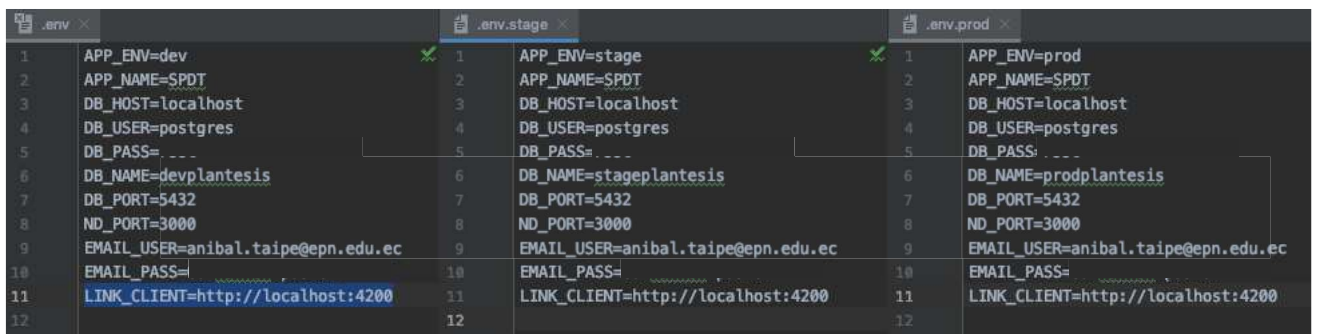


Figura 10 Comandos de instalación y creación de proyecto en Angular

3.7 Desarrollo de la aplicación web para la gestión de los planes de unidad de titulación.

Conexión del *backend* con la base de datos

Para realizar la conexión en los diferentes ambientes de desarrollo se crearon archivos `.env` como se muestra en la Figura 11, para cada uno de los ambientes con sus respectivas configuraciones, claves y rutas de acceso.



Line	dev (.env)	stage (.env.stage)	prod (.env.prod)
1	APP_ENV=dev	APP_ENV=stage	APP_ENV=prod
2	APP_NAME=\$PDT	APP_NAME=\$PDT	APP_NAME=\$PDT
3	DB_HOST=localhost	DB_HOST=localhost	DB_HOST=localhost
4	DB_USER=postgres	DB_USER=postgres	DB_USER=postgres
5	DB_PASS=...	DB_PASS=...	DB_PASS=...
6	DB_NAME=devplantesis	DB_NAME=stageplantesis	DB_NAME=prodplantesis
7	DB_PORT=5432	DB_PORT=5432	DB_PORT=5432
8	ND_PORT=3000	ND_PORT=3000	ND_PORT=3000
9	EMAIL_USER=anibal.taipe@epn.edu.ec	EMAIL_USER=anibal.taipe@epn.edu.ec	EMAIL_USER=anibal.taipe@epn.edu.ec
10	EMAIL_PASS=	EMAIL_PASS=	EMAIL_PASS=
11	LINK_CLIENT=http://localhost:4200	LINK_CLIENT=http://localhost:4200	LINK_CLIENT=http://localhost:4200
12			

Figura 11 Configuración ambientes de desarrollo backend

Conectar el *backend* con la base de datos no requiere una configuración muy grande, para ello utilizamos las líneas de código descritos en la Figura 12.

```
const {Pool} = require('pg');
require('custom-env').env();

const pool = new Pool({
  user: process.env.DB_USER,
  host: process.env.DB_HOST,
  database: process.env.DB_NAME,
  password: process.env.DB_PASS,
  port: process.env.DB_PORT
});

console.log(process.env.NODE_ENV);

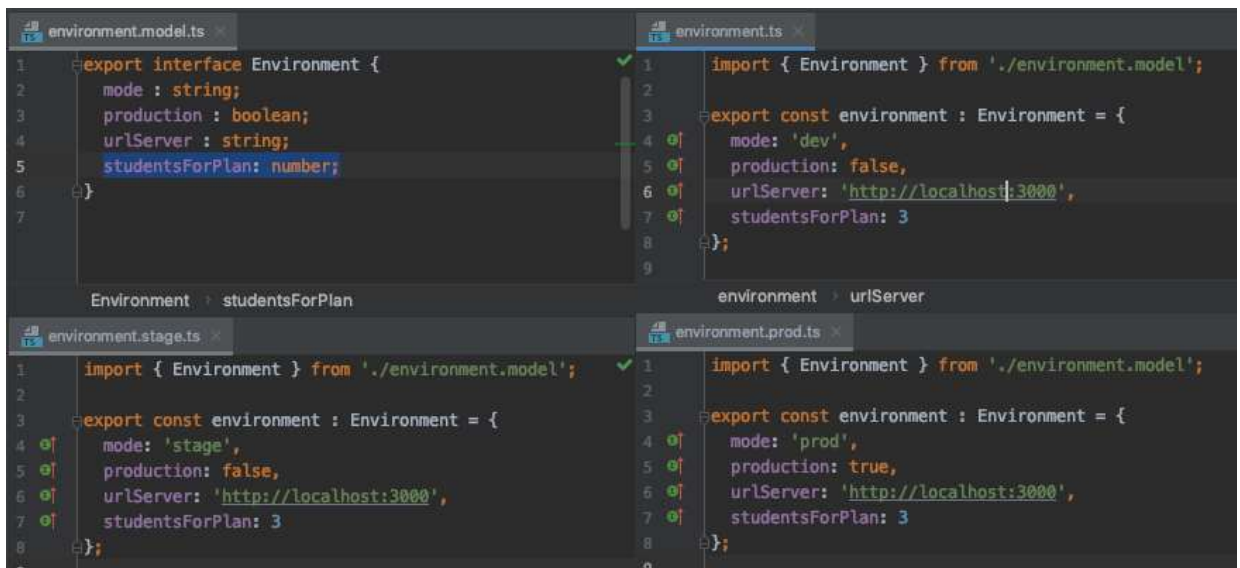
module.exports = {
  query: (text, params, callback) => {
    return pool.query(text, params, callback);
  }
};
```

Figura 12 *Pool* de conexión con la base de datos

Al crear el *pool* de conexión mostrado en la Figura 12, se puede realizar el CRUD de todas las tablas y las consultas que se necesiten a la base de datos.

Conexión del *frontend* con las APIs del backend

Para comunicar y transmitir los datos desde el *backend* al *frontend* y viceversa, utilizaremos las APIs, que proporcionan una interfaz de comunicación mediante protocolos http. Es necesario configurar en el *frontend* los archivos que contendrán las configuraciones de los ambientes de desarrollo con las distintas conexiones al *backend* como se muestra en la Figura 13.



```
environment.model.ts
1 export interface Environment {
2   mode : string;
3   production : boolean;
4   urlServer : string;
5   studentsForPlan: number;
6 }
7

environment.ts
1 import { Environment } from './environment.model';
2
3 export const environment : Environment = {
4   mode: 'dev',
5   production: false,
6   urlServer: 'http://localhost:3000',
7   studentsForPlan: 3
8 };
9

environment.stage.ts
1 import { Environment } from './environment.model';
2
3 export const environment : Environment = {
4   mode: 'stage',
5   production: false,
6   urlServer: 'http://localhost:3000',
7   studentsForPlan: 3
8 };
9

environment.prod.ts
1 import { Environment } from './environment.model';
2
3 export const environment : Environment = {
4   mode: 'prod',
5   production: true,
6   urlServer: 'http://localhost:3000',
7   studentsForPlan: 3
8 };
9
```

Figura 13 Configuración ambientes de desarrollo *frontend*

Se crean los servicios de cada una de las entidades que se requieren que se comuniquen con nuestro backend mediante REST. En la Figura 14 se muestran las carpetas que contienen el código de los servicios creados en el desarrollo de la aplicación web y en la Figura 15 se visualiza el código de un servicio.

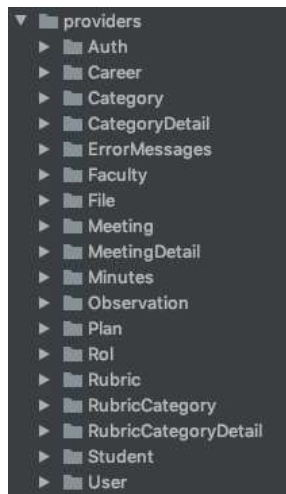


Figura 14 Servicios para conexión con el *backend*

```
import { Injectable } from '@angular/core';
import { environment } from '../../environments/environment';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { CategoryModel } from '../../models/CategoryModel';

@Injectable( {
  providedIn: 'root'
} )
export class CategoryService {

  constructor( private http : HttpClient ) {
  }

  getAll( id ) {
    return this.http.get( url: environment.urlServer + '/category/byMinute/' + id );
  }

  create( category : CategoryModel ) : Observable<any> {
    return this.http.post( url: environment.urlServer + '/category', category );
  }

  updateStatus( id : number, category : Partial<CategoryModel> ) {
    return this.http.patch( url: environment.urlServer + '/category/' + id, category );
  }

  update( id : number, category : CategoryModel ) {
    return this.http.put( url: environment.urlServer + '/category/' + id, category );
  }

  delete( id : number ) {
    return this.http.delete( url: environment.urlServer + '/category/' + id );
  }

  getOne( id ) {
    return this.http.get( url: environment.urlServer + '/category/' + id );
  }
}
```

Figura 15 Ejemplo de conexión con API REST en backend

En nuestro *backend* se deben configurar las rutas mediante las cuales nuestro *frontend* va a comunicarse, las rutas se muestran en la Figura 16.

```
index.js
const careerRouter = require('../api/career');
const planRouter = require('../api/plan');
const fileRouter = require('../api/file');
const studentRouter = require('../api/student');
const minutesRouter = require('../api/minutes');
const categoryRouter = require('../api/category');
const categoryDetailRouter = require('../api/category_detail');
const meetingRouter = require('../api/meeting');
const meetingDetailRouter = require('../api/meeting_detail');
const rubricRouter = require('../api/rubric');
const rubricCategoryRouter = require('../api/rubric_category');
const rubricCategoryDetailRouter = require('../api/rubric_category_detail');
const observationRouter = require('../api/observation');

module.exports = (app) => {
  app.use('/users', usersRouter);
  app.use('/', indexRouter);
  app.use('/login', authRouter);
  app.use('/rois', roisRouter);
  app.use('/faculty', facultyRouter);
  app.use('/career', careerRouter);
  app.use('/plan', planRouter);
  app.use('/file', fileRouter);
  app.use('/student', studentRouter);
  app.use('/minutes', minutesRouter);
  app.use('/category', categoryRouter);
  app.use('/categoryDetail', categoryDetailRouter);
  app.use('/meeting', meetingRouter);
  app.use('/meetingDetail', meetingDetailRouter);
  app.use('/rubric', rubricRouter);
  app.use('/rubricCategory', rubricCategoryRouter);
  app.use('/rubricCategoryDetail', rubricCategoryDetailRouter);
  app.use('/observation', observationRouter);
  app.use('/public', express.static('public'));
};
```

Figura 16 Rutas de comunicación con el *backend*

Creación del módulo de autenticación

Para la autenticación se requiere un formulario que contenga un correo y una contraseña que se envíe al correo electrónico cuando se hace el proceso de registro del usuario por parte del administrador. En la Figura 17 encontramos el diseño final de nuestro formulario de Inicio de Sesión y en la Figura 18 el modelo del correo que recibe el usuario.

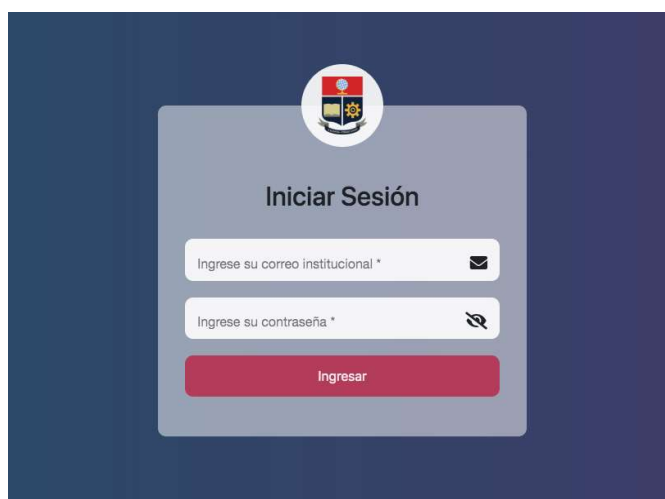


Figura 17 Inicio de Sesión

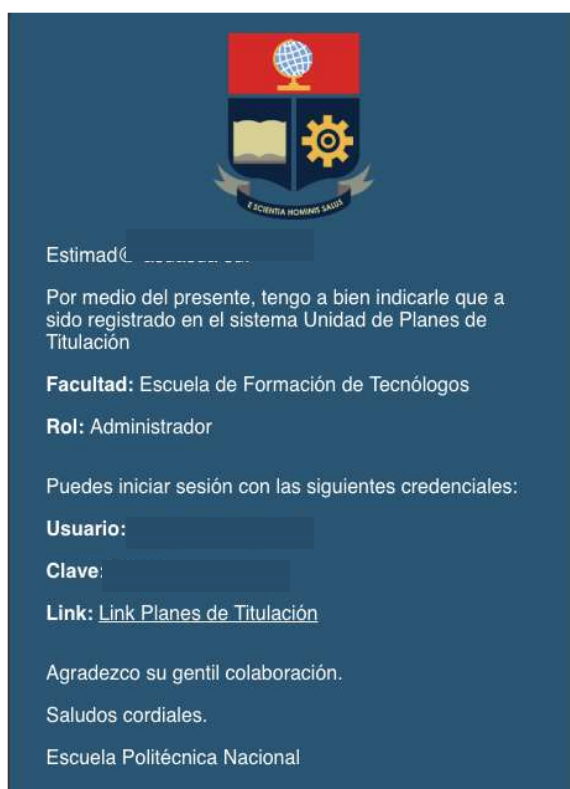


Figura 18 Correo con las credenciales de acceso

Creación del módulo de restaurar contraseña

Este módulo contiene un formulario para validar una nueva contraseña como lo visualizamos en la Figura 19. Se ejecuta la primera vez que un usuario intenta ingresar con las credenciales enviadas al correo electrónico para brindarle mayor seguridad a la hora de ingresar en el sistema.

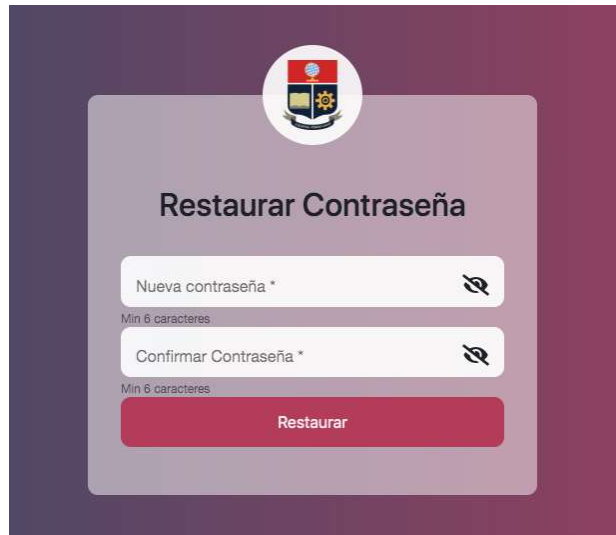


Figura 19 Restaurar contraseña

Creación del módulo de administración

El módulo de administración está pensado para realizar todas las actividades que el usuario con el rol Administrador las necesite. La Figura 20 muestra el menú de Administración con los ítems Carreras y Usuarios.



Figura 20 Menú administración

Para el ítem carreras se visualiza una lista que detalle las carreras según la facultad seleccionada, con una sección para acciones (editar, eliminar) como se puede ver en la Figura 21.

Administración Facultades

Facultades
Escuela de Formación de Tecnólogos

No.	Carrera	Tipo	Fecha de Creación	Estado	Acciones
1	Tecnología en Electromecánica	Grado	28-05-2019	<input checked="" type="checkbox"/> Deshabilitar	
2	Tecnología en Análisis de Sistemas Informáticos	Grado	28-05-2019	<input checked="" type="checkbox"/> Deshabilitar	
3	Tecnología en Electrónica y Telecomunicaciones	Grado	28-05-2019	<input checked="" type="checkbox"/> Deshabilitar	
4	Tecnología en Agua y Saneamiento Ambiental	Grado	29-07-2019	<input checked="" type="checkbox"/> Deshabilitar	

Figura 21 Lista de carreras

Se creó un formulario para ingresar nuevas carreras las mismas que deben pertenecer únicamente a la facultad seleccionada. En la Figura 22 se muestra un formulario para crear la carrera que puede ser de tipo Grado o Postgrado.

Ingresar nueva carrera

Nombre de la carrera *

Seleccionar Tipo *

Figura 22 Formulario para crear nueva carrera

El ítem usuarios hace referencia a todos los usuarios que van a utilizar la aplicación. Se creó en primer lugar una lista para visualizar los usuarios creados, un buscador y también para realizar distintas acciones (editar, eliminar) a cada uno de estos como se muestra en la Figura 23.

Buscar Usuario						
No. Usuario	Correo	Rol	Facultad	Carrera	Estado	Acciones
1	Edwin Salvador edwin.salvador@epn.edu.ec	Administrador	Escuela de Formación de Tecnólogos		<input checked="" type="checkbox"/> Deshabilitar	
2	Melany Borja meli.borja@epn.edu.ec	Secretaría	Escuela de Formación de Tecnólogos		<input checked="" type="checkbox"/> Deshabilitar	
3	Carlos Romo Herrera c.romo@epn.edu.ec	Subdirector	Escuela de Formación de Tecnólogos		<input checked="" type="checkbox"/> Deshabilitar	
4	Juan Pablo Zaldumbide juan.zaldumbide@epn.edu.ec	Comité	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	<input checked="" type="checkbox"/> Deshabilitar	
5	Ivonne Maldonado ivonne.maldonado@epn.edu.ec	Comité	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	<input checked="" type="checkbox"/> Deshabilitar	
6	Marina Vintimilla marina.vintimilla@epn.edu.ec	Comité	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	<input checked="" type="checkbox"/> Deshabilitar	

Figura 23 Lista de usuarios

También se creó un formulario para ingresar los nuevos usuarios. A cada nuevo usuario creado se enviará un correo con las credenciales para ingresar al sistema como se pudo visualizar en la Figura 18. Como se muestra en la Figura 24 todos los campos del formulario son obligatorios y la contraseña es generada automáticamente.

Ingresar nuevo usuario

Ingresar nombres *

Ingresar apellidos *

Ingresar email *

Ingresar contraseña *

Seleccionar Rol *

Seleccionar Facultad

Seleccionar Carrera

Figura 24 Formulario para crear nuevo usuario

La creación de nuevos usuarios tiene las restricciones descritos en la Tabla 8:

Tabla 8 Cantidad permitida de usuarios a crear

Rol	Cantidad	Descripción
Administrador	1	Un usuario por facultad habilitado
Director	1	Un usuario por facultad habilitado
Subdirector	1	Un usuario por facultad habilitado
Secretaría	ilimitado	No tiene cantidad definida
Comité	3	Tres usuarios por carrera habilitados

Elaborado por: Andrés Taipe

El súper administrador no es administrable, es decir, no se puede crear, editar o eliminar desde el sistema para resguardar la seguridad la información, ya que este usuario puede acceder a toda la información de todas las facultades, su creación se la realiza a nivel de base de datos directamente en el servidor que alojará la aplicación web.

Creación del módulo planes

En este módulo se crearon dos ítems importantes. En la Figura 25 vemos un ítem llamado Nuevo para ingresar al sistema los planes nuevos de titulación y otro llamado Lista para visualizar los planes que ya están creados.



Figura 25 Menú planes

En el ítem Nuevo del módulo Planes se creó un formulario para ingresar toda la información necesaria del plan de titulación. El sistema acepta únicamente archivos en formato PDF. Como se observa en la Figura 26, para crear un plan de titulación se lo dividió en 3 partes, en primer lugar, se ingresa en el formulario presentado la información de los planes de titulación.

Figura 26 Formulario con datos del plan a crear

En segundo lugar, se ingresa en el formulario presentado en la Figura 27 la información de el o los estudiantes que realizan el plan de titulación.

The screenshot shows a three-step process for creating a plan. Step 2, 'Datos del Estudiante', is active. It features a header with three steps: '1 Datos del Plan', '2 Datos del Estudiante', and '3 Resumen'. Below the header, there is a section for 'Estudiante 1' with four input fields: 'Nombres *', 'Apellidos *', 'Correo *', and 'Seleccionar Carrera *'. A red button with a plus sign and the text 'Añadir estudiante' is centered below the fields. At the bottom left, there are navigation buttons: '< Atrás' and 'Siguiente >'.

Figura 27 Formulario de creación de estudiantes

Y finalmente, a modo de resumen se muestra la información añadida y una pequeña visualización del plan de titulación el cual podemos ingresarlo al presionar el botón de guardar como lo observamos en la Figura 28.

The screenshot shows the 'Resumen' screen, which is the third step in the process. The header shows '1 Datos del Plan', '2 Datos del Estudiante', and '3 Resumen'. The main content area is divided into two columns. The left column contains a summary of the plan: 'Título:', 'Director:', 'Codirector:', 'Archivo:', 'Páginas: 1/1', and 'Estudiante 1: Correo:'. The right column is a large dark blue rectangular area, likely a placeholder for a visualization of the plan. At the bottom left, there are navigation buttons: '< Atrás', 'Borrar', and 'Guardar'.

Figura 28 Resumen del plan a crear

El ítem Lista contiene la información de todos los planes que han sido ingresados en el sistema, en la Figura 29 se muestra un ejemplo de una lista de planes de titulación con la fecha de creación y un botón para visualizar el archivo añadido.

Buscar plan

No.	Título	Propuesto por	Estado	Fecha creación	Acciones
1	DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE DISTRIBUIDORES Y EXPENDIO DE GAS LICUADO DE PETRÓLEO (GLP) A DOMICILIO EN LA CIUDAD DE QUITO	Katherine Sorelly Lasluisa Villagran Bryan Steeven Bohórquez Castillo	Presentado	31-07-2019	
2	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	Presentado	31-07-2019	

Usuarios por página 10 1 - 2 of 2 < >

Figura 29 Lista de planes ingresados

En todo el ciclo de los planes de titulación se presentarán diversos estados los cuales han sido definidos en la Tabla 9.

Tabla 9 Estados de los planes de titulación

Estado	Descripción
Presentado	Cuando el plan ha sido ingresado
En revisión	El plan está listo para ser revisado por la comisión
Pendiente	El plan necesita modificaciones
Aprobado	El plan se encuentra aprobado
Rechazado	El plan ha sido negado

Elaborado por: Andrés Taipe

Creación del módulo convocatoria

Las convocatorias hacen referencias a las reuniones que realizan todos los miembros de las comisiones de las distintas carreras para calificar los planes de titulación. Para ello se creó un módulo Convocatoria que se visualiza en la Figura 30, en el cual se pueden añadir los planes que se requieren calificar por cada carrera en el ítem Activas y en Historial encontramos una bitácora de las convocatorias pasadas.

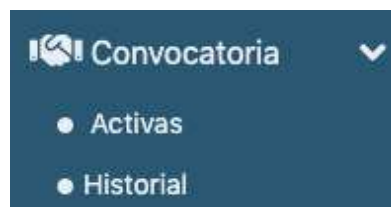


Figura 30 Menú convocatoria

En el ítem Activas podremos visualizar la lista de todas las convocatorias activas que se han creado en una tabla mostrada en la Figura 31.

Buscar Convocatoria

No.	Período	Carrera	Facultad	Fch. Creación	Estado	Acciones
1	2019-A	Tecnología en Análisis de Sistemas Informáticos	Escuela de Formación de Tecnólogos	31/07/2019	Activa	
2	2019-A	Tecnología en Electrónica y Telecomunicaciones	Escuela de Formación de Tecnólogos	05/08/2019	Activa	

Usuarios por página 10 1 - 2 of 2 < >

Figura 31 Lista de convocatorias activas

Además, fue necesario crear una lista seleccionable con los planes que aún no se han añadido para revisión, como se observa en la Figura 32 se puede añadir uno o mas planes a la convocatoria que se desee según la carrera a la que pertenecen los estudiantes.

Seleccionar Planes

<input type="checkbox"/> Título	Propuesto por	Fecha creación	Estado
<input type="checkbox"/> DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE DISTRIBUIDORES Y EXPENDIO DE GAS LICUADO DE PETRÓLEO (GLP) A DOMICILIO EN LA CIUDAD DE QUITO	Katherine Sorelly Lastuisa Villagran	31-07-2019	Presentado
<input type="checkbox"/> DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Bryan Steeven Bohórquez Castillo	31-07-2019	Presentado
<input type="checkbox"/> DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	31-07-2019	Presentado

Añadir **Cancelar**

Figura 32 Lista de planes para revisar

En la Figura 33 se puede ver las convocatorias activas con la lista de los planes que se añadieron.

Buscar Convocatoria

No.	Período	Carrera	Facultad	Fch. Creación	Estado	Acciones								
1	2019-A	Tecnología en Análisis de Sistemas Informáticos	Escuela de Formación de Tecnólogos	31/07/2019	Activa									
<table border="1"> <thead> <tr> <th>No.</th> <th>Título del Plan</th> <th>Propuesto por</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>1.1</td> <td>DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT</td> <td>Anibal Andrés Taipe Guamán</td> <td>En Revisión</td> </tr> </tbody> </table>							No.	Título del Plan	Propuesto por	Estado	1.1	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	En Revisión
No.	Título del Plan	Propuesto por	Estado											
1.1	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	En Revisión											
2	2019-A	Tecnología en Electrónica y Telecomunicaciones	Escuela de Formación de Tecnólogos	05/08/2019	Activa									

Usuarios por página 10 1 - 2 of 2 < >




Figura 33 Detalles de convocatorias activas

Creación del módulo estudiantes

En este módulo solo se presentará la lista de los estudiantes que se han creado al añadir un plan. En la Figura 34 se ve la lista de los estudiantes mostrados en una tabla con el

estado de sus respectivos planes de titulación, esta lista se la visualizará en el módulo Estudiantes, en el ítem Lista visualizada en la Figura 35

Buscar Estudiante

No. Estudiante	Correo	Estado del Plan	Facultad	Carrera	Acciones	
1	Anibal Andrés Taipe Guamán	aniba.taipe@epn.edu.ec	En Revisión	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	
2	Bryan Steveen Bohórquez Castillo	bryanjpg1@hotmail.com	Presentado	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	
3	Katherine Sorelly Lasluisa Villagran	katherinelasluisa@hotmail.com	Presentado	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	

Usuarios por página 10 1 - 3 of 3 < >

Figura 34 Lista de estudiantes



Figura 35 Menú estudiantes

Creación del módulo comisión

El módulo comisión presenta varios ítems, los cuales nos brindan opciones específicas a realizar por un miembro de la comisión, tales como, administración de las actas, calificar los planes de titulación y una lista de los planes, convocatorias y estudiantes. A todas las opciones que se han descrito se puede acceder en el menú Comisión que se visualiza en la Figura 36.



Figura 36 Menú comisión

En el ítem Admin Actas se puede visualizar el acta de reunión de la comisión evaluadora. La cual es administrable en sus criterios de evaluación, como se visualiza en la Figura 37 la Rúbrica muestra una descripción breve del tema del acta y un botón para ingresar a editar.



Figura 37 Acta de reunión para editar

En la Figura 38 se visualiza el acta de la reunión de la comisión evaluadora, el cual contiene de un título y se encuentra dividido en categorías y detalles. Las categorías y detalles pueden ser creadas, editadas y eliminadas.

Título del Acta *	
ACTA DE REUNIÓN COMISIÓN UNIDAD DE TITULACIÓN	
#	Criterio de evaluación
Categoría 1 * Generalidades	
1	Detalle 1 * El plan cumple el formato vigente.
2	Detalle 2 * El perfil/experiencia del (los) auspiciantes se relaciona al área de estudio del Trabajo de Titulación.
3	Detalle 3 * Las citas y referencias están escritas acorde a las normas APA o IEEE.
Categoría 2 * Título del trabajo de titulación	
1	Detalle 1 * Está redactando de forma clara, precisa y entendible.
2	Detalle 2 * Describe el problema y la solución.
3	Detalle 3 * Describe el tema que es objetivo del estudio.

Figura 38 Administración del acta de reunión

En el ítem calificar planes se presenta un listado de las convocatorias que se crearon y que se encuentran activas con sus respectivos planes de titulación que se necesitan revisar. En la Figura 39 visualizamos la convocatoria perteneciente a una carrera y con el plan que se necesita revisar.

No.	Período	Carrera	Fch. Creación	Estado	Acciones												
1	2019-A	Tecnología en Análisis de Sistemas Informáticos	31/07/2019	Activa	  												
<table border="1"> <thead> <tr> <th>No.</th> <th>Título</th> <th>Propuesto por</th> <th>Fecha creación</th> <th>Estado</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT</td> <td>Anibal Andrés Taipe Guamán</td> <td>31-07-2019</td> <td>En Revisión</td> <td> </td> </tr> </tbody> </table>						No.	Título	Propuesto por	Fecha creación	Estado	Acciones	1	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	31-07-2019	En Revisión	 
No.	Título	Propuesto por	Fecha creación	Estado	Acciones												
1	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taipe Guamán	31-07-2019	En Revisión	 												
Items per page: 10 1 - 1 of 1																	

Figura 39 Lista de convocatorias y actas a revisar

Los miembros de la comisión pueden realizar las observaciones de cada plan y colocar sus comentarios de cada plan que se encuentran revisándolo. Como se observa en la

Figura 40 las observaciones se pueden en cada una de las divisiones que se han realizado, esto concuerda con las categorías que se crearon en la Figura 38.

ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE DOCENCIA
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CARRERA DE TECNOLOGÍA EN ANÁLISIS DE SISTEMAS
INFORMÁTICOS
PLAN DE TRABAJO DE TITULACIÓN
TIPO DE TRABAJO DE TITULACIÓN: PROYECTO INTEGRADOR

I.- INFORMACIÓN BÁSICA	
PROPUESTO POR: Anibal Andrés Taípe Guamán	LÍNEA DE INVESTIGACIÓN: Sistemas de Información ÁREA DE CONOCIMIENTO: <i>Sistemas de Información y Computación Centrada en el Humano</i>
AUSPICIADO POR: Director: Ing. Luis Anibal Almeida Molina, MSc Codirector: Ph.D. María Gabriela Pérez Hernández	FECHA: 04/02/2019
RELACIÓN: Nombre del Proyecto de Investigación: Director del Proyecto de Investigación:	
II.- INFORMACIÓN DEL TRABAJO DE TITULACIÓN	
1. Título del Trabajo de Titulación	

1. Generalidades

Observación realizada para prueba
Juan Pablo | 07-08-2019 15:30

✖ Escribir Observación...

2. Título del trabajo de titulación

✖ Escribir Observación...

3. Planteamiento del Problema

✖ Escribir Observación...

Figura 40 Formulario de observaciones de cada plan

Para la reunión esta desarrollada una página donde puedan calificar cada plan de acuerdo con los parámetros creados en el acta de reunión que anteriormente se creó. En esta página podrán aprobar, negar o enviar a modificar los planes de titulación. En la Figura 41 mostramos la rúbrica final que debe ser ingresada en la reunión de los miembros de la comisión.

ESCUELA POLITÉCNICA NACIONAL			
ESCUELA DE FORMACIÓN DE TECNÓLOGOS			
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"			
Estudiante(s)	Anibal Andrés Taípe Guamán		
Título del Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT		
Director	Ing. Luis Anibal Almeida Molina, MSc		
Codirector	PhD. María Gabriela Pérez Hernández		
Fecha de Revisión	07-08-2019		
Nº Revisión	1		
Resultado	Aprobado	Pendiente	Rechazado

ACTA DE REUNIÓN COMISIÓN UNIDAD DE TITULACIÓN
CARRERA TECNOLOGÍA EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

#	Criterios de Evaluación	Cumple (No/Si)	Observación
Generalidades			
1	El plan cumple el formato vigente.	<input checked="" type="checkbox"/> Si	
2	El perfil/experiencia del (los) auspiciantes se relaciona al área de estudio del Trabajo de Titulación.	<input type="checkbox"/> No	Si se relaciona
3	Las citas y referencias están escritas acorde a las normas APA o IEEE.	<input type="checkbox"/> No	


Título del trabajo de titulación

Figura 41 Acta de reunión y calificación del plan

En la Figura 42 se muestra el acta final que presenta el resultado de todas las actas calificadas.



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS



Período: 2019-A
Carrera: Tecnología en Análisis de Sistemas Informáticos
Fecha: 07-08-2019
Hora: 15:36
Asisten a la Reunión: Ph.D. Carlos Romo Herrera

ACTA DE REUNIÓN DE LA COMISIÓN ACADÉMICA

Ivonne Maldonado
Marina Vintimilla
Juan Pablo Zaldumbide

NO.	TEMA A TRATARSE	ESTADO
1	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTES DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	PENDING

Firmas:

Figura 42 Acta de reunión final de la comisión

Para las actas creadas se desarrolló la opción que puedan imprimir. La Figura 43 muestra el acta ya lista para imprimir.

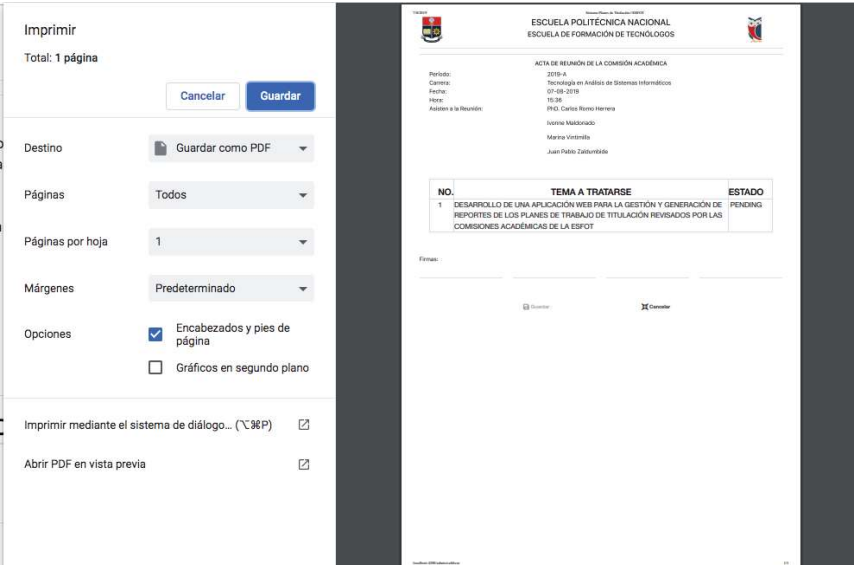


Figura 43 Ventana de impresión del acta de la reunión

En los ítems historial de planes y estudiantes que lo visualizamos en el menú comisión (Figura 36), tenemos la lista de todos los planes ingresados hasta el momento a manera de bitácora de consulta y de los estudiantes que lo han presentado como se puede visualizar en la Figura 44 y 45 respectivamente.

Buscar plan

No.	Título	Propuesto por	Estado	Fecha creación	Acciones
1	DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE DISTRIBUIDORES Y EXPENDIO DE GAS LICUADO DE PETRÓLEO (GLP) A DOMICILIO EN LA CIUDAD DE QUITO	Katherine Sorely Lasluisa Villagran Bryan Steeven Bohórquez Castillo	Presentado	31-07-2019	
2	DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y GENERACIÓN DE REPORTE DE LOS PLANES DE TRABAJO DE TITULACIÓN REVISADOS POR LAS COMISIONES ACADÉMICAS DE LA ESFOT	Anibal Andrés Taípe Guamán	En Revisión	31-07-2019	

Usuarios por página 10 1 - 2 of 2 < >

Figura 44 Historial de planes en módulo comisión

Buscar Estudiante

No. Estudiante	Correo	Estado del Plan	Facultad	Carrera	Acciones
1	Anibal Andrés Taípe Guamán aniba.taípe@epn.edu.ec	En Revisión	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	
2	Bryan Steeven Bohórquez Castillo bryanjpg1@hotmail.com	Presentado	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	
3	Katherine Sorely Lasluisa Villagran katherinelasluisa@hotmail.com	Presentado	Escuela de Formación de Tecnólogos	Tecnología en Análisis de Sistemas Informáticos	

Usuarios por página 10 1 - 3 of 3 < >

Figura 45 Lista de estudiantes en modulo comisión

En la Figura 46 mostramos las convocatorias que ya han finalizado que se muestra en el ítem Historial de convocatorias.

No.	Período	Carrera	Fch. Creación	Fch. Finalización	Estado	Acciones
1	2019-A	Tecnología en Análisis de Sistemas	22/08/2019	22/08/2019	Finalizado	
2	2019-A	Tecnología en Análisis de Sistemas	01/08/2019	22/08/2019	Finalizado	

Figura 46 Lista de estudiantes en modulo comisión

Creación del módulo perfil de usuario

Todos los usuarios presentan la opción de poder editar su perfil como se visualiza en la Figura 47.



Figura 47 Botón editar perfil

No todos los campos son editables, como observamos en la Figura 48 el rol, facultad y carrera solo puede editar el Administrador del sistema, para brindarle una mayor seguridad y que los datos puedan mantenerse íntegros.

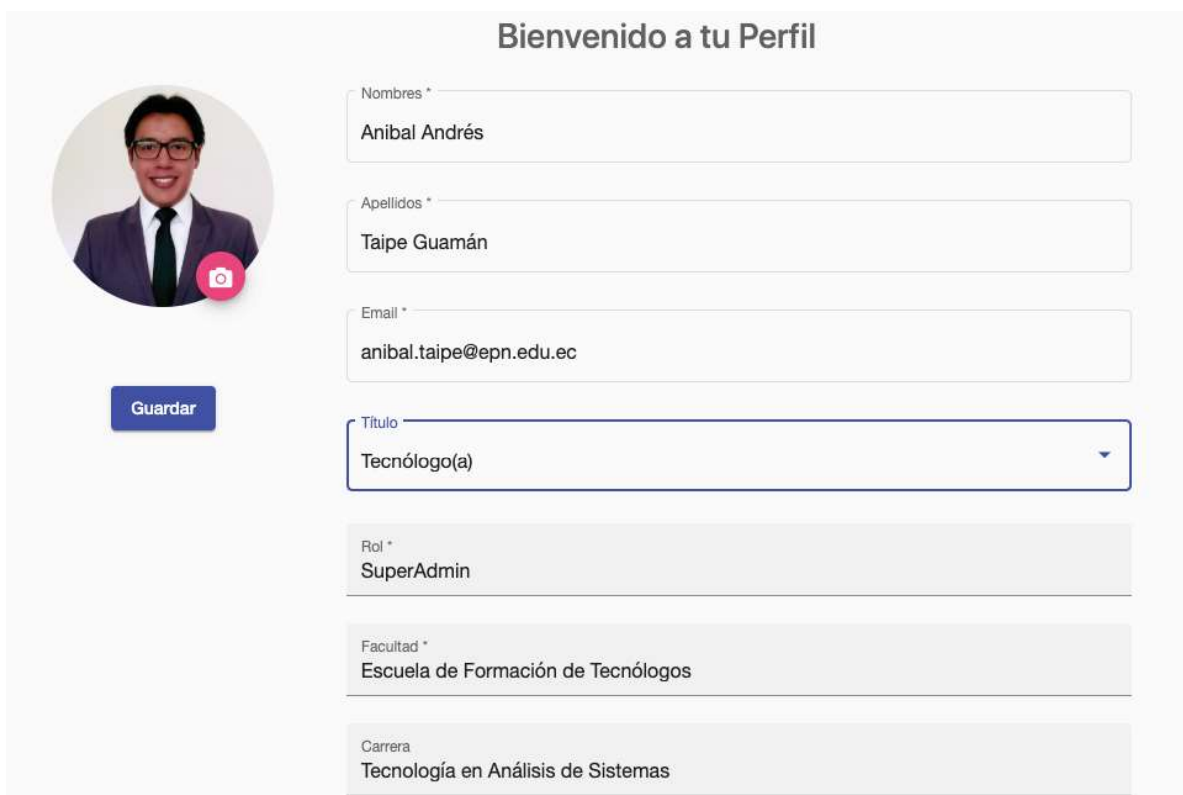
A user profile editing form titled 'Bienvenido a tu Perfil'. On the left is a circular profile picture of the same man, with a blue 'Guardar' button below it. The form fields on the right are: 'Nombres *' with the value 'Anibal Andrés'; 'Apellidos *' with the value 'Taipe Guamán'; 'Email *' with the value 'anibal.taipe@epn.edu.ec'; 'Titulo' with a dropdown menu showing 'Tecnólogo(a)'; 'Rol *' with the value 'SuperAdmin'; 'Facultad *' with the value 'Escuela de Formación de Tecnólogos'; and 'Carrera' with the value 'Tecnología en Análisis de Sistemas'. The 'Rol', 'Facultad', and 'Carrera' fields are disabled.

Figura 48 Formulario de edición del perfil

Creación del *dashboard*

Se creó una página que se muestra para todos los usuarios con pequeñas estadísticas sobre las facultades, carreras, planes de titulación por facultad y por carrera que se han creado. En la Figura 49 se muestra el menú *Dashboard* que al ingresar se visualiza todas las estadísticas en gráficos como lo visualizamos en la Figura 50.

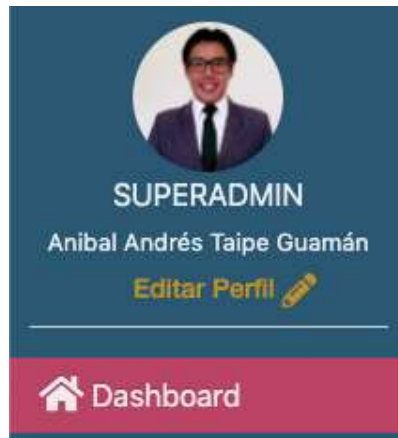


Figura 49 Menú *dashboard*

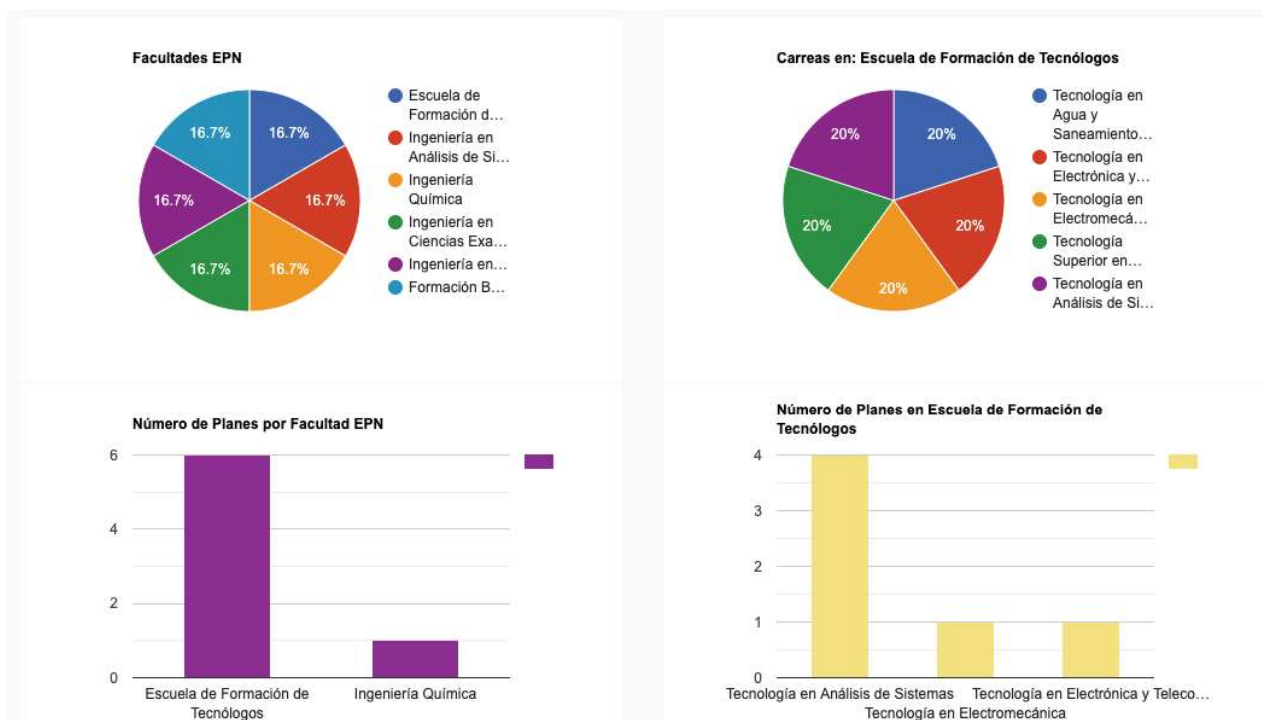


Figura 50 *Dashboard*

3.8 Pruebas de software del sistema web.

Para las pruebas se utilizó las configuraciones de *backend* y *frontend* en ambiente “stage”, esto quiere decir que se usará la base de datos, el puerto y configuraciones internas en el ambiente de pruebas. La IP fija proporcionada por la EPN es la 172.31.115.15 que nos permitirá ingresar mediante el navegador a nuestro sistema. El puerto a utilizar será el 4210 en pruebas y el 4200 en producción.

Pruebas de compatibilidad.

Las pruebas de compatibilidad hacen referencia a probar el sistema en diferentes entornos, en este caso, se verifica que el sistema funcione correctamente en los tres navegadores sugeridos a utilizar en la sección 3.1 Requerimientos generales del sistema web (Google Chrome, Safari, Firefox). Todas las Figuras expuestas en este documento, así como el desarrollo del software se probado en el navegador Google Chrome, por lo que recomendamos usar este navegador para una mejor experiencia. En la Figura 51 mostramos pequeños recortes del sistema funcionando en el navegador Safari y en la Figura 52 en el navegador Firefox.

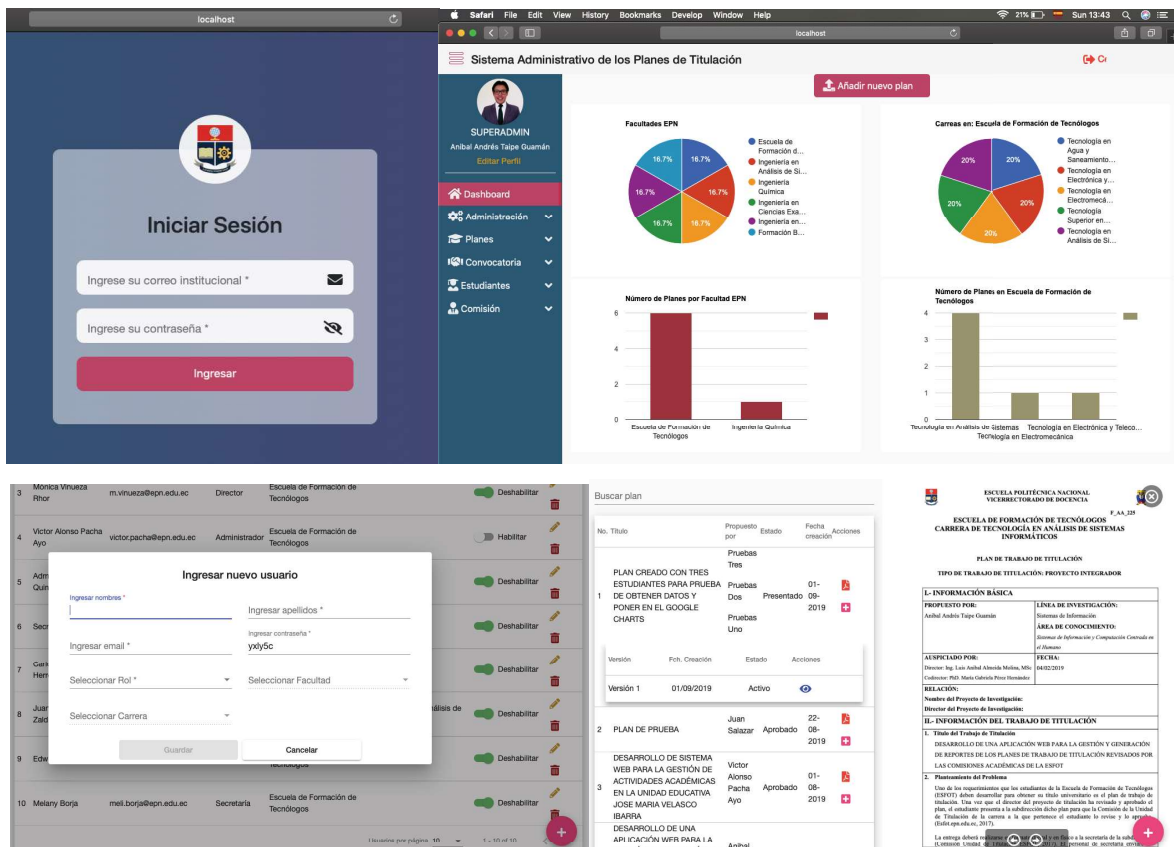


Figura 51 Pruebas en el navegador safari

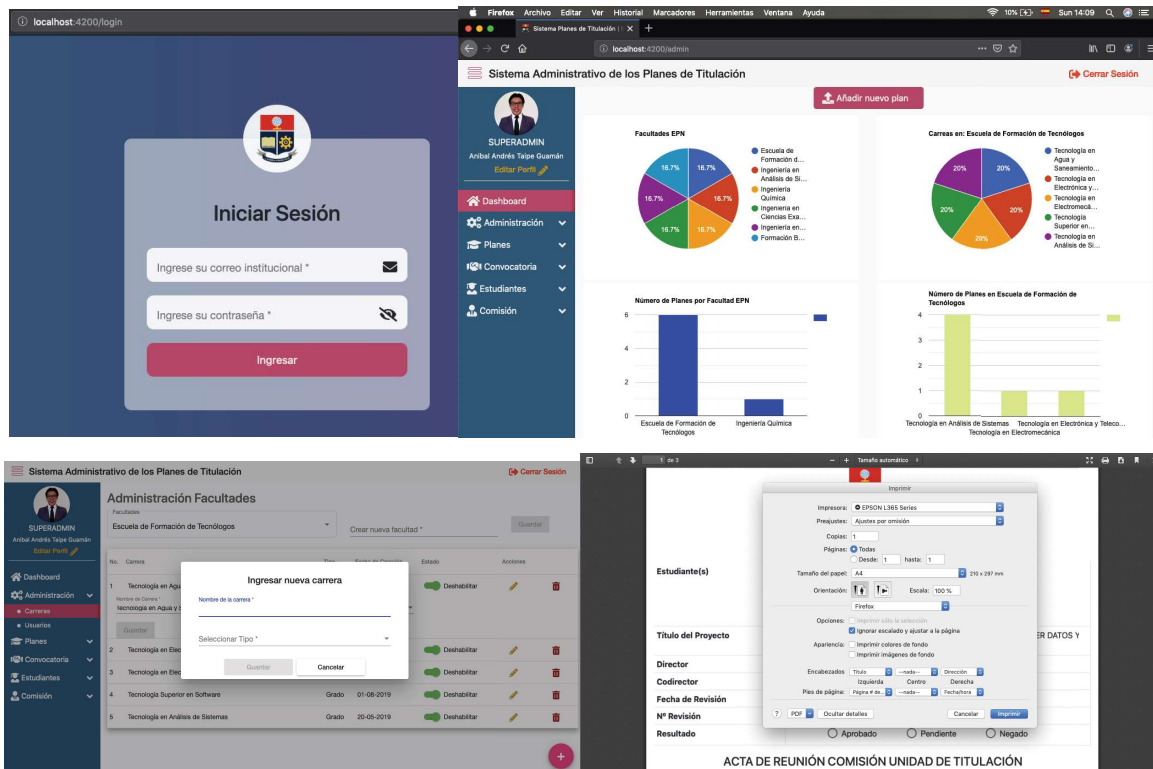


Figura 52 Pruebas en el navegador Firefox

Los resultados obtenidos y los parámetros que se tomaron en cuenta se los visualizan en la Tabla 10. Se califa sobre un valor de Error, Bueno, Muy bueno y Excelente.

Tabla 10 Resultados de las pruebas de compatibilidad.

Funcionalidad	Google Chrome	Safari	Firefox
Botones	Excelente	Excelente	Excelente
Tablas	Excelente	Excelente	Excelente
Modales	Excelente	Excelente	Excelente
Diseño	Excelente	Excelente	Excelente
Imágenes	Excelente	Excelente	Excelente
Notificaciones	Excelente	Excelente	Excelente
Animaciones	Excelente	Muy Bueno	Excelente

Elaborado por: Andrés Taipe

Pruebas de seguridad.

Para las pruebas de seguridad se realiza con una herramienta llamada Codacy (Codacy, 2019), que mediante nuestra integración del código a GitHub lo analiza y presenta los datos de varias pruebas, entre ellas las de seguridad. En la Figura 52 podemos ver la calificación de varias pruebas entre ellas de autenticación, DoS que es un ataque a un sistema de

computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos (Europol, 2018), acceso a archivos, almacenamiento inseguro e inyección de código SQL entre las mas importantes y todas con un resultado satisfactorio.

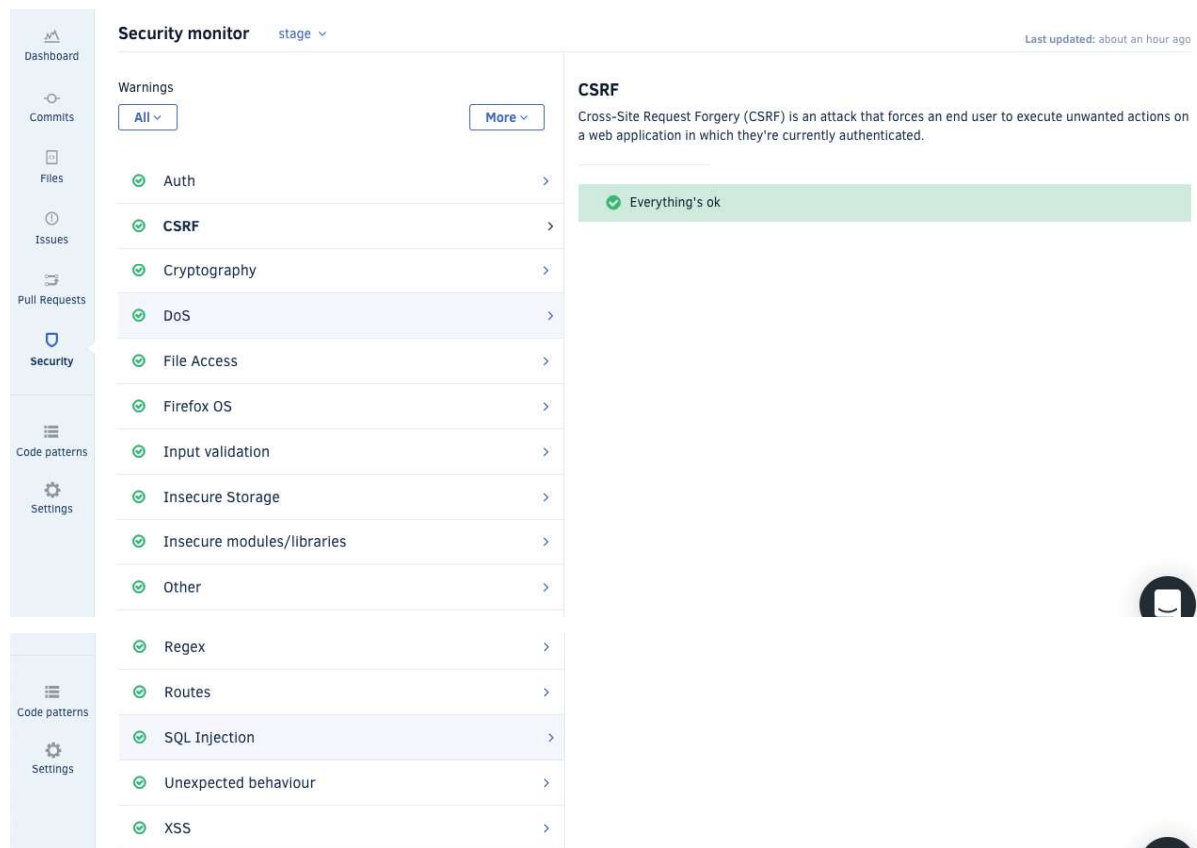


Figura 53 Pruebas de seguridad en Codacy

Pruebas unitarias.

Las pruebas unitarias ayudan a verificar el correcto funcionamiento por unidad de código y puede ser calificado en varios parámetros, en la herramienta Codacy (Codacy, 2019), brinda una página donde podemos visualizar nuestro código por archivos y ver la cantidad de problemas encontrados para ir resolviéndolos como se visualiza en la Figura 54.

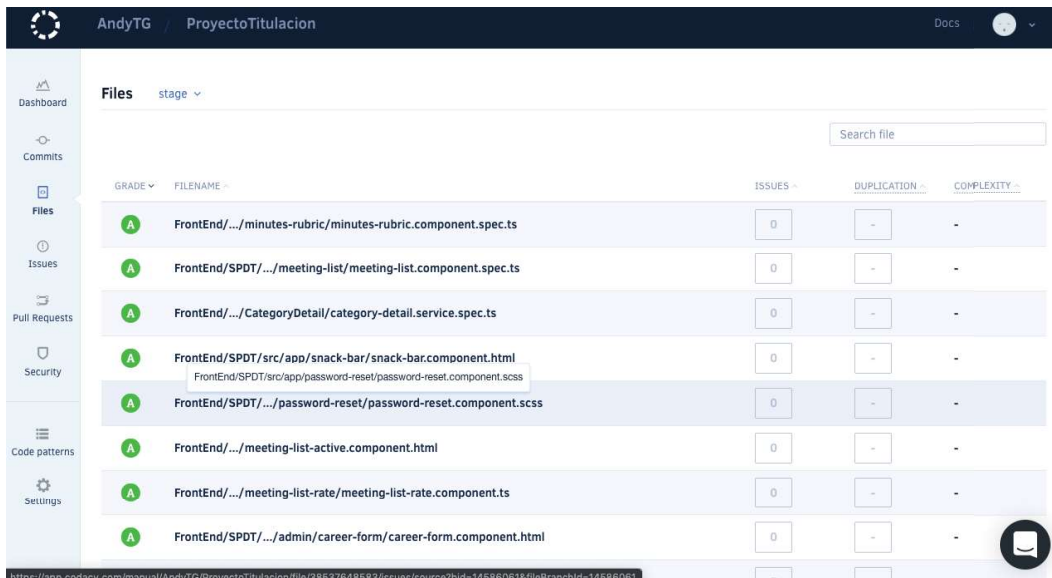


Figura 54 Pruebas unitarias en Codacy

Los problemas en Codacy se miden desde la A hasta la F, donde A son los archivos en con ningún problema y F los archivos con mayor problema. En base a esto se obtuvo los datos expuestos en la Tabla 11.

Tabla 11 Resultados de las pruebas unitarias.

Resultado	Archivos Analizados
A	177
B	0
C	4
D	29
E	2
F	40
Total	252

Elaborado por: Andrés Taipe

Pruebas de carga.

Para las pruebas de carga se toma como base que el sistema en un inicio va a ser utilizado por la ESFOT y que el límite de usuarios sean 20 por cada facultad. Esta información es indispensable para las solicitudes HTTP que se realizará a la ip del sistema para verificar como reacciona ante una cantidad determinada de peticiones.

Para esto utilizamos la herramienta de Apache Jmeter. Se crea un grupo de pruebas que como observamos en la Figura 55 se la llamó Prueba 1.

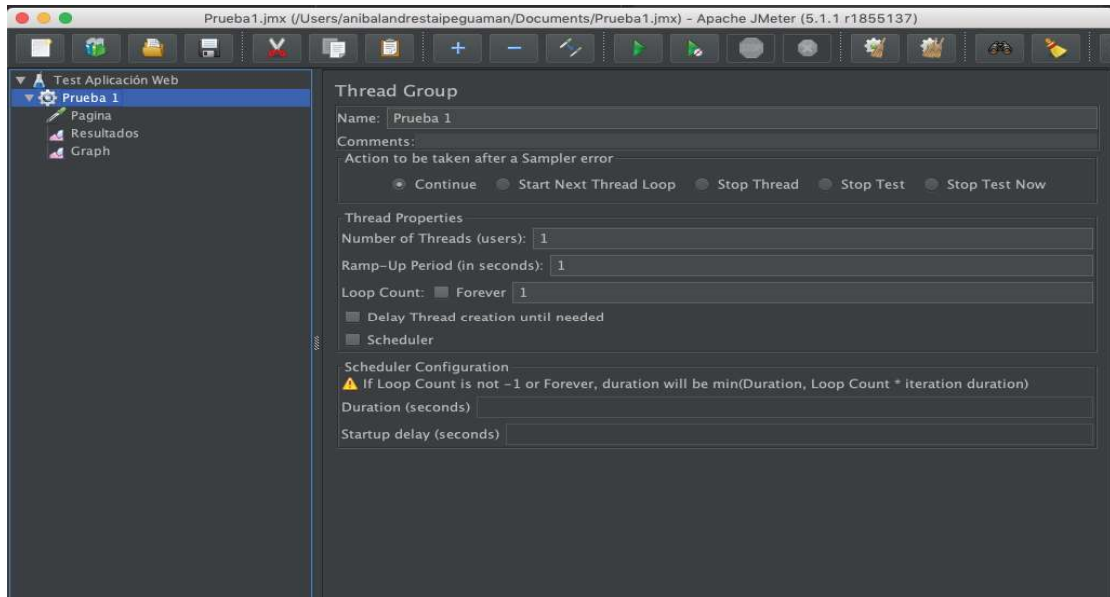


Figura 55 Jmeter grupo de pruebas

A continuación, se crea las solicitudes http para cada página que se necesita probar que al observar la Figura 56 fue denominada Página.

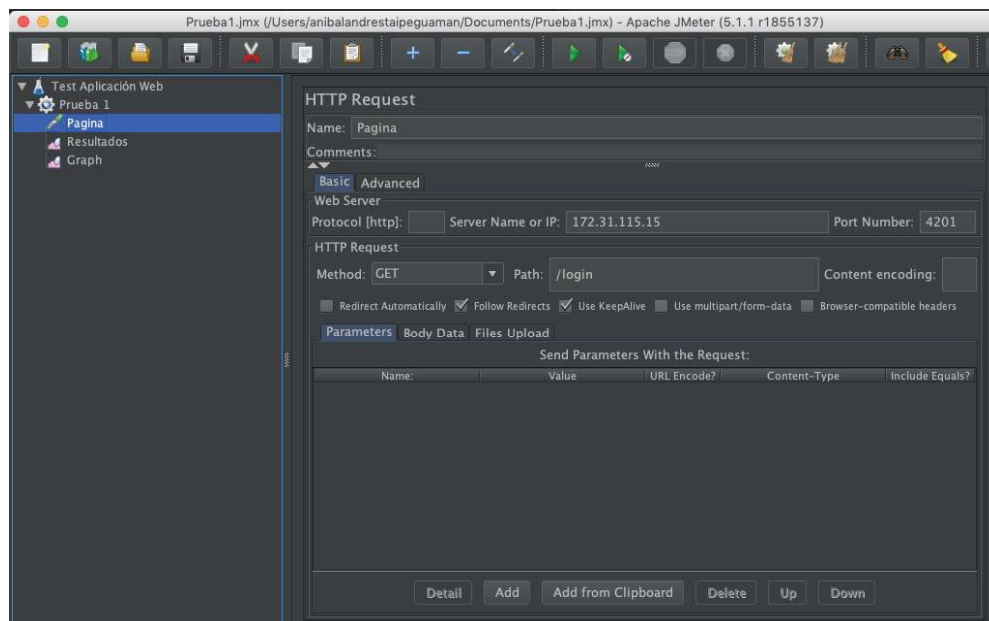


Figura 56 Jmeter http request

Elaborado por: Andrés Taípe

Al finalizar se crea un visualizador de resultados en el que se observa como reaccionaron a las peticiones que se realiza. El resultado obtenido a las peticiones realizadas fue satisfactorio como lo muestra la Figura 57.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	16:39:43.064	Prueba 1 1-1	HTTP Request	4	Success	434	134	4	1
2	16:39:53.667	Prueba 1 1-1	HTTP Request	5	Success	434	134	5	2
3	16:39:53.681	Prueba 1 1-2	HTTP Request	7	Success	434	134	7	1
4	16:39:53.689	Prueba 1 1-3	HTTP Request	6	Success	434	134	6	2
5	16:39:53.704	Prueba 1 1-4	HTTP Request	4	Success	434	134	4	1
6	16:39:53.711	Prueba 1 1-5	HTTP Request	4	Success	434	134	4	1
7	16:39:53.720	Prueba 1 1-6	HTTP Request	3	Success	434	134	3	1
8	16:39:53.732	Prueba 1 1-7	HTTP Request	4	Success	434	134	4	1
9	16:39:53.739	Prueba 1 1-8	HTTP Request	3	Success	434	134	3	1
10	16:39:53.750	Prueba 1 1-9	HTTP Request	12	Success	434	134	12	1
11	16:39:53.762	Prueba 1 1-10	HTTP Request	3	Success	434	134	3	1
12	16:39:53.771	Prueba 1 1-11	HTTP Request	3	Success	434	134	3	1
13	16:39:53.778	Prueba 1 1-12	HTTP Request	3	Success	434	134	3	1
14	16:39:53.787	Prueba 1 1-13	HTTP Request	3	Success	434	134	3	1

Figura 57 Jmeter resultados de la prueba

Pruebas de calidad.

Para esta prueba se utilizó una herramienta proporcionada por el navegador Chrome llamada Lighthouse la cual está diseñada para verificar el rendimiento, accesibilidad, buenas prácticas y SEO (*search engine optimization*). Las pruebas se realizó ya con el servidor en funcionamiento en la ESFOT como se muestra en la IP de la Figura 58.



Figura 58 Auditando la aplicación web

Como es una aplicación web que funcionará únicamente dentro de la red informática de la EPN el SEO que ayuda con el posicionamiento en buscadores no es importante por el momento aun así se obtuvo una calificación de 89/100. El rendimiento depende mucho del servidor que fue proporcionado por la ESFOT y es lo que se debe mejorar en calificación con una nota de 13/100. Las buenas prácticas de programación permiten que sea un sistema con un buen funcionamiento y ágil en todas las transacciones que realiza y se

obtuvo 79/100. En accesibilidad se obtuvo una calificación perfecta de 100/100 como los visualizamos en la Figura 59.

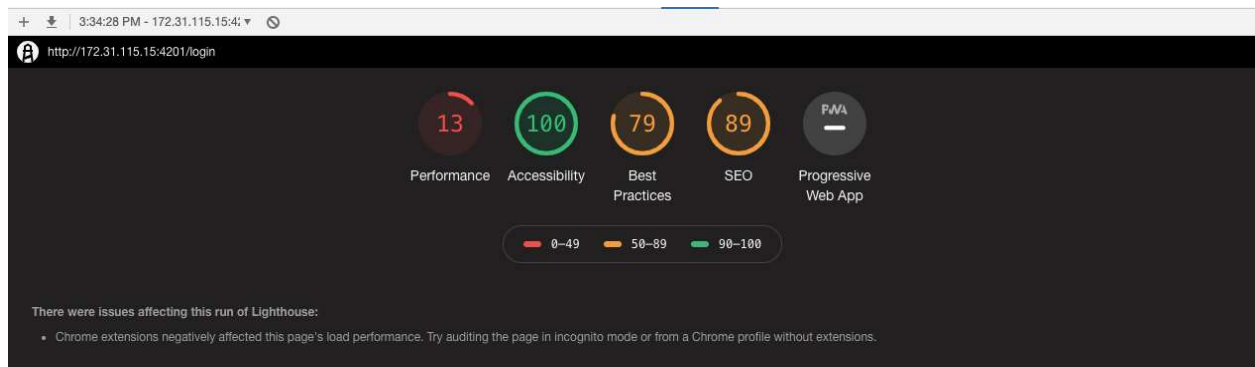


Figura 59 Resultado de las pruebas de calidad

Pruebas de usabilidad.

Para las pruebas de usabilidad se optó por usar el cuestionario SUS (*System Usability Scale*), que es una herramienta para la medición y evaluación de la usabilidad del software desarrollado, se trata de un cuestionario desarrollado en 1986 por Jonh Brooke (Castillo, 2015).

SUS tiene 10 preguntas, cada una con un valor de 1 a 5 según lo otorgue el usuario. Para las preguntas 1,3,5,7,9 se obtiene el SUS restando el valor obtenido menos 1, para las preguntas 2,4,6,8,10 se obtiene restando 5 menos el valor obtenido. A los resultados se multiplica por 2.5 para obtener el resultado final (Castillo, 2015). Las preguntas de SUS se presentan a continuación y los resultados obtenidos se los visualiza en la Tabla 12.

Preguntas SUS.

1. ¿Me gusta usar frecuentemente este software?
2. ¿Encuentro al software innecesariamente complejo?
3. ¿Pienso que es un software fácil de usar?
4. ¿Pienso que necesito el soporte de algún técnico para usar el software?
5. ¿Encontré varias funciones en el sistema que fueron bien integradas?
6. ¿Pienso que hay muchas inconsistencias en el software?
7. ¿Imagino que la mayoría de gente aprenderá a utilizar el software rápidamente?
8. ¿Me pareció un sistema muy complicado de usar?
9. ¿Me siento muy seguro al usar el sistema?
10. ¿Necesito aprender bastantes cosas antes de manejar el software?

Tabla 12 Matriz de evaluación de resultados SUS.

Pregunta	Usuario 1	Cálculo	Usuario 2	Cálculo	Usuario 3	Cálculo
1	4	3	3	2	4	3
2	2	3	2	3	1	4
3	5	4	5	4	4	3
4	3	2	3	2	4	1
5	5	4	4	3	3	2
6	3	2	2	3	1	4
7	5	4	5	4	5	4
8	2	3	1	4	1	4
9	4	3	5	4	5	4
10	3	2	3	2	2	3
SUS		30		31		32
% satisfacción		75%		77,5		80%

Elaborado por: Andrés Taipe

Para finalizar, se entrega el manual de usuario que se encuentra en el anexo 3 para más información del uso según el rol que cumplirán los usuarios.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En la actualidad, existen muchas alternativas para crear aplicaciones web, podemos encontrar aplicaciones administrativas por las cuales pagamos una cantidad de dinero e intentamos adaptarlas a nuestras necesidades, sin embargo, no todos los requerimientos que tienen los clientes son satisfechos.

Se desarrollo una aplicación web personalizada con las necesidades de la ESFOT, analizando sus requerimientos, pero con la capacidad de funcionar en otras facultades dentro de la EPN. No es solo una aplicación web administrativa, además, ayuda en el ahorro de tiempo a los estudiantes que ya no necesitan ir hasta la facultad para preguntar por los resultados de la calificación de su plan de titulación, toda la información le llega directamente a su correo electrónico

Para los usuarios de la aplicación web presenta un diseño estéticamente agradable, sencillo y que la información a ingresar sea únicamente la necesaria para el correcto funcionamiento de la aplicación

Evita el desperdicio de papel. Toda la información almacenada se realiza de forma digital, creando una bitácora de consulta de datos mediante los buscadores creados en cada una de las páginas web.

El código de desarrollo utiliza buenas prácticas de programación como se lo visualiza en las pruebas realizadas, lo que ayuda a una fácil comprensión de las líneas de código y es escalable, además, al utilizar la metodología Scrum, permitió una planificación y manejo de los tiempos de entrega del sistema, además de ayudarnos a mantener una comunicación continua con el cliente. El uso de esta metodología suele generar dudas por el tiempo que se toma en reuniones, sin embargo, para este proyecto fue indispensable toda la información proporcionada por el cliente en las reuniones sobre los procesos que realiza la EPN, para implementarlo en el sistema de acuerdo con las normas establecidas en la institución.

El sistema está desarrollado pensando en la seguridad de la información, es por eso por lo que los datos son guardados únicamente en el servidor que está implementado dentro de la ESFOT con una IP estática mediante la cual pueden hacer el ingreso los miembros que obtengan la autorización, únicamente dentro de la red informática de la EPN.

4.2 Recomendaciones

Para un mejor entendimiento del sistema se puede verificar el manual de usuario que se encuentra en el Anexo 3 de este documento el cual presenta cada una de las características de acuerdo con los roles de cada usuario.

El usuario administrador de cada facultad debe tener un vasto conocimiento del sistema ya que de él parte todo el flujo de los procesos a realizar por parte de sistema.

Al funcionar en un servidor interno de la ESFOT es asegurar que se encuentre siempre activo para que los usuarios puedan ingresar al sistema sin ningún inconveniente, además de asegurar la protección de este ya que en él se encuentra toda la información almacenada de los proyectos de titulación.

La aplicación web es escalable, lo que se recomienda usar como base para desarrollar módulos para creación de respaldos de la información, y en un futuro tener todo el proceso

de la unidad de titulación automatizado, desde los planes de titulación hasta la defensa de la tesis incorporado a este sistema.

Se dedicó al desarrollo de este sistema más tiempo del planificado para poder utilizar nuevas herramientas como Angular y Node.js que no fueron impartidos en clase pero que están siendo utilizadas en el mundo actual del desarrollo de software.

5. REFERENCIAS BIBLIOGRÁFICAS

Bahit, E. (2012). *Scrum & extreme Programming*. Buenos Aires: safecreative.

Comisión Unidad de Titulación ESFOT. (Marzo de 2017). *GUÍA PARA LA ELABORACIÓN DEL PLAN DE PROYECTO DE TITULACIÓN*. Recuperado el Julio de 2019, de <https://esfot.epn.edu.ec/index.php/component/jdownloads/send/6-esfot/375-gui-elaboracion-reforma>

Scrum Study. (2017). Obtenido de <https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-3rd-edition.pdf>

NodeJs. (Julio de 2019). *Acerca de Node.js*. Obtenido de <https://nodejs.org>

Orós, J. (2014). *Guía práctica de XHTML, JavaScript y CSS*. Madrid: Alfaomega Group Editor.

Pérez, E., & Altadill, P. (2016). *Diseño de Interfaces Web*. Madrid: IBERGACETA PUBLICACIONES, S.L.

PostgreSQL. (Julio de 2019). Obtenido de PostgreSQL: The World's Most Advanced Open Source Relational Database: <https://www.postgresql.org/>

Sass. (Julio de 2019). Obtenido de CSS with superpowers: <https://sass-lang.com/>

Castillo, P. S. (Junio de 2015). *Desarrollo de una Plataforma Web para el Sistema de Gestión de la Información de Proyectos de Fiscalización realizados por la empresa*

- TECNIE, Accesible Local y Remotamente.* Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/10786/1/CD-6318.pdf>
- Chapaval, N. (2017). *Qué es Frontend y Backend.* Obtenido de <https://platzi.com/blog/ques-frontend-y-backend/>
- Codacy. (2019). *We're on a mission to help developers ship better code, faster.* Obtenido de <https://www.codacy.com/about>
- Europol. (25 de Abril de 2018). *How can DDoS attacks paralyse the internet.* Obtenido de https://web.archive.org/web/20181221113146/https://www.europol.europa.eu/sites/default/files/documents/ddos_infographicv3_copy.pdf
- Gallego, M. T. (2015). *Metodología Scrum.* Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612 memoria.pdf>
- MDN web docs. (23 de Marzo de 2019). *CRUD.* Obtenido de <https://developer.mozilla.org/es/docs/Glossary/CRUD>
- Ministerio del Trabajo del Ecuador. (2019). *Ministerio del Trabajo. Biblioteca.* Obtenido de <http://www.trabajo.gob.ec/biblioteca/>
- Trinidad, P. A. (2015). *Guía de estilo para diseñar y desarrollar una API.* Obtenido de <https://platzi.com/blog/como-crear-apis/>
- Villanueva, L. M. (s.f.). *GUÍA de POSICIONAMIENTO WEB: SEO On y Off Pag.* Recuperado el 07 de 2019, de <https://luismvillanueva.com/wp-content/uploads/2014/11/ebook-posicionamiento-seo-lmv.pdf>