

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA WEB DE RETROALIMENTACIÓN CONTINUA PARA LOS DOCENTES DE LA ESFOT

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

SEBASTIÁN ALEJANDRO MORALES DURÁN

sebastian.morales@epn.edu.ec

XAVIER GONZALO CALLE AVILA

xavier.calle@epn.edu.ec

DIRECTOR: MSC. JUAN PABLO ZALDUMBIDE PROAÑO

juan.zaldumbide@epn.edu.ec

CODIRECTORA: MSC. LUZ MARINA VINTIMILLA JARAMILLO

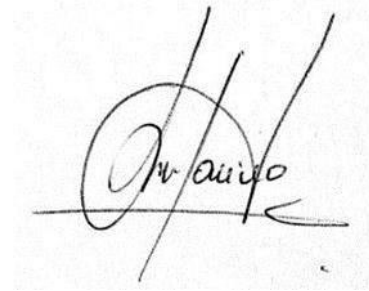
marina.vintimilla@epn.edu.ec

Quito, enero 2022

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Calle Avila Xavier Gonzalo y el Sr. Morales Durán Sebastián Alejandro como requerimiento parcial a la obtención del título de TECNÓLOGO EN ANÁLISIS DE SISTEMA

S INFORMÁTICOS, bajo nuestra supervisión:



**Juan Pablo Zaldumbide
Proaño**
DIRECTOR DEL PROYECTO

Luz Marina Vintimilla Jaramillo
CODIRECTORA DEL PROYECTO

DECLARACIÓN

Nosotros Calle Avila Xavier Gonzalo con CI: 1727337501 y Morales Durán Sebastián Alejandro con CI: 1718976010 declaramos bajo juramento que el trabajo aquí descrito es nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

DEDICATORIA

Dedico este proyecto a las personas que confiaron en mí y me apoyaron durante todo el proceso, especialmente a mi madre y a mi hermana quienes siempre han estado pendiente del desarrollo de mi proyecto, motivándome y dándome ánimos para continuar.

Sebastián Alejandro Morales Durán

AGRADECIMIENTO

Quiero agradecer a mi madre Alexandra y a mi hermana Cristina por estar siempre motivándome en mi ámbito laboral y en el académico, por estar pendiente de mí y de mi crecimiento como persona. Agradecer también a mis profesores que me han enseñado como ser un buen profesional y que además han solventado todas mis dudas incluso en el ámbito no académico.

Gracias a mis amigos quienes han estado pendientes y supieron dar esas palabras de aliento en momentos clave para no desfallecer y seguir adelante.

Sebastián Alejandro Morales Durán

DEDICATORIA

El presente trabajo está dedicado a mis padres por enseñarme que la mejor forma de aprender es hacerlo uno mismo y que no importa cuán grande sea un trabajo, se puede lograr paso a paso. También está dedicado a todas las personas que han sido mi apoyo a lo largo de mi vida.

Xavier Gonzalo Calle Avila

AGRADECIMIENTO

Quiero expresar mi gratitud a Dios que me ha colmado de bendiciones y buenas personas en mi vida.

Agradezco a mis padres por ser mi ejemplo y haberme apoyado con amor y paciencia durante toda mi carrera universitaria.

Finalmente quiero agradecer a la Escuela Politécnica Nacional, autoridades, profesores y compañeros que han estado presentes en mi vida educativa y personal.

Xavier Gonzalo Calle Avila

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Objetivo general	2
1.2	Objetivos específicos.....	2
1.3	Alcance.....	2
2	Metodología.....	4
2.1	Metodología de Desarrollo.....	4
2.1.1	Roles	4
2.1.2	Artefactos	5
2.2	Diseño de interfaces (<i>mockups</i>).....	7
2.2.1	Herramienta utilizada para el diseño.....	7
2.2.2	Sistema Web	8
2.3	Diseño de la arquitectura	8
2.3.1	Patrón arquitectónico.....	8
2.3.2	Sistema Web	9
2.4	Herramientas de desarrollo.....	10
2.4.1	Sistema Web	10
3	Resultados y Discusión.....	12
3.1	<i>Sprint 0</i> . Configuraciones iniciales	12
3.1.1	Creación de mockups.....	12
3.1.2	Diseño de base de datos.....	14
3.2	<i>Sprint 1</i> . Creación de <i>EndPoints</i> , relaciones y recursos.....	14
3.2.1	Creación de relaciones.....	14
3.2.2	Creación de <i>EndPoints</i>	15
3.2.3	Creación de recursos.....	15
3.2.4	Creación de seeders.....	15
3.3	<i>Sprint 2</i> . Diseño de páginas principales	16
3.3.1	Implementación de JWT para autenticación de usuarios.....	16

3.3.2	Conexión entre el backend y el frontend.....	17
3.3.3	Diseño de página principal.....	17
3.3.4	Implementación de rutas de navegación.....	17
3.3.5	Diseño de página de inicio de sesión.....	18
3.3.6	Diseño de página de perfil.	18
3.4	<i>Sprint 3. Roles y autorizaciones</i>	19
3.4.1	Implementación de roles.....	19
3.4.2	Implementación de autorización de usuarios.	19
3.5	<i>Sprint 4. Creación de encuestas</i>	19
3.5.1	Creación de políticas de encuestas.	19
3.5.2	Diseño de página de encuestas (Creación de encuestas).	19
3.5.3	Diseño de página de encuestas (Envío de respuestas a las encuestas). 20	
3.6	<i>Sprint 5. Visualización de encuestas</i>	20
3.6.1	Diseño de página de encuestas (Visualización de encuestas).....	20
3.6.2	Diseño de tablero de visualización de encuestas para docentes.	21
3.6.3	Comparación de resultados de encuestas.	22
3.7	<i>Sprint 6. Despliegue del sistema web y pruebas</i>	23
3.7.1	Despliegue del sistema web	23
3.7.2	Pruebas de funcionalidad, de carga y compatibilidad	25
4	Conclusiones y Recomendaciones	27
4.1	Conclusiones	27
4.2	Recomendaciones	28
5	Referencias Bibliográficas	29
6	ANEXOS.....	i
6.1	Manual Técnico	i
6.2	Manual de Usuario.....	i
6.3	Manual de Instalación.....	i

ÍNDICE DE FIGURAS

Fig. 1: Pagina inicial del Sistema Web	8
Fig. 2: Patrón arquitectónico del sistema web	10
Fig. 3: Mockup: Módulo inicio de sesión	13
Fig. 4: Mockup: Perfil.....	13
Fig. 5: Diseño de base de datos	14
Fig. 6: Relación materia – usuario	14
Fig. 7: Endpoints para materias.....	15
Fig. 8: Recurso para capítulos.....	15
Fig. 9: Seeder de respuestas	16
Fig. 10: JWTMiddleware.....	16
Fig. 11: Configuración de API en frontend.....	17
Fig. 12: Página principal.....	17
Fig. 13: Rutas de navegación.....	17
Fig. 14: Módulo inicio de sesión	18
Fig. 15: Información de perfiles	18
Fig. 16: Roles y autorizaciones	19
Fig. 17: Políticas de preguntas	19
Fig. 18: Creación de encuestas.....	20
Fig. 19: Repuesta de encuestas.....	20
Fig. 20: Visualización de encuestas (estudiante).....	21
Fig. 21: Visualización de encuestas (docente).....	21
Fig. 22: Detalles de encuestas (docente)	22
Fig. 23: Visualización de encuestas a comparar.....	22
Fig. 24: Visualización de resultados de encuestas comparadas.....	23
Fig. 25: Despliegue del sistema web en Heroku.....	24
Fig. 26: Verificación del despliegue en Vercel	24
Fig. 27: Verificación de subida de base de datos	25
Fig. 28: Ejemplo prueba de carga.....	26

ÍNDICE DE TABLAS

TABLA I: Equipo de trabajo y sus roles.....	5
TABLA II: Estructura de historia de usuario	7
TABLA III: Herramientas para el desarrollo del Sistema Web	10
TABLA IV: Formato pruebas funcionales	25

RESUMEN

El presente trabajo surge de la necesidad de una retroalimentación continua a docentes de la Escuela de Formación de Tecnólogos de la Escuela Politécnica Nacional (EPN); por esta razón se ha implementado un Sistema Web para que los docentes puedan generar encuestas de sus materias, organizado por capítulos, dirigido a los estudiantes. Los resultados se pueden visualizar y se comparan entre diferentes materias.

El Sistema Web resuelve la problemática de que los estudiantes no se expresan y, por lo tanto, los docentes no conocen oportunamente y con exactitud cuál es la situación académica de los estudiantes con respecto a la materia.

Este trabajo se ha desarrollado con *Laravel* que es un *Framework* de *PHP*, *React* que es una librería de *JavaScript* y como Sistema Gestor de Base de Datos (SGBD) *MySQL*, gracias a su gratuidad, su escalabilidad y rendimiento. El proyecto se ha basado en el marco de trabajo ágil *Scrum*, aplicando varios de sus beneficios, algunos de ellos son: interacción directa con el usuario final, adecuada recopilación de requerimientos, definición de roles del equipo, priorización de tareas, respuesta al cambio, entre otros; permitiendo que el proyecto se realice en el tiempo establecido. Por último, en este trabajo se muestran las tareas y los resultados obtenidos por cada *sprint* y el despliegue del sistema web a producción.

PALABRAS CLAVE: Sistema Web, Scrum, Laravel, Sprint, MySQL

ABSTRACT

The present work arises from the need for continuous feedback to teachers at the Technologists Training School of the National Polytechnic School (EPN); For this reason, a Web System has been implemented so that teachers can generate surveys of their subjects, organized by chapters, aimed at students. The results can be visualized and compared with respect to other subjects.

The Web System solves the problem that students do not express themselves and, therefore, teachers do not know in a timely manner and exactly what the academic situation of the students is with respect to the subject.

This work has been developed with Laravel, which is a PHP Framework, React, which is a JavaScript library, and MySQL as a Database Management System (SGBD), thanks to its free nature, scalability, and performance. The project has been based on the Scrum agile framework, applying several of its benefits, some of them are direct interaction with the end user, adequate collection of requirements, definition of team roles, prioritization of tasks, response to change, among others; allowing the project to be carried out in the established time. Finally, this work shows the tasks and the results obtained by each sprint and the deployment of the web system to production.

This work has been done with Laravel, which is a PHP Framework and the MySQL Database Management System (DBMS), thanks to its scalability and performance. During the development of the project, it has been based on the agile Scrum framework, which has several benefits when it is implemented in the project, some of them are direct interaction with the end user, adequate compilation of requirements, definition of team roles, prioritization of tasks, response to change, among others. Allowing the project to be carried out in a set time without any mishaps. Finally, this work shows the tasks and the results obtained by each sprint and the deployment of the system to production.

KEYWORDS: Sistema Web, Scrum, Laravel, Sprint, MySQL

1 INTRODUCCIÓN

La Escuela de Formación de Tecnólogos (ESFOT) de la Escuela Politécnica Nacional (EPN) tiene como misión preparar profesionales con una sólida formación académica, humanística, conceptual y práctica, cuya interacción con el medio social procure un aporte al desarrollo tecnológico del país, para viabilizar el proceso de apropiación tecnológica y transferirlo al medio productivo cuidando siempre el ecosistema [1].

Actualmente, como en todas las Facultades de la EPN, su trabajo se circunscribe y se adapta a los nuevos reglamentos, normativas y directrices que proveen los organismos de educación superior del estado y de los órganos colegiados de la EPN [1]. Por ende, los docentes también reciben una heteroevaluación por parte de los estudiantes cuyo objetivo es “Conocer el grado de satisfacción de los estudiantes, con la formación que están recibiendo en la EPN” [1]. Esta evaluación se realiza a través del Sistema Integrador de Información (SII) Académico para todos los docentes de las materias en las que está inscrito el estudiante [2].

Los resultados de la heteroevaluación son obtenidos al finalizar el semestre, lo que implica que los docentes reciben una retroalimentación acumulada al finalizar un periodo académico [2], esto impide que puedan tomar acciones durante el semestre para poder mejorar de manera inmediata, por el contrario, los docentes aplican mejoras para el siguiente semestre con otros estudiantes. El estudio realizado por Guerrero Radillo [3] menciona la importancia de recibir retroalimentación para obtener mejores resultados, sin embargo, destaca los beneficios de recibir una retroalimentación continua. Los resultados de este estudio pueden ser aplicados para realizar una retroalimentación continua a los docentes de la ESFOT por parte de los estudiantes que permita ejecutar acciones correctivas durante el semestre en curso.

Por ende, la solución consiste en crear un sistema web donde los estudiantes brindan retroalimentación continua a sus docentes sobre los temas de la materia tratados en clase, es decir, el docente puede conocer el nivel de entendimiento, los temas que deben ser reforzados y la satisfacción de sus estudiantes acerca de la metodología utilizada y tomar acciones correctivas de manera inmediata. Esta encuesta está conformada por preguntas genéricas que se reutilizan por parte de todos los docentes en diferentes materias. Los resultados parciales de estas encuestas pueden ser visualizados únicamente por el docente a cargo de la materia y se presenta de una manera global al final del semestre, lo que le permite conocer como ha sido su desempeño en diferentes aspectos de la materia durante el transcurso del periodo académico.

1.1 Objetivo general

Desarrollar un sistema web de retroalimentación continua para los docentes de la ESFOT

1.2 Objetivos específicos

- Determinar los requerimientos del sistema web.
- Diseñar el flujo del sistema, la base de datos y las interfaces de usuario.
- Desarrollar el sistema web en base a los requerimientos y diseños establecidos.
- Realizar pruebas de aceptación y usabilidad del sistema.
- Implementar el sistema web en un servidor de acceso para docentes y estudiantes.

1.3 Alcance

Debido a la necesidad de mejorar el sistema de evaluación actual se ha desarrollado un sistema web que permite a los estudiantes brindar una retroalimentación adecuada a los docentes sobre aspectos importantes de las materias.

El sistema web se ha desarrollado para su uso dentro de la Escuela de Formación de Tecnólogos tanto de docentes como de estudiantes. Ambos hacen uso del sistema web para generar y responder encuestas correspondientemente.

Los estudiantes pueden y deben realizar las encuestas creadas por los docentes dentro del sistema web para cada una de las materias y capítulos; las encuestas deben llenarse en plenitud de cada una de sus preguntas, adicionalmente constan de una pregunta opcional que el estudiante puede omitir.

Los docentes pueden crear encuestas dentro del sistema web en base a la materia que dictan durante el semestre y a los capítulos de estas materias; pueden plantear un objetivo, para de esta manera tener una visión clara de cada encuesta creada. Adicionalmente, los docentes podrán ver un acumulado de datos ordenados de manera que puedan medir su desempeño en la materia durante el transcurso del periodo académico.

- El sistema web les permite a los docentes:
- Iniciar sesión dentro del sistema.
- Visualizar información personal.
- Crear encuestas.

- Visualizar resultados de encuestas de los estudiantes.

El sistema web les permite a los estudiantes:

- Iniciar sesión dentro del sistema.
- Visualizar información personal.
- Visualizar encuestas pendientes y realizadas.
- Realizar encuestas.
- Visualizar resultados de encuestas realizadas por el estudiante.

2 METODOLOGÍA

Dentro de la metodología utilizada encontramos lo siguiente:

2.1 Metodología de Desarrollo

En este proyecto se ha trabajado bajo la metodología ágil *Scrum* la misma que trae varios beneficios en el desarrollo. Primeramente, *Scrum* ha facilitado el dimensionamiento del proyecto mediante la estimación de dificultad y por ende tiempo que una tarea puede tomar, por consiguiente, permite calcular una fecha estimada bastante acertada en la que el proyecto es entregado. El tener ciclos cortos de entrega, ha ayudado a recibir retroalimentación de manera más consecutiva lo que permite tomar acciones en el momento adecuado, mencionando también que, al ser una metodología ágil, posee la capacidad de responder a grandes cambios sin inconvenientes [4].

El equipo de trabajo ha estado en constante comunicación para que todos conozcan el estado de las tareas y puedan aportar si tienen conocimiento al respecto. Cabe mencionar que también el proyecto se basa en el manifiesto ágil por lo que se ha enfocado en la entrega funcional del software y además la agilidad que puede tener el proyecto frente a distintas circunstancias [4] [5].

2.1.1 Roles

Las personas que son partícipes del proyecto y se encuentran en el proceso de *Scrum* [4]:

Product Owner:

Es la persona que conoce el negocio, al cliente y el producto que se está desarrollando, por ende, es el encargado de la toma de decisiones del proyecto, priorizando las tareas principales [4]. Este rol ha sido representado por el director del proyecto, el mismo que ha proporcionado al equipo de desarrollo todos los requerimientos y la información requerida para el desarrollo del proyecto. La ESFOT de la EPN tiene como misión preparar profesionales con una sólida formación académica, humanística, conceptual y práctica, cuya interacción con el medio social procure un aporte al desarrollo tecnológico del país, para viabilizar el proceso de apropiación tecnológica y transferirlo al medio productivo cuidando siempre el ecosistema [1].

Scrum Master

Es el encargado de liderar el proyecto, es decir se asegura que no exista inconvenientes en el proceso de desarrollo [4]. Este rol ha sido empleado por el director del proyecto, el

mismo que posee abundantes conocimientos para poder ayudar al equipo de desarrollo en sus objetivos.

Development Team

El equipo de desarrollo suele estar conformado de 5 a 9 personas, quienes están autorizados para tomar decisiones y organizarse para cumplir sus objetivos [4]. El equipo de desarrollo ha sido conformado únicamente por 2, representados por los estudiantes encargados del proyecto.

A continuación, en la **TABLA I** se muestra un resumen de las personas y los roles que han representado dentro del proyecto.

TABLA I: Equipo de trabajo y sus roles

NOMBRE	ROL
Ing. Edwin Salvador, MSc.	<i>Product Owner</i>
Ing. Juan Pablo Zaldumbide	<i>Scrum Master</i>
Xavier Calle Sebastián Morales	<i>Development Team</i>

Las personas que no son partícipes dentro de *Scrum*, pero son fundamentales para la planeación de cada *sprint* [4]:

Usuarios:

Son las personas a quienes está dirigido el producto [4], en este caso son los estudiantes y docentes.

Stakeholders:

Son las personas a quienes beneficia el proyecto de desarrollo [4], en este caso el principal *stakeholder* es el departamento de mejora continua de la ESFOT.

2.1.2 Artefactos

Dentro de los artefactos utilizados encontramos:

Recopilación de Requerimientos

La recopilación de requerimientos funcionales y no funcionales se ha realizado en base a un análisis del equipo de desarrollo acerca de cómo debe funcionar el sistema web y que se requiere para su correcta implementación.

Requisitos funcionales

- El sistema web debe permitir ingresar a un usuario a la plataforma.
- El sistema web debe diferenciar entre tipos de usuarios.
- El sistema web debe permitir a cualquier usuario ver su perfil.
- El sistema web debe permitir visualizar las materias en las que está inscrito un usuario.
- El sistema web debe permitir a un usuario docente crear una encuesta.
- El sistema web debe permitir a un usuario docente visualizar un reporte de las encuestas respondidas.
- El sistema web debe permitir a un usuario estudiante visualizar las encuestas que no ha realizado.
- El sistema web debe permitir a un usuario estudiante responder una encuesta
- El sistema debe web permitir a un usuario estudiante visualizar las encuestas que ha respondido.

Requisitos no funcionales

- El sistema debe permitir a un estudiante visualizar en su perfil su nombre y correo.
- El sistema web debe ser capaz de procesar N resultados por segundo.
- El tiempo de aprendizaje del sistema web por un usuario deberá ser menor a 2 horas.
- El sistema web debe contar con manuales de usuario estructurados adecuadamente.
- El sistema web debe proporcionar mensajes de error que sean informativos y orientados a los usuarios.
- El sistema web debe poseer un diseño *Responsive*.
- El sistema web debe asegurar que los datos estén protegidos del acceso no autorizado

Historias de Usuario

En base a los requerimientos se han planteado historias de usuario que describen de manera clarificada la funcionalidad del sistema web.

Las historias de usuario están estructuradas en base al formato que se presenta en la **TABLA II**.

TABLA II: Estructura de historia de usuario

Historia de Usuario	
Número: (Número de historia de usuario).	
Nombre de Historia: (Nombre de la historia de usuario).	
Prioridad en Negocio: (Impacto en el negocio).	Usuario: (Usuario al que se hace referencia en la historia).
Puntos estimados: (Estimación de dificultad de desarrollo).	Riesgo en desarrollo: (Potencial riesgo al momento de desarrollar).
Programador Responsable: (Persona responsable de desarrollar la historia de usuario).	
Descripción: (Descripción de la historia de usuario).	

Las historias de usuario se encuentran redactadas en los anexos (Manual técnico, sección 3. Historias de usuario página 6).

Product Backlog

El producto *backlog* consta de todos los requisitos que tiene el sistema web y está ordenado de forma prioritaria.

El producto *backlog* completo se encuentra en los anexos (Manual técnico, sección 4. *Product Backlog* página 13).

2.2 Diseño de interfaces (*mockups*)

El diseño de las interfaces ha sido realizado utilizando *mockups* como se describe a continuación.

2.2.1 Herramienta utilizada para el diseño

Inicialmente se ha utilizado la herramienta *Marvel* donde se ha realizado un prototipado básico, posterior a ello se ha utilizado *Figma* ya que se puede ha integrado con la librería de *UI Ant Design* que se ha utilizado dentro del proyecto, facilitando así tener *mockups* mucho más fáciles de desarrollar porque ya se sabía que componentes utilizar y el fragmento de código de estos.

Para el desarrollo de los prototipos se ha tenido en cuenta y sigue los principios Heurísticos de Jakob Nielsen, los mismos que favorecen a que un usuario se entienda con el sistema para lograr un mismo objetivo, por otro lado, guía al diseñador en todo el

proceso, asiste a los evaluadores en la identificación de problemas con las interfaces de usuario, verificando que se respeten las reglas de usabilidad, por último proveen la pauta adecuada para conocer el por qué los usuarios cometen determinados errores [6].

2.2.2 Sistema Web

La **Fig. 1** muestra el prototipo de la página de inicio del sistema web, mientras que el prototipado de las demás páginas se encuentran en los anexos (Manual técnico, sección 6. Diseño de interfaces, página 16).



Fig. 1: Pagina inicial del Sistema Web

2.3 Diseño de la arquitectura

Posteriormente de haber realizado los *Sprints* que tiene el proyecto y a su vez el diseño de interfaces del Sistema Web, a continuación, se describe el patrón de arquitectura que se ha utilizado para el desarrollo.

2.3.1 Patrón arquitectónico

En este caso se ha utilizado el patrón de arquitectura Modelo Vista Controlador (MVC) tal como se muestra en la columna intermedia de la **Fig. 2** [7]. Primeramente, hay que tener claro lo que realiza cada una de las partes [7] :

- El modelo es el que representa los datos del programa, es el encargado del manejo de datos y de su transformación.
- La vista es, como su nombre lo indica, una representación visual de los datos que contiene el modelo, con el objetivo de mostrar los datos al usuario.
- El controlador es el encargado de dar significado a las peticiones del usuario, actuando sobre los datos proporcionados por el modelo. Es el intermediario entre la vista y el modelo, por esta razón cuando se realiza una modificación en el modelo o en la vista, el controlador actúa.

2.3.2 Sistema Web

La arquitectura completa del sistema web se basa en tres columnas, la primera representa la interfaz de usuario, la segunda que representa la lógica de la aplicación y por último la tercera columna que representa la base de datos que se utiliza.

Para el desarrollo de la interfaz de usuario se ha utilizado *ReactJs* que es una librería de *JavaScript* para poder construir interfaces de usuario [8], una vez desarrollada la interfaz de usuario, esta se despliega en *Vercel* que es una plataforma que permite alojar sitios web estáticos como la interfaz de usuario de una aplicación [9], por consiguiente, se puede acceder al sitio web desde cualquier navegador. Para el desarrollo de la lógica de la aplicación se ha utilizado *Laravel* que es un marco de trabajo basado en *PHP* donde se crea el *API* que interactúa entre la base de datos y el usuario final [10], es en este punto donde se utiliza el patrón arquitectónico MVC, la lógica de la aplicación se ha desplegado en la plataforma *Heroku* que permite alojar toda la lógica de la aplicación trabajando como un *PaaS* [11]. Por último, para el desarrollo de la base de datos se utiliza *MySQL* que es una base de datos relacional [12], la misma que se ha desplegado en *Alwaysdata* que es una plataforma que permite alojar los datos del sistema web [13].

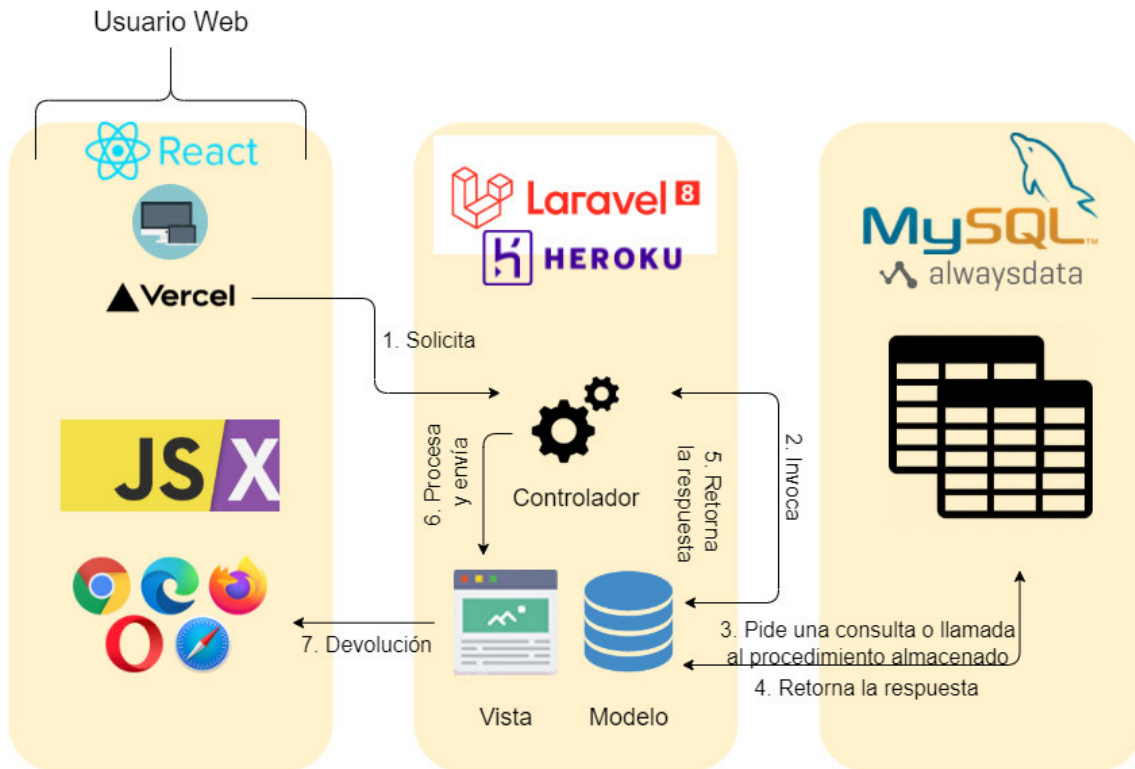


Fig. 2: Patrón arquitectónico del sistema web

2.4 Herramientas de desarrollo

Entre las herramientas de desarrollo utilizadas podemos encontrar:

2.4.1 Sistema Web

TABLA III: Herramientas para el desarrollo del Sistema Web

Herramienta	Justificación
XAMPP	Se ha utilizado esta distribución Apache ya que contiene diferentes softwares libres que se utilizan en el proyecto como <i>MySQL</i> y a su vez el servidor de apache que permite ejecutar la aplicación en el servidor local, sin preocuparse de tener acceso a internet [14] [15].
Composer	Es un gestor de dependencias de <i>PHP</i> que permite administrar y añadir librerías, se encarga de descargar las dependencias necesarias de manera automática [16].
JWT	<i>JSON Web Token</i> es un estándar abierto que permite transmitir de forma segura entre las partes de un <i>JSON</i> mediante un <i>token</i> generado asegurando que estos sean válidos y seguros, además estos <i>JWT</i> se pueden firmar [17]. Esto se utiliza en el sistema web para que cuando el usuario requiere iniciar sesión enviando sus credenciales, se genere un <i>token</i> único para mantener la sesión activa.

ReactJs	Se ha utilizado esta biblioteca de JavaScript primeramente porque la librería de <i>UI</i> que se define se integra de manera sencilla con <i>ReactJs</i> , además que está basado en componentes que poseen su propio estado, finalmente utiliza la sintaxis <i>JSX (JavaScript XML)</i> que facilita la integración del código con elementos visuales <i>HTML</i> .
Vercel	Se utiliza esta plataforma ya que permite alojar sitios web estáticos como interfaces de usuario dentro de la aplicación, además que se integra fácilmente con <i>Git</i> para poder desplegar un repositorio sin necesidad de hacer muchas configuraciones, finalmente si el repositorio de <i>Git</i> recibe cambios, <i>Vercel</i> automáticamente los sincroniza y la interfaz de usuario estará siempre actualizada [18].
Laravel	Se ha utilizado esta herramienta por el motivo de que provee varios instrumentos que aportan en el desarrollo, enfocándose en mejorar productividad y ahorrar tiempo al programar, además este <i>framework</i> toma varias librerías y las abstrae para que sean más fáciles de usar, finalmente se puede realizar una configuración bastante rápida para conectar a la base de datos, creación de la <i>API</i> a través de los métodos mágicos [19].
Heroku	Se ha utilizado esta plataforma ya que permite alojar la lógica de una aplicación trabajando como una <i>PaaS</i> que está basada en contenedores, además es bastante simple desplegar la lógica de la aplicación con simples comandos que su página mismo proporciona, por último, esta plataforma permite subir un número determinado de aplicaciones de manera gratuita [20].
MySQL	Se ha utilizado esta base de datos primeramente porque en el desarrollo en local se puede integrar con la herramienta <i>XAMPP</i> fácilmente, además asegura un gran desempeño, seguridad y completo control, finalmente es una base de datos que se puede integrar con <i>Laravel</i> el mismo que utiliza el <i>ORM</i> de <i>Eloquent</i> [21].
Alwaysdata	Se ha utilizado esta plataforma ya que permite alojar la base de datos en línea, además que ofrece una alta disponibilidad de los datos a través de una cadena de conexión, por último, posee una interfaz gráfica para visualizar la base de datos [22].

3 RESULTADOS Y DISCUSIÓN

Como resultado se ha obtenido un sistema web que utiliza *JSON Web Token (JWT)* para autenticar a los usuarios, diferenciándolos por su rol correspondiente, esto permite mostrar una interfaz diferente a cada usuario dependiendo de su rol. A su vez se utilizan políticas basadas en los roles de los usuarios por lo que se restringe la información a la que puede acceder un usuario u otro dependiendo de su rol y de su identificador de usuario. El estudiante posee tres vistas sencillas que son:

- El perfil: Donde puede visualizar sus datos personales como su nombre, la carrera a la que pertenece y las materias en las cuales se encuentra inscrito.
- Encuestas: Donde puede visualizar las encuestas pendientes con su respectivo tema y objetivo, estas encuestas están organizadas por materia en un botón desplegable. Por otro lado, también se muestran las encuestas realizadas que de igual forma se organizan por materia, pero se diferencian en que la información de la encuesta es el tema y la fecha en la que el estudiante responde la encuesta, esto se lo toma como una evidencia de que estudiante si ha llenado la encuesta.
- Encuesta: Esta visualización se muestra cuando el estudiante selecciona una encuesta para ser respondida, esta visualización posee las 5 preguntas correspondientes y un comentario donde el estudiante puede expresarse acerca del tema.
- Confirmación de envío de encuesta: Esta visualización aparece cuando se verifica que la encuesta se ha cargado correctamente a la base de datos y posee un mensaje para motivar al estudiante a que continúe proporcionando su retroalimentación para que exista un mejoramiento continuo.

3.1 *Sprint* 0. Configuraciones iniciales

Basándose en la organización del *Sprint Backlog*, en este *Sprint* 0 contiene las tareas básicas para las configuraciones iniciales del proyecto, así como el entorno de desarrollo para poder realizar el desarrollo de las funcionalidades. Para ese primer *Sprint* se han realizado las siguientes tareas:

3.1.1 Creación de mockups.

Se ha realizado el diseño de todas las pantallas con *Figma*:

La Fig. 3 muestra el *mockup* de la pantalla de módulo de inicio de sesión.



Fig. 3: *Mockup*: Módulo inicio de sesión

Dentro de los *mockups* del módulo de docente y estudiante se tiene su respectivo perfil de docente como se indica en la Fig. 4.



Fig. 4: *Mockup*: Perfil

Todos los demás *mockups* se encuentran en los anexos (Manual técnico, sección 6. Diseño de interfaces, página 16).

3.1.2 Diseño de base de datos.

La base de datos se ha diseñado en *MySQL* y se ha diagramado con las herramientas de diseño de *MySQL* como se muestra en la **Fig. 5**.

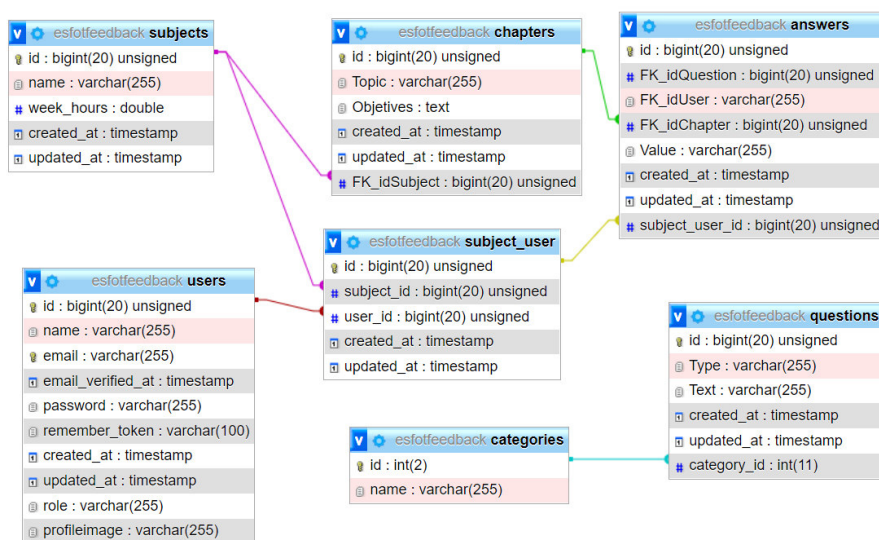


Fig. 5: Diseño de base de datos

Los resultados del primer *sprint* han sido: diseños de interfaz claros a seguir y se ha establecido el proceso de negocio a través de un análisis de la base de datos.

3.2 *Sprint* 1. Creación de *Endpoints*, relaciones y recursos

Dentro de este *sprint* constan la creación de *endpoints* para consumir la información de la base de datos a través de un *API*.

3.2.1 Creación de relaciones.

Las relaciones entre las entidades se han realizado con las claves foráneas, creando migraciones para establecer dichas relaciones en la base de datos como se aprecia en la **Fig. 6** es el formato en el cual se realizan todas las relaciones.

```
Schema::create('subject_user', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->unsignedBigInteger('subject_id');
    $table->foreign('subject_id')->references('id')->on('subjects')->onDelete('restrict');
    $table->unsignedBigInteger('user_id');
    $table->foreign('user_id')->references('id')->on('users')->onDelete('restrict');
    $table->timestamps();
});
```

Fig. 6: Relación materia – usuario

3.2.2 Creación de EndPoints.

Se han creado *endpoints* para los diferentes tipos de consulta a la base de datos, cabe mencionar que dichos *endpoints* se han creado bajo la versión de *Laravel 8* y en la **Fig. 7** se muestra la estructura de estos.

```
//Subjects
Route::get('subjects', [SubjectController::class, 'index']);
Route::get('subjects/{subject}', [SubjectController::class, 'show']);
Route::post('subjects', [SubjectController::class, 'store']);
Route::put('subjects/{subject}', [SubjectController::class, 'update']);
Route::delete('subjects/{subject}', [SubjectController::class, 'delete']);
```

Fig. 7: Endpoints para materias

3.2.3 Creación de recursos.

Los recursos permiten obtener archivos *JSON* desde los modelos, es decir, actúan como una capa intermedia de transformación, la estructura para los recursos ha sido como se muestra en la **Fig. 8**.

```
public function toArray($request)
{
    return [
        'id' => $this->id,
        'Topic' => $this->Topic,
        'Objetives' => $this->Objetives,
        'Subject' => "/api/subjects/" . $this->FK_idSubject,
        'answers' => new AnswerCollection($this->answers),
        'created_at' => $this->created_at,
        'updated_at' => $this->updated_at,
    ];
}
```

Fig. 8: Recurso para capítulos

3.2.4 Creación de seeders.

Los *seeders* permiten crear data de prueba en la base de datos, con el fin de probar las diferentes funcionalidades del sistema, los *seeders* tienen una estructura como se muestra en la **Fig. 9**. Para la creación de los *seeders* se ha utilizado la librería de *Faker* para poder generar data aleatoriamente dependiendo de los parámetros que se le pase.

```

// Vaciamos la tabla answers
Answer::truncate();
$faker = \Faker\Factory::create();
// Obtenemos todos los question de la bdd
$users = User::all();
foreach ($users as $user) { // iniciamos sesión con este usuario
    JWTAuth::attempt(['email' => $user->email, 'password' => '123123']);

    $question = Question::all();
    // Obtenemos todos los chapters de la bdd
    $chapter = Chapters::all();
    $subject_users= SubjectUser::all();

    foreach($subject_users as $subject_user){
        foreach($chapter as $chapters){
            foreach ($question as $questions) {
                $num_answer = 5;
                for ($j = 0; $j < $num_answer; $j++) {
                    Answer::create([
                        'FK_idUser' => $faker->uuid,
                        'FK_idChapter' => $chapters->id,
                        'FK_idQuestion' => $questions->id,
                        'user_subject_id' => $subject_user->id,
                        'Value' => $faker->numberBetween(1,5)
                    ]);
                }
            }
        }
    }
}

```

Fig. 9: Seeder de respuestas

3.3 Sprint 2. Diseño de páginas principales

Las tareas desarrolladas en este *sprint* han permitido consumir el *API* desde una página web y a su vez desplegar la información necesaria para el proceso.

3.3.1 Implementación de JWT para autenticación de usuarios.

JWT permite controlar la autenticación de usuarios mediante la implementación de *tokens*, además se puede controlar y proteger las rutas de los *endpoints* con *JWTMiddleware* como se muestra en la Fig. 10.

```

Route::group(['middleware' => ['cors']], function () {
    Route::post('register', [UserController::class, 'register']);
    Route::post('login', [UserController::class, 'authenticate']);

    Route::group(['middleware' => ['jwt.verify']], function () {
        Route::get('user', [UserController::class, 'getAuthenticatedUser']);
    });
}

```

Fig. 10: *JWTMiddleware*

3.3.2 Conexión entre el backend y el frontend.

Para conectar el *backend* y *frontend* se configura la dirección del *API* que se consume en el *frontend*, esto se debe hacer en el archivo “.env.development” como se muestra en la **Fig. 11**.

```
.env.development
1 REACT_APP_API_HOST=http://localhost:8000/api
```

Fig. 11: Configuración de *API* en *frontend*

3.3.3 Diseño de página principal.

La página principal ha sido diseñada en base a los *mockups* y como resultado se obtuvo una interfaz como se muestra en la **Fig. 12**.



Fig. 12: Página principal

3.3.4 Implementación de rutas de navegación.

Las rutas de navegación permiten a los usuarios moverse entre las diferentes vistas del sistema, esto ha sido implementado en un *navbar* en donde se encuentran las opciones por las cuales puede navegar el usuario dependiendo de su rol, este menú se puede apreciar en la **Fig. 13**.

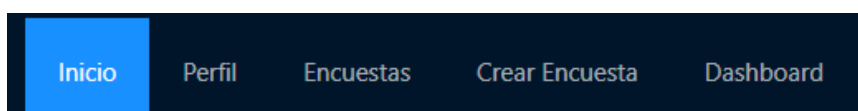



Fig. 13: Rutas de navegación

3.3.5 Diseño de página de inicio de sesión.


La página de inicio de sesión permite a los usuarios ingresar al sistema, el resultado final se lo muestra en la **Fig. 14**.



Modulo de inicio se sesión

Correo institucional:

Contraseña:

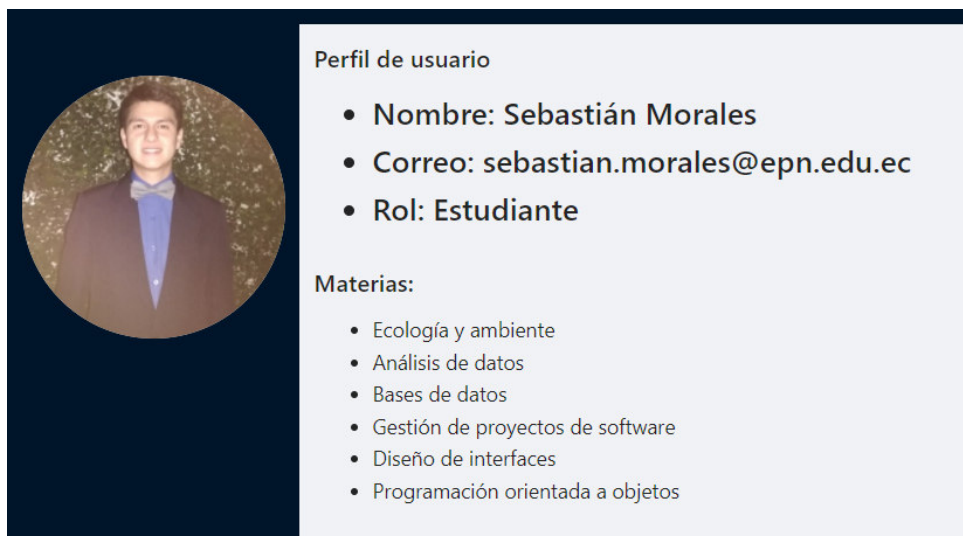
 

Ingresar

Fig. 14: Módulo inicio de sesión

3.3.6 Diseño de página de perfil.

La página de perfil permite a los usuarios visualizar su información personal como se muestra en la **Fig. 15**.



Perfil de usuario

- Nombre: Sebastián Morales
- Correo: sebastian.morales@epn.edu.ec
- Rol: Estudiante

Materias:

- Ecología y ambiente
- Análisis de datos
- Bases de datos
- Gestión de proyectos de software
- Diseño de interfaces
- Programación orientada a objetos

Fig. 15: Información de perfiles

3.4 *Sprint 3*. Roles y autorizaciones

En este *sprint* se han creado políticas de autorización a los diferentes roles para consumir el *API*.

3.4.1 Implementación de roles

Implementar roles permite al sistema web diferenciar entre los diferentes tipos de usuario (docente y estudiante) para de esta manera mostrar una u otra vista dependiendo del tipo de usuario que ingresa al sistema, el código que permite todo esto se muestra en la **Fig. 16**.

```
public function view(User $user, Answer $answer)
{
    return $user->isGranted(User::ROLE_STUDENT) ||
        ($user->isGranted(User::ROLE_TEACHER) &&
            $user->id === $answer->user_subject_id);
}
```

Fig. 16: Roles y autorizaciones

3.4.2 Implementación de autorización de usuarios.

Para el manejo de autorización se han utilizado los diferentes roles mencionados en la **Fig. 16**.

3.5 *Sprint 4*. Creación de encuestas

En este *sprint* se ha realizado el proceso de creación de encuestas y su respectivo despliegue en la página, tanto para la creación de encuestas como para responderlas.

3.5.1 Creación de políticas de encuestas.

Para las encuestas se han creado políticas que permiten que los usuarios de tipo docente puedan crearlas, los estudiantes responderlas y ambos puedan verlas como se indica en la **Fig. 17**.

```
public function create(User $user)
{
    return $user->isGranted( role: User::ROLE_STUDENT) || $user->isGranted( role: User::ROLE_TEACHER);
}
```

Fig. 17: Políticas de preguntas

3.5.2 Diseño de página de encuestas (Creación de encuestas).

El sistema web permite a los usuarios de tipo docente crear las encuestas como se indica en la **Fig. 18**.

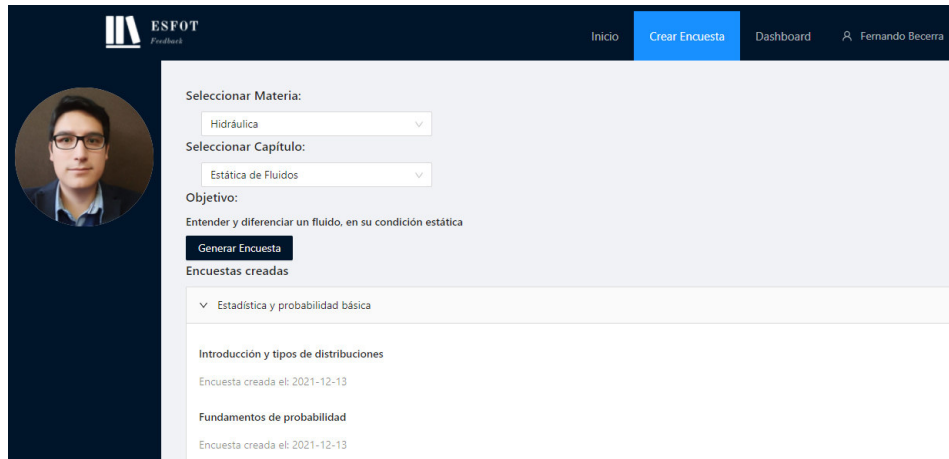


Fig. 18: Creación de encuestas

3.5.3 Diseño de página de encuestas (Envío de respuestas a las encuestas).

El sistema web permite a los usuarios de tipo estudiante responder y enviar encuestas como se indica en la **Fig. 19**.

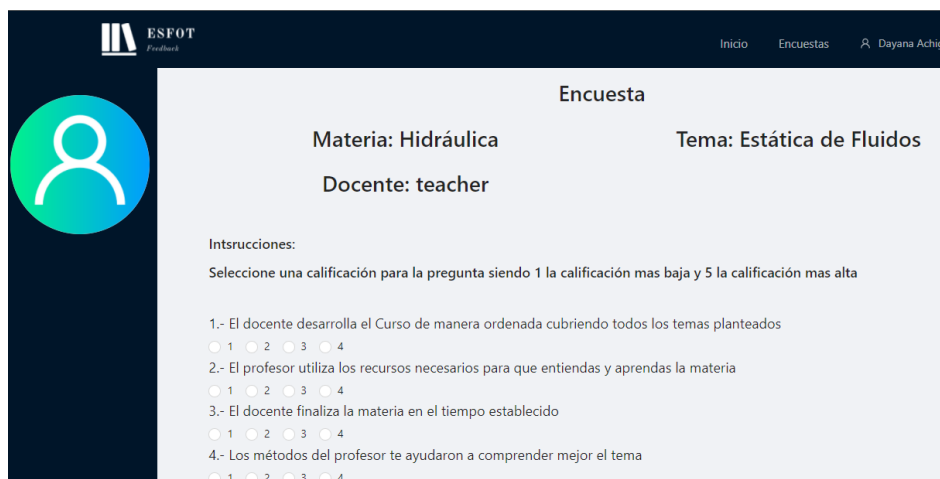


Fig. 19: Respuesta de encuestas

3.6 *Sprint* 5. Visualización de encuestas

Dentro de este *sprint* se ha desplegado la información de encuestas, las respuestas de estas y el despliegue en un tablero y por último se realizan las respectivas pruebas de funcionalidad, de carga y de compatibilidad.

3.6.1 Diseño de página de encuestas (Visualización de encuestas).

Los estudiantes pueden ver las encuestas pendientes y realizadas como se indica en la **Fig. 20**.

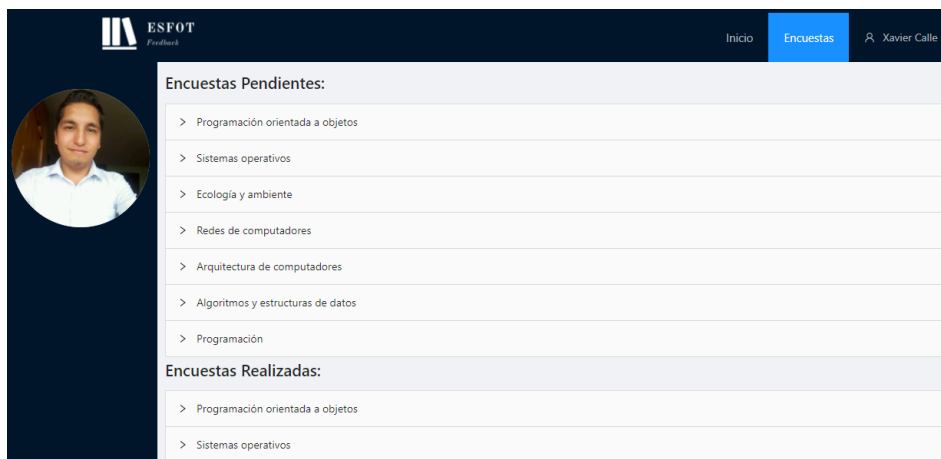


Fig. 20: Visualización de encuestas (estudiante)

3.6.2 Diseño de tablero de visualización de encuestas para docentes.

Los docentes pueden ver las encuestas, realizadas por los estudiantes, organizadas por materias y capítulos como se indica en la **Fig. 21** y más detalles como se indica en la **Fig. 22** donde se observa el detalle ponderado de cada una de las preguntas respondidas, los comentarios de los estudiantes hacia el profesor, un medidor de rendimiento en general y finalmente un grafico tipo radar donde se puede observar en que categoría el profesor destaca o no destaca.

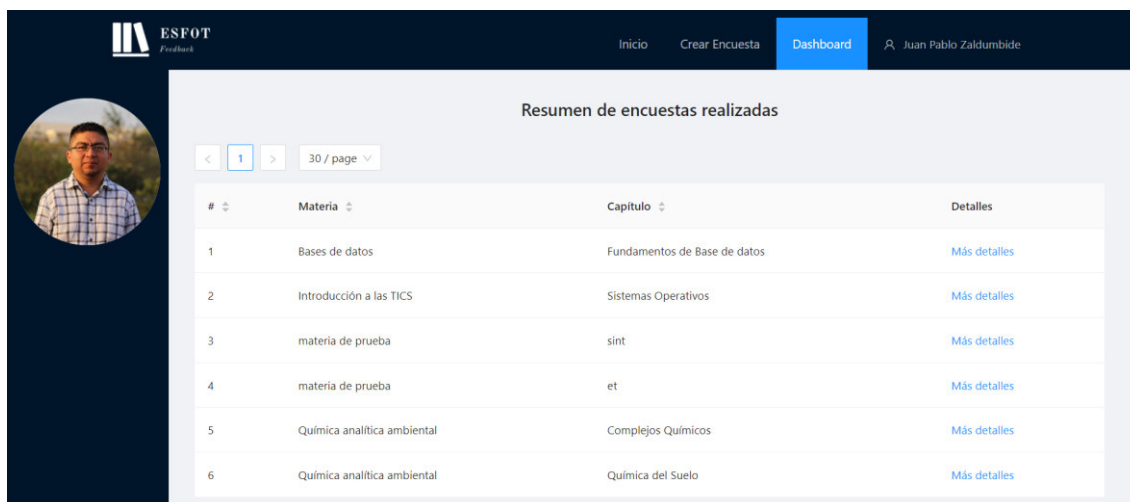


Fig. 21: Visualización de encuestas (docente)

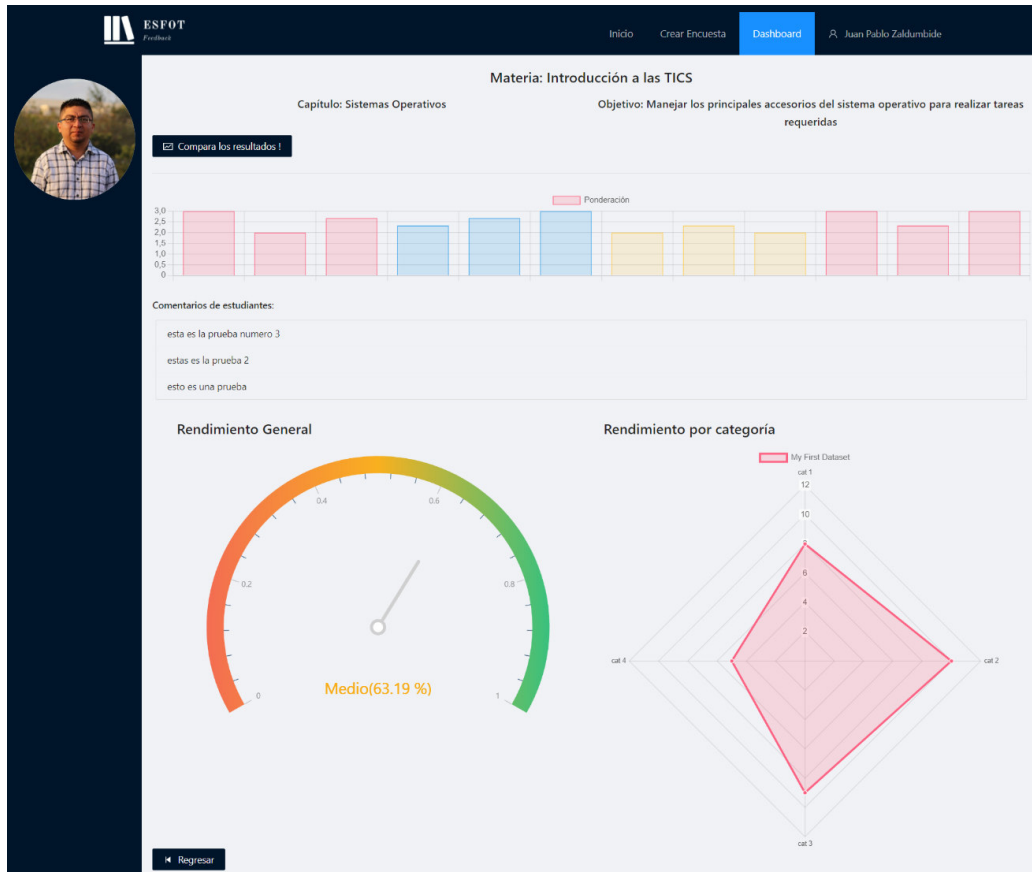


Fig. 22: Detalles de encuestas (docente)

3.6.3 Comparación de resultados de encuestas.

Un docente puede comparar los resultados de dos encuestas, para esto debe hacer clic en el botón “Compara los resultados!” que se muestra en la Fig. 22, lo hará esta acción es redirigir al docente al tablero donde mira todas las encuestas (Fig. 21) exceptuando la que ha seleccionado para comparar, y cambiando la columna del detalle por el texto comparar, tal como se muestra en la Fig. 23.

The dashboard displays the following information:

- Navigation:** Inicio, Crear Encuesta, **Dashboard**, Juan Pablo Zaldumbide
- Header:** Resumen de encuestas realizadas
- Table:**

#	Materia	Capítulo	Detalles
1	Bases de datos	Fundamentos de Base de datos	Comparar
3	materia de prueba	sint	Comparar
4	materia de prueba	et	Comparar
5	Química analítica ambiental	Complejos Químicos	Comparar
6	Química analítica ambiental	Química del Suelo	Comparar
- Footer:** Regresar

Fig. 23: Visualización de encuestas a comparar

Por último, la visualización de encuestas comparadas distingue cada uno de los resultados obtenidos y muestra una visualización tal como se muestra en la

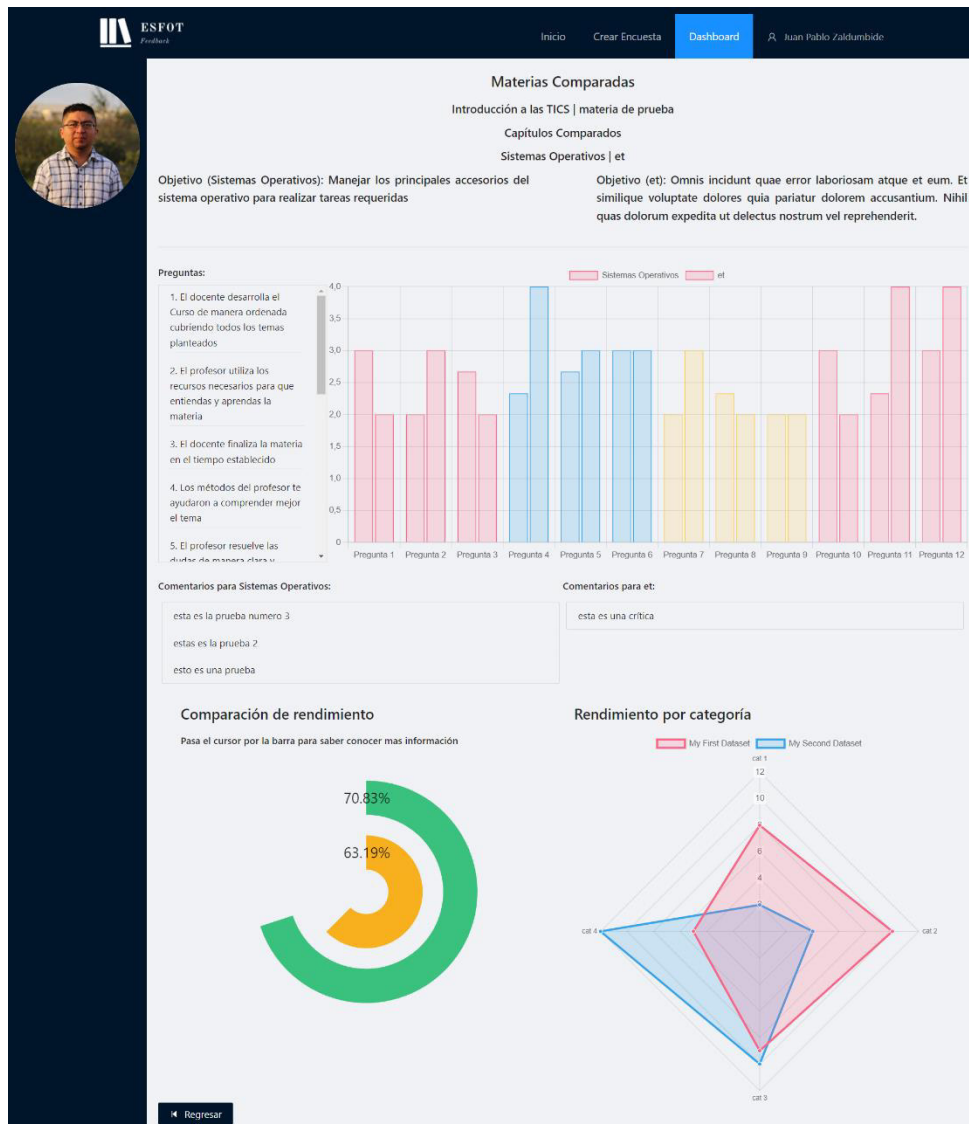


Fig. 24: Visualización de resultados de encuestas comparadas

3.7 Sprint 6. Despliegue del sistema web y pruebas

Dentro de este *sprint* se han realizado las tareas finales como son el despliegue del sistema web 100% en línea y las respectivas pruebas tanto de carga, de funcionalidad y de compatibilidad.

3.7.1 Despliegue del sistema web

Para el despliegue del sistema web se han utilizado herramientas diferentes para subir cada una de las partes de la aplicación como son el *backend*, *frontend* y base de datos.

Para desplegar el *frontend* se utiliza la herramienta *Heroku* y se comprueba que se haya subido correctamente visitando el enlace que ha provisto la herramienta, esto se muestra en la **Fig. 25**.

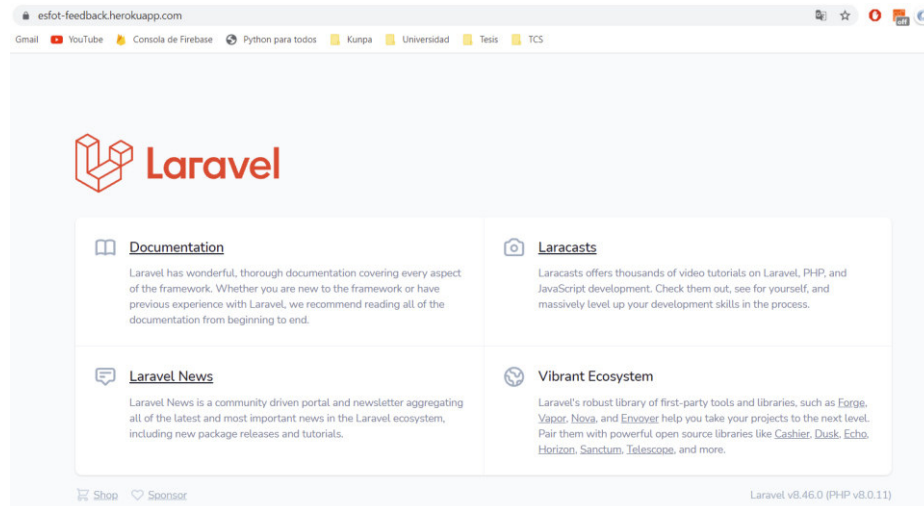


Fig. 25: Despliegue del sistema web en *Heroku*

Para el despliegue del *frontend* se utiliza la herramienta *Vercel*, de igual manera que para desplegar el *backend* se comprueba que la *URL* otorgada por la herramienta posea la interfaz correspondiente, esto se puede apreciar en la **Fig. 26**.

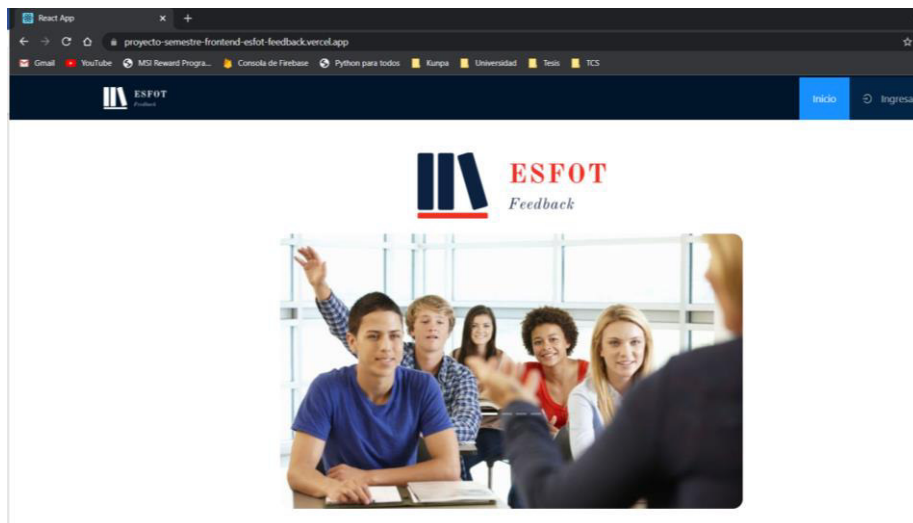


Fig. 26: Verificación del despliegue en *Vercel*

Por último, para el despliegue de la base de datos se utiliza la herramienta *Alwaysdata* e igual que el *backend* y el *frontend* se comprueba que la *URL* provista contenga la base de datos, esto se muestra en la **Fig. 27**.

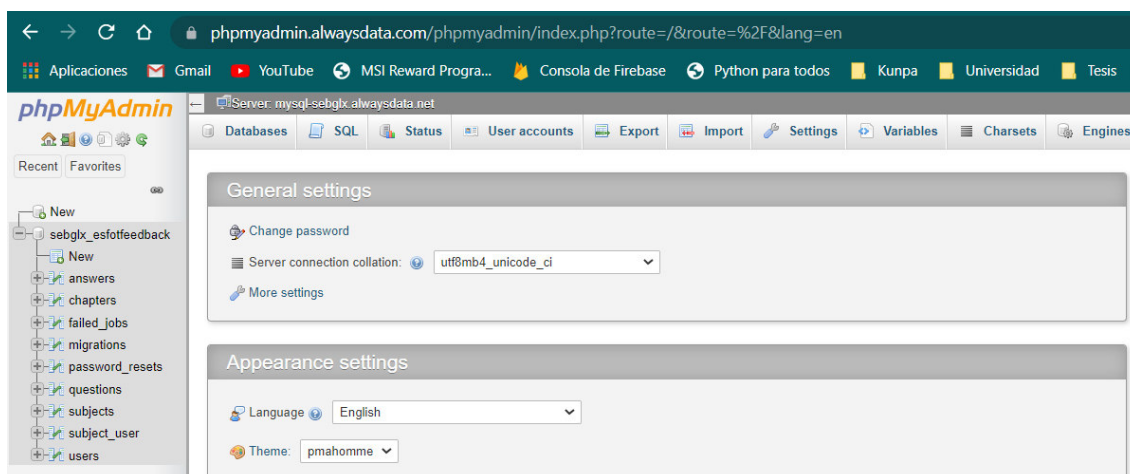


Fig. 27: Verificación de subida de base de datos

Todo el detalle y los pasos para la subida de cada una de las partes del sistema web que se mencionan anteriormente se las detalla en los anexos (Manual de instalación, sección 2, 3 y 4, página 3).

3.7.2 Pruebas de funcionalidad, de carga y compatibilidad

Las pruebas funcionales se las realiza ingresando al sistema web y probando cada una de las características de forma manual, teniendo claro los criterios de evaluación de la prueba y verificando que pase correctamente, el formato de una de las pruebas de funcionalidad se muestra en la **TABLA IV**.

TABLA IV: Formato pruebas funcionales

PRUEBA DE FUNCIONALIDAD	
Identificador (ID): PF001	Identificador historia de usuario: HU001
Nombre prueba de funcionalidad: Inicio de sesión	
Descripción: El usuario cualquiera que sea su rol y que exista en la base de datos debe poder iniciar sesión en el Sistema Web.	
Pasos de ejecución: 1. El usuario ingresa al Sistema Web. 2. Dar clic en el módulo "Ingresar". 3. El usuario llena los campos requeridos (correo y contraseña). 4. Dar clic en el botón "Ingresar".	
Resultado deseado: El Sistema Web redirige al usuario a la página de inicio y muestra en el menú las opciones respectivas dependiendo del rol del usuario.	
Evaluación de la prueba: Resultado exitoso. El Sistema Web permite que un usuario inicie sesión y distinga su rol.	

Las pruebas de carga se las realizaron en la Herramienta *JMeter*, un ejemplo de estas pruebas se aprecia en la **Fig. 28** y se la detalla más en los anexos (Manual técnico, sección 8. Pruebas de Carga, página 22) junto con sus respectivos resultados.

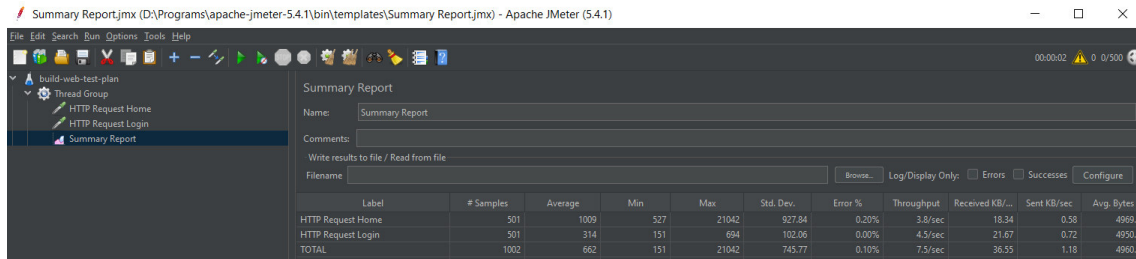


Fig. 28: Ejemplo prueba de carga

Las pruebas de compatibilidad únicamente se las realizan abriendo el sistema web en distintos navegadores, en este caso se ha utilizado *Chrome*, *Edge* y *Firefox*, esas pruebas están presentes en los anexos (Manual técnico, sección 9. Pruebas de Compatibilidad, página 23).

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El sistema web cumple con los requerimientos determinados a partir de un estudio e investigación del problema, lo que permite a usuarios docentes recibir retroalimentación continua por parte de sus estudiantes por medio de encuestas con el fin de que exista un mejoramiento continuo por parte del docente en el momento de impartir la materia.
- Tener en claro el flujo del sistema, la base de datos y las interfaces de usuario, ha permitido que el desarrollo se realice de manera efectiva evitando malentendidos entre los desarrolladores, comprendiendo complementemente como se unen cada una de las partes del sistema web para su correcto funcionamiento.
- La implementación de un sistema web para que docentes reciban una retroalimentación continua, ha permitido que los estudiantes puedan dar su opinión de manera segura y anónima, erradicando el problema de que los estudiantes no dan su punto de vista cuando el docente pide retroalimentación, por miedo a que se tomen represalias en su contra.
- La integración de pruebas de funcionalidad ha permitido que se verifique que todo en el sistema web funcione de manera correcta y óptima en diferentes navegadores, cumpliendo el flujo establecido.
- Las pruebas de carga permiten conocer si la aplicación web acepta varias solicitudes a la vez sin generar errores o bajar su rendimiento.
- Las pruebas de compatibilidad permiten conocer si el sistema web funciona en distintos navegadores.
- El despliegue del sistema web a un servidor en línea ha permitido que, tanto estudiantes como docentes de la ESFOT, puedan acceder desde cualquier navegador al sistema web, haciendo que esté disponible desde cualquier parte.
- El uso del marco de trabajo *Scrum* ha facilitado el desarrollo del sistema web permitiendo mantener un avance continuo, una buena sincronización y que se tengan claros los requerimientos del sistema web con el equipo de desarrollo.
- Llevar a cabo la creación de un sistema web que permite generar y analizar los resultados de encuestas mediante un solo clic ha permitido ahorrar tiempo ya que integra el análisis de datos en la misma plataforma.
- El manejo de un control de versiones adecuado en el equipo de desarrollo ha permitido determinar que tarea debe ser completada para que otra pueda iniciar,

haciendo que estas se prioricen y no exista un retraso en el desarrollo, a su vez dentro del flujo del sistema web, tener en claro cuáles son los actores principales ha permitido generar una mejor experiencia de usuario.

- La implementación de un sistema web que cumple con las especificaciones de usuario ha permitido que el mismo cumpla con su funcionalidad y no haya sido necesario realizar cambios grandes dentro del proceso.
- El uso de *ReactJs* como *Framework* ha permitido colocar el maquetado y la lógica dentro de un mismo componente gracias a *JSX*, esto simplifica la organización de los componentes del proyecto permitiendo la reutilización de estos con sus respectivas funcionalidades.

4.2 Recomendaciones

- Se recomienda tener siempre las versiones de los respectivos *Frameworks* para que el sistema web siempre se encuentre actualizado y evitar errores de versión.
- Conocer las herramientas de desarrollo y leer la documentación oficial de las mismas para que no existan dudas ni percances en el desarrollo del proyecto.
- Se recomienda que el proyecto se integre con el departamento de mejora continua de la ESFOT para que se tenga información de cómo es el desempeño de un docente en las clases que imparte.
- Se recomienda realizar una copia de la base de datos de manera bimestral en la nube para que exista alta disponibilidad de los datos en el caso de que ocurra una pérdida de datos en el servidor principal.
- Se recomienda utilizar el manual de usuario para el correcto uso del sistema.
- Se recomienda a los estudiantes revisar el sistema web periódicamente para llenar las encuestas y cumplir con el proceso requerido de parte de los docentes.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] ESFOT, «¿Quiénes Somos?,» 2020. [En línea]. Available: <https://esfot.epn.edu.ec/index.php/quienes-somos>. [Último acceso: 5 Agosto 2020].
- [2] Escuela Politécnica Nacional, *Manual De Procedimientos*, Quito, Pichincha, 2017.
- [3] A. P. O. R. G. Guerrero Radillo, «El papel de la retroalimentación y la ausencia o presencia de instrucciones en la elaboración de,» Universidad Católica de Colombia, Bogota, 2007.
- [4] M. T. Gallego, «Metodología Scrum,» 2012.
- [5] M. B. A. v. B. A. C. W. C. M. F. J. G. J. H. A. H. R. J. J. K. B. M. R. C. M. S. M. K. S. J. S. D. T. Kent Beck, «Manifiesto for Agile Software Development,» 2001. [En línea]. Available: <https://agilemanifesto.org/>. [Último acceso: 29 agosto 2021].
- [6] J. L. A. P. María Paula Gonzales, «Evaluación Heurística,» Universitat de Lleida, 2001.
- [7] Y. D. G. Yenisleidy Fernández Romero, «Patrón Modelo-Vista-Controlador,» *TELEM@TICA*, vol. 11, nº 1, pp. 47-57, 2012.
- [8] Facebook INC, React, [En línea]. Available: <https://es.reactjs.org/>. [Último acceso: 29 agosto 2021].
- [9] Vercel Inc, «about,» Vercel, [En línea]. Available: <https://vercel.com/about>. [Último acceso: 29 agosto 2021].
- [10] Laravel LLC, Laravel, [En línea]. Available: <https://laravel.com/>. [Último acceso: 29 agosto 2021].
- [11] salesforce, «about,» Heroku, [En línea]. Available: <https://www.heroku.com/about>. [Último acceso: 29 agosto 2021].
- [12] Oracle Corporation, «About MySQL,» MySQL, [En línea]. Available: <https://www.mysql.com/about/>. [Último acceso: 29 agosto 2021].

- [13] alwaysdata, alwaysdata, [En línea]. Available: <https://www.alwaysdata.com/en/>. [Último acceso: 29 agosto 2021].
- [14] Apache Friends, «¿Qué es XAMPP?,» Apache Friends, [En línea]. Available: <https://www.apachefriends.org/es/index.html>. [Último acceso: 20 Septiembre 2021].
- [15] «¿Qué es XAMPP y para que sirve?,» Cuaderno de Clase, 23 Marzo 2017. [En línea]. Available: <http://janda1617smr2curro.blogspot.com/2017/03/que-es-xampp-y-para-que-sirve.html>. [Último acceso: 20 Septiembre 2021].
- [16] Yair, «Qué es Composer y cómo usarlo,» styde, 23 Diciembre 2019. [En línea]. Available: <https://styde.net/que-es-composer-y-como-usarlo/>. [Último acceso: 20 Septiembre 2021].
- [17] JWT, «Introduction to JSON Web Tokens,» JWT, [En línea]. Available: <https://jwt.io/introduction>. [Último acceso: 20 Septiembre 2021].
- [18] V. Inc, «Introduction to Vercel,» Vercel, [En línea]. Available: <https://vercel.com/docs/concepts>. [Último acceso: 20 Septiembre 2021].
- [19] Laravel LLC, «Why Laravel?,» Laravel, [En línea]. Available: <https://laravel.com/docs/8.x#why-laravel>. [Último acceso: 20 Septiembre 2021].
- [20] ricardocelis, «Heroku: qué es, cómo funciona y para qué sirve - Platzi,» Platzi, 2017. [En línea]. Available: <https://platzi.com/blog/que-es-heroku/>. [Último acceso: 20 Septiembre 2021].
- [21] «8 Major Advantages of Using MySQL,» Datamation, 16 Noviembre 2016. [En línea]. Available: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/>. [Último acceso: 16 Septiembre 2021].
- [22] alwaysdata , «WHAT WE DO,» alwaysdata , [En línea]. Available: <https://www.alwaysdata.com/en/>. [Último acceso: 20 Septiembre 2021].

6 ANEXOS

6.1 Manual Técnico

- Recopilación de requerimientos
- Requisitos del sistema
- Historias de usuario
- *Product Backlog*
- *Sprint Backlog*
- Diseño de interfaces
- Diseño de base de datos
- Pruebas de Carga
- Pruebas de Compatibilidad
- Pruebas de funcionalidad

6.2 Manual de Usuario

https://drive.google.com/drive/folders/1Yce2uQn6_YTGzqF1pDXna94nv8M4Nt8P?usp=sharing

6.3 Manual de Instalación

- Requisitos
- Despliegue de base de datos en *AlwaysData*
- Despliegue de *backend* en *Heroku*
- Despliegue del *frontend* en *Vercel*
- Credenciales de acceso al sistema web
- Repositorio del código fuente del sistema web

De igual forma todos los manuales y la documentación se encuentran en el siguiente enlace de GitHub: https://github.com/mastercode15/Documentacion_Tesis