



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TEGNOLOGÍAS DE LA INFORMACIÓN

TEMA:

**SISTEMA WEB RESPONSIVE UTILIZANDO CONTENEDORES Y EL STACK MERN
PARA MEJORAR LA EXPERIENCIA DE USUARIO EN EL SISTEMA DE
EVALUACIÓN DOCENTE DE LA UNIVERSIDAD TÉCNICA DE AMBATO**

Trabajo de Integración Curricular Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Célimo David Constante Prócel

TUTOR: Ing. Oscar Fernando Ibarra Torres

Ambato - Ecuador

marzo – 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Integración Curricular con el tema: SISTEMA WEB RESPONSIVE UTILIZANDO CONTENEDORES Y EL STACK MERN PARA MEJORAR LA EXPERIENCIA DE USUARIO EN SISTEMA DE EVALUACIÓN DOCENTE DE LA UNIVERSIDAD TÉCNICA DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Célimo David Constante Prócel, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y el numeral 7.4 del respectivo instructivo del reglamento.

Ambato, marzo 2023.

Ing. Mg. Oscar Fernando Ibarra Torres

TUTOR

AUTORÍA

El presente trabajo de Integración Curricular titulado: SISTEMA WEB RESPONSIVE UTILIZANDO CONTENEDORES Y EL STACK MERN PARA MEJORAR LA EXPERIENCIA DE USUARIO EN SISTEMA DE EVALUACIÓN DOCENTE DE LA UNIVERSIDAD TÉCNICA DE AMBATO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023.



Célimo David Constante Prócel

CC. 1850176213


AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023.



Célido David Constante Prócel

CC. 1850176213

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por el señor Célimo David Constante Prócel, estudiante de la Carrera de tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA WEB RESPONSIVE UTILIZANDO CONTENEDORES Y EL STACK MERN PARA MEJORAR LA EXPERIENCIA DE USUARIO EN SISTEMA DE EVALUACIÓN DOCENTE DE LA UNIVERSIDAD TÉCNICA DE AMBATO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y al numeral 7.6 del respectivo instructivo del reglamento. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023.

Ing. Elsa Pilar Urrutia Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. David Omar Guevara Aulestia, Mg.

PROFESOR CALIFICADOR

Ing. Rubén Eduardo Nogales Portero, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

Todo tiene un fin, graduarse es el final de una carrera pero es el principio de una vida. Mi logro más ambicioso hasta la fecha está dedicado a mis papás, Patricia y Sixto, que desde un inicio aun conociendo todas mis “capacidades” nunca dudaron en apoyarme un solo segundo a la hora de tomar este viaje que además de largo, fue lo que yo siempre anhelé desde pequeño.

A mis hermanas Naty y Vero, que siempre supieron hacerme un espacio en sus vidas para ayudarme, y que además fueron y son un ejemplo de lo que yo quería y quiero ser cuando sea grande.

A mi pequeña hermana Estefy que, aunque ella no se haya dado cuenta me enseñó a mantener los pies sobre la tierra y entender que yo también puedo ser un ejemplo a seguir.

- Célimo David Constante Prócel

AGRADECIMIENTO

Se suele decir que somos el promedio de la gente que nos rodea, y para mi buena fortuna la vida y la Universidad Técnica de Ambato me dieron la oportunidad de rodearme de personas muy capaces, con metas y objetivos que van en un mismo sentido. Gracias a mis amigos: Luis, Anthony P., Felipe, Anthony L. y Paul por acompañarme en esta travesía en la cual me enseñaron el verdadero significado de la palabra amistad.

A todas las personas que consciente o inconscientemente pusieron su granito de arena para que yo pudiera acercarme a mi sueño de ser ingeniero. A mis amigos: Franco, David y Freddy que a pesar de los años siguen siendo mis amigos, a todas las personas que forman parte de mi vida, y a las que ya no también. Gracias.

INDICE GENERAL

Aprobación del tutor	ii
Autoría.....	iii
derechos de autor	iv
Aprobación del tribunal de grado	v
Dedicatoria.....	vi
Agradecimiento	vii
Resumen Ejecutivo	xv
Abstract.....	xvi
CAPÍTULO I.- marco teórico.....	17
1.1. Tema de Investigación	17
1.1.1. Planteamiento del Problema.....	17
1.2. Antecedentes Investigativos.....	18
1.3. Fundamentación Teórica.....	20
1.4. Objetivos	23
1.4.1. Objetivo General	23
1.4.2. Objetivos Específicos.....	23
CAPÍTULO II.- METODOLOGÍA.....	25
2.1. Materiales.....	25
2.2. Métodos.....	25
2.2.1. Modalidad de la Investigación	26
2.2.2. Población y muestra	26

2.2.3. Recolección de la Información.....	28
2.2.4. Procesamiento y Análisis de Datos	41
CAPÍTULO III.- rESULTADOS Y DISCUSIÓN	43
3.1. Análisis y discusión de los resultados.....	43
3.2. Desarrollo de la propuesta	70
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES.....	120
4.1. Conclusiones	120
4.2. Recomendaciones	121

ÍNDICE DE FIGURAS

Figura 1.	Comparación de usuarios móviles vs de escritorio	23
Figura 2.	Tabulación de resultados pregunta 1.....	30
Figura 3.	Tabulación de resultados pregunta 2.....	31
Figura 4.	Tabulación de resultados pregunta 3.....	32
Figura 5.	Tabulación de resultados pregunta 4.....	33
Figura 6.	Tabulación de resultados pregunta 5.....	34
Figura 7.	Tabulación de resultados pregunta 6.....	35
Figura 8.	Tabulación de resultados pregunta 8.....	36
Figura 9.	Tabulación de resultados pregunta 9.....	37
Figura 10.	Tabulación de resultados pregunta 10.....	38
Figura 11.	Tabulación de resultados pregunta 11.....	39
Figura 12.	Flujo de la metodología xp.....	44
Figura 13.	Reglas en las que se basa la metodología Kanban.....	45
Figura 14.	Flujo de la metodología SCRUM.....	46
Figura 15.	Flujo de trabajo.....	53
Figura 16.	Tablero metodología híbrida	54
Figura 17.	Diferencias entre VM y Contenedor.....	55
Figura 18.	Operaciones de Lectura y Escritura (menos es mejor).....	58
Figura 19.	Arquitectura de la plataforma.....	59
Figura 20.	Docker Hub.....	60
Figura 21.	Diagrama replica set.....	61
Figura 22.	Arquitectura backend.....	62

Figura 23.	Leyenda de prioridad de tareas.....	70
Figura 24.	Sprint 1: Diseño del Mockup.....	70
Figura 25.	Sprint 2: Inicialización del servidor.....	71
Figura 26.	Sprint 3: Desarrollo Api	71
Figura 27.	Sprint 4: Test de la Api.....	72
Figura 28.	Sprint 5: Desarrollo del Frontend.....	72
Figura 29.	Sprint 6: Despliegue de la plataforma	73
Figura 30.	Hoja de Ruta	73
Figura 31.	Epic 1, Sprint 1 – Diseño del Mockup.....	74
Figura 32.	Login de Usuario (1/2)	75
Figura 33.	Login de usuario (2/2)	76
Figura 34.	Mockup Home	78
Figura 35.	Mockup Docente Evaluacion.....	79
Figura 36.	Mockup Docente Estadísticas.....	80
Figura 37.	Mockup panel de adminstración.....	82
Figura 38.	Epic 2, Sprint 2 – Servidor inicial	84
Figura 39.	Repositorio en Github.....	85
Figura 40.	package.json	86
Figura 41.	docker-compose.yml – Configuración MongoDB	87
Figura 42.	Conexión a MongoDB Compass	88
Figura 43.	Base de datos desde MongoDB Compass	88
Figura 44.	Backend de la Aplicación.....	89
Figura 45.	Conexión a la Base de Datos	90
Figura 46.	Middleware manejo de errores	91

Figura 47.	Servidor Inicial	92
Figura 48.	Servidor corriendo en express	93
Figura 49.	Epic 3, Sprint 3 – Desarrollo Api	93
Figura 50.	Esquema de modelo – comentarios	94
Figura 51.	Modelos entidades	95
Figura 52.	Método obtener comentarios	95
Figura 53.	Método obtener comentarios mediante un id de docente	96
Figura 54.	Método crear comentario.....	96
Figura 55.	Método like comentario.....	97
Figura 56.	Método dislike comentario	97
Figura 57.	Método actualizar comentario	98
Figura 58.	Método eliminar comentario	98
Figura 59.	Rutas generales.....	99
Figura 60.	Rutas comentarios.....	100
Figura 61.	Generar Json Web Token	101
Figura 62.	Validación del Json Web Token.....	102
Figura 63.	Esquema Joi comentarios	103
Figura 64.	Middeware validar petición	104
Figura 65.	Rol invitado	104
Figura 66.	Epic 3, Sprint 4	105
Figura 67.	Test en el puerto de la API	105
Figura 68.	Test login y renew session.....	106
Figura 69.	Sistema de archivos de modulos y test.....	107
Figura 70.	Ejecución de los test	108

Figura 71.	Epic 4, Sprint 5	109
Figura 72.	Maquetado Home Page.....	110
Figura 73.	Maquetado Teacher Page.....	111
Figura 74.	Maquetado Login Page.....	112
Figura 75.	Maquetado Manage Page.....	112
Figura 76.	Epic 5, Sprint 6	113
Figura 77.	Action runner CI Backend.....	114
Figura 78.	Configuración del proxy inverso nginx	115
Figura 79.	Hoja de ruta	116
Figura 80.	Página de estadísticas generales coordinador	117
Figura 81.	Página de estadísticas por carrera.....	118

ÍNDICE DE TABLAS

Tabla 1.	Población con la que se realizó la investigación.....	26
Tabla 2.	Varianza de los resultados obtenidos en la encuesta realizada	28
Tabla 3.	Resultados de la ficha de observación.....	40
Tabla 4.	Criterios ponderados características metodologías	47
Tabla 5.	Metodología híbrida	49
Tabla 6.	Sql vs NoSql.....	56
Tabla 7.	Comparación software para mockup.....	65
Tabla 8.	Comparación gestores de peticiones	67
Tabla 9.	Comparacion SGBDS para MongoDB	68
Tabla 10.	Breakpoints según el tamaño de pantalla.	69
Tabla 11.	Tipos de usuarios y permisos.....	76
Tabla 12.	Criterios de evaluación	81
Tabla 13.	Dieño responsive.	83

RESUMEN EJECUTIVO

Existen diferentes componentes dentro del proceso de educación superior, la mayoría de las universidades del país tienen como objetivo formar profesionales con pensamiento crítico que se han especializado en una rama del conocimiento que les permite resolver diferentes problemáticas sociales, es por ello que las instituciones hacen su mayor esfuerzo para que los futuros profesionales durante su proceso de formación tengan a la mano los mejores recursos para garantizar una educación de calidad.

Además, son varios los elementos que interfieren durante el aprendizaje, infraestructura, p \acute{e} nsum acad \acute{e} mico, valores y principios, etc. Aun que se puedan tomar diferentes puntos de vista los docentes son una pieza fundamental durante la formaci \acute{o} n de futuros profesionales, ya que son ellos los encargados de guiar a los mismos durante su trayectoria acad \acute{e} mica. Por consiguiente, se debe poner especial cuidado en cuales son las practicas que los docentes imparten en sus c \acute{a} tedras y si estas est \acute{a} n alineadas con los estatutos institucionales. Solo de esta manera se podr \acute{a} garantizar que al culminar su proceso de formaci \acute{o} n un profesional podr \acute{a} desempe \acute{n} ar su cargo de la mejor manera posible.

Esta herramienta busca simplificar el proceso de evaluaci \acute{o} n docente para as \acute{i} poder generar una retroalimentaci \acute{o} n constante que permite elevar los est \acute{a} ndares de calidad en el proceso de aprendizaje. Se utiliz \acute{o} el stack MERN y Docker como herramientas principales para facilitar la distribuci \acute{o} n de la aplicaci \acute{o} n, todo esto mediante una metodolog \acute{i} a de desarrollo hibrida que recolecta las caracter \acute{i} sticas m \acute{a} s relevantes de las metodolog \acute{i} as comunes que se adaptan al contexto de desarrollo de la aplicaci \acute{o} n.

Palabras Clave: MERN, Docker, sistema web, responsive.

ABSTRACT

There are different components within the process of higher education, most universities in the country aim to train professionals with critical thinking that have specialized in a branch of knowledge that allows them to solve different social problems, which is why the institutions make their best effort so that future professionals during their training process have at hand the best resources to ensure a quality education.

In addition, there are several elements that interfere during the learning process, infrastructure, academic curriculum, values and principles, etc. Even though different points of view may be taken, teachers are a fundamental piece during the formation of future professionals, since they are the ones in charge of guiding them during their academic trajectory. Therefore, special care should be taken to determine what practices teachers teach in their courses and whether these are aligned with the institutional statutes. Only in this way will it be possible to guarantee that at the end of their training process a professional will be able to perform their position in the best possible way.

This tool seeks to simplify the teaching evaluation process in order to generate a constant feedback that allows raising the quality standards in the learning process. The MERN stack and Docker were used as main tools to facilitate the distribution of the application, all this through a hybrid development methodology that collects the most relevant features of common methodologies that are adapted to the development context of the application.

Key words: MERN, Docker, web system, responsive.

CAPÍTULO I.- MARCO TEÓRICO

1.1. Tema de Investigación

SISTEMA WEB RESPONSIVE UTILIZANDO CONTENEDORES Y EL STACK MERN PARA MEJORAR LA EXPERIENCIA DE USUARIO EN SISTEMA DE EVALUACIÓN DOCENTE DE LA UNIVERSIDAD TÉCNICA DE AMBATO

1.1.1. Planteamiento del Problema

El sistema de educación superior en todos sus aspectos y niveles está constituido de varios factores para que el mismo pueda ser considerado de calidad y por ende de provecho para la sociedad. La educación se considera como una relación multidimensional que engloba diferentes aspectos como: infraestructura, institución académica, estudiantes, docentes, etc. Según Arreola [1] El alumno es el actor social sin el cual no tendría sentido la educación, es el protagonista de su propio aprendizaje y por lo tanto el nivel de compromiso que este asuma determinará en mucho el logro educativo. Generalmente se señala al estudiante como un elemento hegemónico dentro del proceso educativo es necesario señalar que el proceso no depende solo de este en cuestión, los diferentes aspectos que lo rodean también son un factor importante dentro de la formación de profesionales de provecho para la sociedad. Arreola [1] también señala que: dado que son las y los docentes quienes interactúan de manera directa, constante y cotidiana con las y los estudiantes, es fundamental el desarrollo de competencias no solo propias de la disciplina que imparten sino, además competencias comunicativas, sociales, efectivo-emocionales, éticas y de superación profesional permanente. Por lo que es importante involucrar de una adecuada manera a los docentes dentro del proceso de enseñanza aprendizaje.

También es importante señalar que la Evaluación docente corresponde a emitir un juicio de valor respecto al cumplimiento de las responsabilidades en el proceso enseñanza-aprendizaje, a fin de obtener información válida, objetiva y fiable, y, finalmente, determinar los logros alcanzados por los estudiantes [2]. Durante su formación los estudiantes son constantemente sometidos a diferentes tipos de evaluaciones, uno de los indicadores poco tomados en cuenta es el desempeño docente, por esta razón es importante que los educadores sean eficientes para poder tener una educación de calidad.

La Ley Orgánica de Educación Superior (LOES) [3] emitida en agosto del 2018 en su Título VIII, Capítulo II que habla sobre el personal académico, en el artículo 151 titulado Evaluación periódica integral, establece que: Las y los profesores se someterán a una evaluación periódica integral según los establecido en la presente Ley y Reglamento de Carrera y Escalafón del Profesor Investigador del Sistema de Educación Superior, en ejercicio de su autonomía responsable. Se observará entre los parámetros de evaluación la que realicen los estudiantes a sus docentes.

Dentro del marco legal que rige a la Universidad Técnica de Ambato en su resolución 0335-CU-P-2022 [4] el Reglamento de Evaluación del desempeño del Personal Académico y Apoyo Académico de la Universidad Técnica de Ambato. Documento en donde se señalan cuáles son las directrices que rigen al personal dentro de la institución. El cual dentro de la Universidad no pasa de ser un documento al que no se le ha dado la importancia necesaria para cumplir eficazmente con sus estatutos.

1.2. Antecedentes Investigativos

Existen diversos estudios que hasta la actualidad han reforzado diferentes temas a tratar durante el presente trabajo de investigación, entre los cuales se puede resaltar:

Según Tannya Naranjo [5] en su tesis “Sistema Web para el seguimiento del cumplimiento de los contenidos de los sílabos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial” Trabajo realizado como Tesis en el año 2016 en la Universidad Técnica de Ambato concluye que: “Al realizar la evaluación de seguimiento del sílabo efectuada por los estudiantes de una forma automática y sencilla, se gestionará el desempeño de los docentes que imparten los contenidos de las diferentes asignaturas mejorando notablemente el control del sílabo en la Facultad.” Al igual que el seguimiento al sílabo la evaluación docente es uno de los procesos y requerimientos de cada facultad dentro de la Universidad Técnica de Ambato que son necesarios al culminar cada periodo académico, por lo que la automatización de dichos procesos libera de carga académica a los docentes responsables.

Duran et al. [6] Señalan en su investigación titulada “Desarrollo de un sistema web de retroalimentación continua para los docentes de la ESFOT.” (Escuela de Formación de

Tecnólogos de la Escuela Politécnica Nacional) Que es importante realizar una retroalimentación durante el semestre en curso para que sea posible emprender acciones correctivas según el caso lo amerite. Además de que el docente en cuestión podrá conocer sus falencias durante el periodo académico de una manera más acertada debido a que los estudiantes de educación superior no suelen comunicarse.

Olarte et al. [7] en su trabajo titulado “Evaluación docente como factor de desarrollo profesional desde una pedagogía reflexiva” evidenciaron que “dentro de la evaluación docente se generan consideraciones pedagógicas que promueven la comprensión de la evaluación docente como un proceso reflexivo, formativo e integral que favorece el desarrollo profesional de los profesores, los procesos educativos y las prácticas académicas de investigación-acción.”

Por ello se puede señalar al proceso de evaluación docente como un mecanismo que permite no solo evaluar las diferentes directrices que rigen a un docente dentro de una institución de educación superior, sino también como un mecanismo de desarrollo profesional para los diferentes docentes en cada una de sus áreas a cargo.

En [8] Gómez y Valdez se señalan que los cuestionarios son la manera más usada para evaluar a docentes universitarios, en donde a medida que la tecnología crece estos dejan de llenarse manualmente para llenarse online.

La evaluación docente es un componente clave de la educación ya que es la única manera de mantener una educación de calidad y que se adapte al contexto social requerido por los futuros profesionales

Gálvez y Milla [9] en su investigación titulada “Evaluación del desempeño docente: Preparación para el aprendizaje de los estudiantes en el Marco de Buen Desempeño Docente” Señalan que los modelos de evaluación docente pueden incomodar de cierta manera a los docentes ya que estos consideran a su desempeño como un perfil privado. Por lo que es importante esclarecer las políticas internas de la institución ya que la evaluación docente no debería ser considerada como una molestia por parte de los docentes si no como una

herramienta para poder mejorar sus aptitudes docentes y así mejorar la calidad de la educación, y en algunos casos favoreciendo el reconocimiento de su profesión.

1.3. Fundamentación Teórica

Bases de datos no relacionales

También conocidas como NoSQL (No Structured Query Language) son bases de datos diseñadas para patrones de acceso a datos de baja latencia, las bases de datos no relacionales funcionan en base a esquemas que tienen esquemas flexibles en donde se da prioridad al volumen y rendimiento antes que la integridad de los datos. Al igual que las bases de datos relacionales éstas pueden almacenar todo tipo de datos como, cadenas, enteros, documentos e imágenes, todo esto mediante un sistema de clave valor, documentos, grafos u orientadas a columnas [10].

MongoDB

MongoDB es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado. El modelo de documentos de MongoDB resulta muy fácil de aprender y usar, y proporciona a los desarrolladores todas las funcionalidades que necesitan para satisfacer los requisitos más complejos a cualquier escala. Se proporcionan drivers para más de diez lenguajes, y la comunidad ha desarrollado varias decenas más [11].

ECMAScript

ECMAScript es la especificación del lenguaje (JavaScript) propuesta por ECMA International, que es una institución encargada de diferentes estándares TC39 es la organización encargada de recopilar, pulir e implementar las diferentes características que incluso pueden ser añadidas por la comunidad después de un extenuante proceso de pruebas, ECMAScript 2023 o ES14 es la última versión disponible a la fecha de esta investigación.[12]

JavaScript

Es un lenguaje de programación de alto nivel que se puede definir como la implementación de las características y reglas, propuestas por ECMA International. JS como es conocido popularmente es un lenguaje interpretado, al no necesitar compilación (transformarse en lenguaje máquina) su ejecución es rápida y permite ejecutarse en varios entornos.[13]

Express

Express es el framework más popular de NodeJs, es una librería de trabajo para entornos de desarrollo web, es de código abierto y se distribuye bajo la licencia MIT (Massachusetts Institute of Technology). Posee varias características que hacen de esta una de las librerías más adecuadas para trabajar con servicios web, APIS (Application Programming Interface) y protocolos de comunicación web [14].

React

Es una biblioteca desarrollada por Facebook que está orientada al desarrollo front-end web, esta implementado de manera declarativa y basado en componentes, además de ser uno de los frameworks más utilizados permite escalar aplicaciones de gran escala debido a su flexibilidad y retrocompatibilidad con varios navegadores. Además, incorpora 12 métodos propios que son conocidos como hooks que facilitan el desarrollo web debido a que le dan solución a varios problemas cotidianos [15].

Contenedores

Es una tecnología que permite interactuar directamente con el kernel del sistema operativo en donde se crean un ambiente simulado mucho más rápido que una máquina virtual y que permite empaquetar diferentes servicios con alto grado de complejidad. Según la Developer Survey realizada por Stack Overflow en el año 2022 [16] Docker y Kubernetes son las herramientas más populares en la actualidad y estas permiten crear arquitecturas basadas en servicios de manera aislada, para así empaquetar todo el software como: plataforma, front-end, back-end, base de datos, redes, sistemas de almacenamiento dentro de un contenedor que facilita en gran medida el transporte y despliegue de plataformas en servidores [17].

MERN

Se conoce como un stack de tecnologías a la agrupación de estas que coexisten dentro de un mismo proyecto con características que se benefician entre sí, MERN es el acrónimo de MongoDB, Express, React y NodeJs [18].

En donde:

- MongoDB – Base de datos.
- ExpressJs – Framework web de NodeJs
- ReactJs – Librería de JavaScript para crear interfaces web.
- NodeJs – Entorno de ejecución de JavaScript fuera del navegador.

Responsive

Con el exponencial crecimiento del mercado de los dispositivos de consumo multimedia como smartphones, tablets, computadoras, televisiones inteligentes y centros de entretenimiento, un gran problema para los desarrolladores surge. Y es que todos estos dispositivos se inventan según la visión del futuro de cada empresa tecnológica, de esa manera es común encontrar en la actualidad dispositivos muy compactos con pantallas muy pequeñas, como también dispositivos con pantallas relativamente grandes, es difícil establecer un estándar sobre el tamaño de las pantallas de los dispositivos móviles ya que son diferentes las necesidades de los usuarios.

Según Filippova [19] el crecimiento de usuarios de dispositivos móviles se ha ido incrementando con el paso de los años, lo que para los usuarios supone un avance tecnológico importante, ya que pueden conectarse a los servicios de la red desde cualquier dispositivo en cualquier momento, los sitios web que son desarrollados de manera tradicional no proporcionan una adecuada experiencia de usuario. Como se puede evidenciar en la figura 1 durante la última década el número de usuarios móviles web se ha ido incrementando con mayor rapidez en relación con el número de usuarios de escritorio.

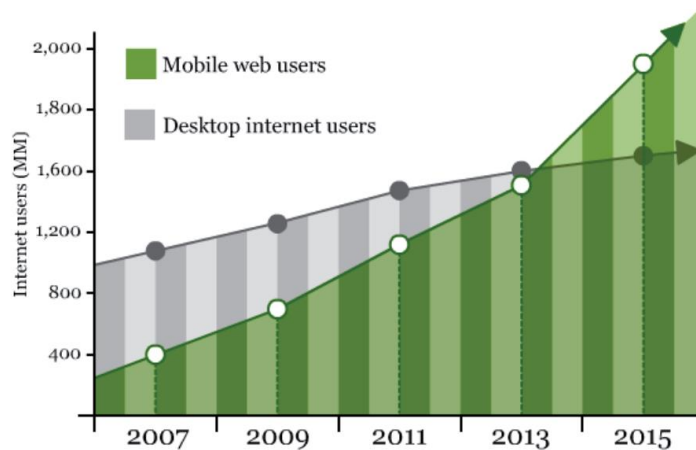


Figura 1. Comparación de usuarios móviles vs de escritorio

Autor: Filippova L. & Svidelskyi R. [19]

El concepto de responsive o responsivo se ha ido acuñando con el pasar del tiempo y hace referencia a la capacidad que tienen las aplicaciones (en este caso web) para adaptar su contenido a la resolución de la pantalla del dispositivo desde el cual se está accediendo, todo esto manteniendo una correcta relación entre los elementos gráficos para así obtener una óptima experiencia de usuario. Existe una popular metáfora dentro de la comunidad de programadores alrededor del mundo en donde al hablar de sistemas responsive se refieren al contenido de las aplicaciones como si fuera agua, y esta toma la forma del recipiente que la contiene, en este caso el tamaño de la pantalla del dispositivo.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un sistema web utilizando contenedores y el stack MERN para mejorar la experiencia de usuario en el Sistema de Evaluación Docente de la Universidad Técnica de Ambato.

1.4.2. Objetivos Específicos

- Analizar la Experiencia de Usuario en el sistema de Evaluación Docente de la Universidad Técnica de Ambato durante el periodo Octubre 2022 – Marzo 2023.

- Determinar la metodología de desarrollo que más se adapte al proyecto.
- Diseñar una arquitectura de Software que se adapte a las necesidades de la Plataforma.
- Desarrollar un sistema web que se adapte a las necesidades de la Plataforma.

CAPÍTULO II.- METODOLOGÍA

2.1. Materiales

Debido a la naturaleza del proyecto la recolección de la información se realizó con la ayuda de formularios de preguntas cerradas, entrevistas a estudiantes y fichas de observación.

Encuesta Estudiantes

Como se observa en el Anexo 1, el formulario dirigido a los estudiantes consta de 11 preguntas cerradas que tuvieron como finalidad recolectar opiniones de los estudiantes de la Universidad Técnica de Ambato sobre el Sistema de Evaluación docente y cuan involucrados se ven los estudiantes en dicho proceso, para que las opiniones de los estudiantes sean más reales no se recolectó datos personales.

Ficha de Observación

Como se observa en el Anexo 2, la ficha en cuestión consta de 8 apartados donde se evalúan las diferentes características relacionadas la experiencia de usuario (UX) y la interfaz de usuario (UI) de la plataforma, estos ítems son evaluados mediante una escala del 1 al 5, además la ficha cuenta con un apartado de observación en donde se podrán detallar cuestiones que se salgan de la escala principal de evaluación.

2.2. Métodos

La investigación tuvo un enfoque mixto, se recolectaron y analizaron datos de los estudiantes referentes a su interacción con el Sistema de Evaluación Docente, además se realizó investigación de campo referente al Funcionamiento del Sistema de Evaluación Docente e investigación Bibliográfica Documental.

2.2.1. Modalidad de la Investigación

Investigación de Campo

La modalidad de campo fue utilizada debido a que el contexto de la investigación tiene lugar en la Universidad Técnica de Ambato, donde todos los actores que intervienen con el sistema de Evaluación Docente tienen como punto común el campus Universitario.

Investigación Bibliográfica-Documental

Esta modalidad ayudó a recolectar información del estado del arte referente a UX (Experiencia de Usuario) y UI (Interfaz de Usuario) y poderla contrastar con las prácticas que se toman a la hora de implementar el actual Sistema de Evaluación Docente.

2.2.2. Población y muestra

Se toman en consideración para la recolección de la información a todos los estudiantes de la Universidad Técnica de Ambato que se encuentren matriculados en el periodo académico Octubre 2022 – Marzo 2023. Los cuales se toman como referencia del portal universitario en su encuesta anual en donde se señala que el número de estudiantes es de 18596 [20].

Tabla 1. Población con la que se realizó la investigación.

Autor: David Constante

Tipo de Población	Descripción	Número	Porcentaje
Estudiantes	Estudiantes Matriculados con acceso al sistema integrado de la Universidad.	18596	100%
Total		18596	100%

Dentro de la rama del desarrollo de software se utiliza por convención el parámetro de 5% de error y 95% de nivel de confianza. Tajpour y Hosseini utilizan estos parámetros para evaluar características a nivel de startups. Debido a que el tiempo es uno de los recursos que más limitan el desarrollo del proyecto y que el contexto de la plataforma planteada no engloba ningún área crítica como la salud o la economía se optó por utilizar un error del 10% para calcular la muestra de estudiantes necesarios. Todo esto con la finalidad de generar un Producto Viable Mínimo (MVP) lo más pronto posible para poder realizar los ajustes necesarios de ser el caso.

Ya que la población de estudiantes excede a los 100 individuos se calcula el tamaño de la muestra con la siguiente ecuación:

$$n = \frac{N\sigma^2 Z^2}{(N - 1)e^2 + \sigma^2 Z^2}$$

Donde:

n = Tamaño de la muestra

N = Población: 18596

Z = Nivel de Confianza 95% Equivalente al 1.96

σ = Desviación estándar constante: 0.5

e = Margen de error: 0.1

$$n = \frac{18596 * 0.5^2 * 1.96^2}{18595 * 0.1^2 + 0.5^2 * 1.96^2}$$

$$n = 95.55$$

Por lo que la muestra de estudiantes se puede aproximar a 96 Estudiantes.

2.2.3. Recolección de la Información

Para aplicar el formulario a los estudiantes de la Universidad se utilizó los formularios de Google, debido a que el alcance y tabulación son facilitadas mediante la misma plataforma.

Validación del instrumento

Con la finalidad de que la recolección de la información sea lo más cercana a la realidad se utiliza el Alfa de Cronbach para validar el instrumento.

Tabla 2. Varianza de los resultados obtenidos en la encuesta realizada

Autor: David Constante

Pregunta	Varianza del Ítem	Varianza del Instrumento
1	0.1074	4.86
2	0.1355	
3	0.1074	
4	0.1701	
5	0.1590	
6	0.1343	
7	0.0000	
8	0.1119	
9	0.1607	
10	0.1212	
11	0.1320	
Total	1.340	

Debido a la pregunta 7 tiene un alto grado de incertidumbre no se la toma en cuenta para realizar el cálculo.

$$\alpha = \frac{K}{K - 1} \left[1 - \frac{\sum S^2}{S^2} \right]$$

Donde:

K es el número de ítems del instrumento: 10

$\sum S^2$ es la sumatoria de la varianza de los ítems: 1.340

S^2 es la varianza total del instrumento: 4.86

$$\alpha = \frac{10}{10 - 1} \left[1 - \frac{\sum 1.34^2}{4.86^2} \right]$$

$$\alpha = 0.81$$

Debido a que el Alfa de Cronbach tiene un valor de 0.81, se puede considerar que está dentro de los rangos aceptables del instrumento.

Resultados de la encuesta realizada a Estudiantes

La encuesta fue aplicada a un total de 107 estudiantes de la Universidad Técnica de Ambato.

Pregunta 1:

¿Está familiarizado con el código de ética de la Universidad Técnica de Ambato?

107 respuestas

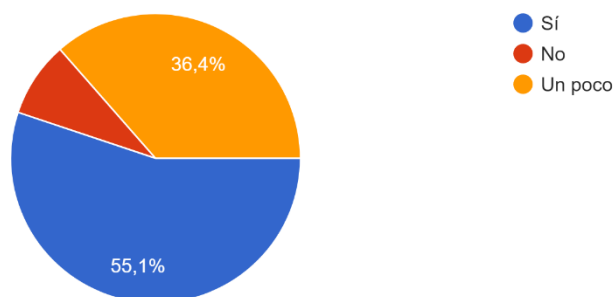


Figura 2. Tabulación de resultados pregunta 1.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 2, el 55.1% de los estudiantes que corresponderían a más de la mitad de los encuestados están familiarizados con código de ética de la Universidad, un 36.4% está un poco familiarizado y tan solo un 13.1% no conoce el código de ética. Lo cual sugiere que a pesar de que en su mayoría los estudiantes conocen dicho estatuto, una gran parte de la comunidad estudiantil no conocen el reglamento en profundidad, por consiguiente, al no conocer sus derechos y obligaciones los estudiantes no podrán desenvolverse adecuadamente dentro de su formación universitaria.

Pregunta 2:

¿Sabía usted que el código de ética también le exige cierto comportamiento dentro de la Universidad a sus docentes?

107 respuestas

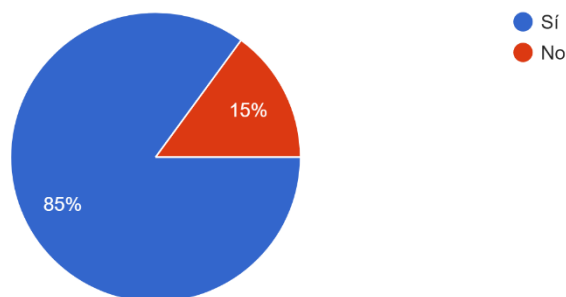


Figura 3. Tabulación de resultados pregunta 2.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 3, el 85% de los encuestados conocen que los docentes también están regidos por el reglamento de la universidad, mientras que el 15% de los mismos desconoce de dicha situación. Lo que evidencia que en su mayoría los estudiantes conocen las acciones que los docentes pueden y deben realizar dentro de su rol como educadores. Es importante recalcar que el 15% de los estudiantes que no están familiarizados con el código que rige a los docentes representan alrededor de 2700 estudiantes, según datos disponibles en el portal universitario esta cantidad equivaldría a una facultad y media, que este caso no podría exigir a sus docentes el comportamiento adecuado debido al desconocimiento.

Pregunta 3:

¿Está usted familiarizad@ con el sistema de Evaluación docente?

107 respuestas

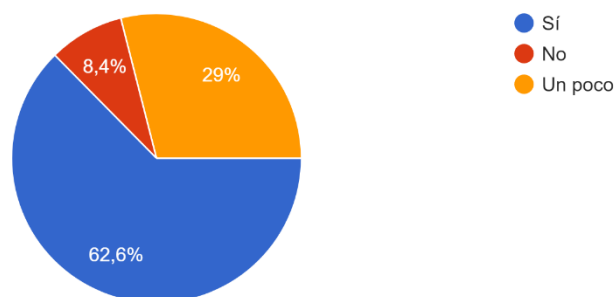


Figura 4. Tabulación de resultados pregunta 3.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 4, el 62.6% de los estudiantes conocen el Sistema de Evaluación Docente lo que es algo común dentro de estudiantes ya que es uno de los requisitos que los docentes requieren de sus alumnos para que la nota del parcial se vea reflejada en el Sistema Integrado, con el 29% de los estudiantes que solo están familiarizados un poco con el sistema se puede inferir que no le dan la importancia necesaria a la evaluación y que muchas veces es llenada como una obligación impuesta por el docente de turno y, mas no como una herramienta que sirve para evaluar el desempeño del mismo. Por otro lado, el 8.4% de los estudiantes que no están familiarizados con el sistema pueden pertenecer al grupo anterior o a estudiantes de primer semestre que no conozcan aun el proceso.

Pregunta 4:

¿Le parece cómodo el acceso al sistema de evaluación docente ?

107 respuestas

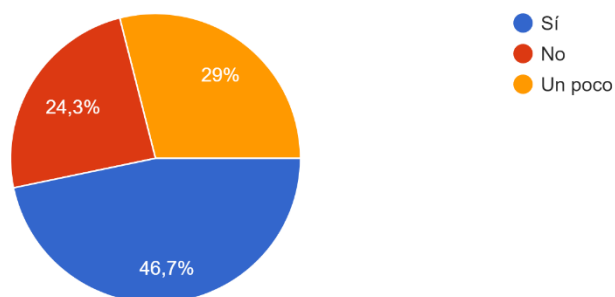


Figura 5. Tabulación de resultados pregunta 4.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 5, el 46.7% de los estudiantes están conformes con el acceso al sistema, más de la mitad de los estudiantes encuestados están en desacuerdo con la ruta que hay que seguir para acceder al Sistema de Evaluación Docente, el 29% que esta algo conforme y el 24.3% que no están de acuerdo con el acceso al sistema, evidencian la complejidad para acceder al mismo, lo cual hace que el sistema no pueda cumplir con su cometido para el cual fue diseñado.

Pregunta 5:

¿Ha perdido alguna materia por culpa de la metodología de enseñanza de un docente ?

107 respuestas

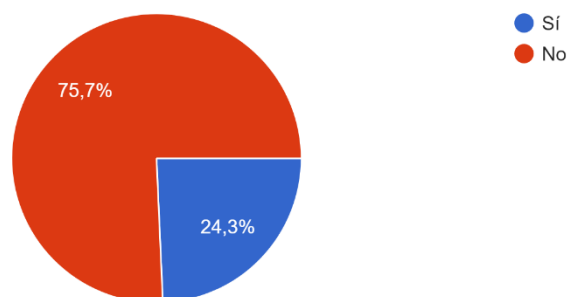


Figura 6. Tabulación de resultados pregunta 5.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 6, el 75.7% de los estudiantes no ha perdido ninguna materia por culpa de la metodología de enseñanza de un docente, pero el 24.3% sí, lo que sugiere que al menos 4500 estudiantes han perdido una materia debido a la metodología utilizada por sus docentes durante cierto periodo académico. Por lo que la metodología del docente puede estar sujeta a una evaluación por parte de los estudiantes que están inscritos en la asignatura impartida por el docente.

Pregunta 6:

¿Sus docentes adaptan la metodología de enseñanza a medida que avanza el periodo académico?

107 respuestas

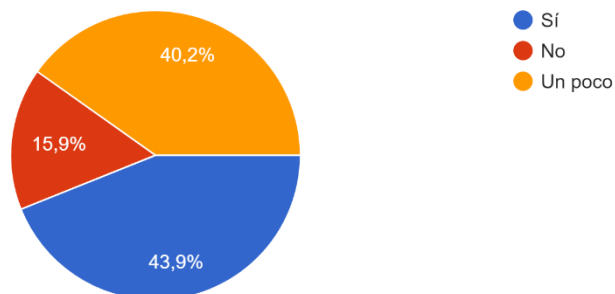


Figura 7. Tabulación de resultados pregunta 6.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 7, el 43.9% de los docentes afirma que los docentes adaptan su metodología de clase, mientras que el 40.2% de los estudiantes afirman que los docentes adaptan un poco sus metodologías, y el 15,9% de los mismos está de acuerdo con que sus docentes no adaptan sus metodologías de enseñanza. Es importante señalar que los cambios generacionales y desastres globales hacen que las generaciones académicas sean diferentes unas de otras, por lo que, si la excelencia académica es el objetivo de la Universidad, los docentes deberían adaptar sus tipos de enseñanza de acorde a su clase. Y en este caso menos de la mitad lo hace adecuadamente.

Pregunta 8:

¿Su docente es transparente con las calificaciones?

107 respuestas

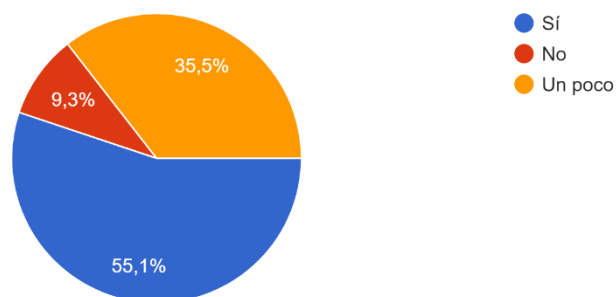


Figura 8. Tabulación de resultados pregunta 8.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 8, el 55.1% de los estudiantes afirma que los docentes son transparentes con el sistema de calificaciones, el 35% y 9% de los estudiantes afirman que la asignación del aprovechamiento no es del todo transparente, lo que cual está en contra del reglamento universitario en donde se especifica que las diferentes secciones de aprovechamiento deben ser respetadas.

Pregunta 9:

¿Alguna vez ha sentido que sus docentes tienen cierto favoritismo hacia algunos estudiantes?

107 respuestas

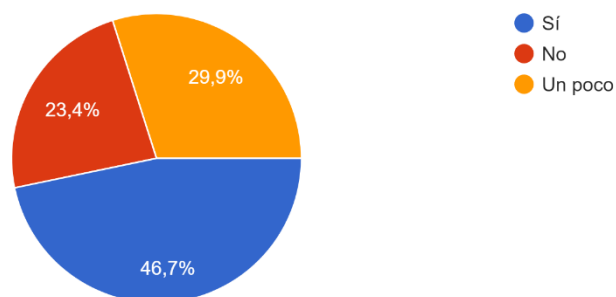


Figura 9. Tabulación de resultados pregunta 9.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 9, se puede evidenciar que el 76.6% de los estudiantes de cierta manera se ha sentido afectado debido al favoritismo de los docentes hacia algunos de sus compañeros, esto tiene una relación con el <grafico tal> que casi la mitad de sus docentes no son del todo transparentes con sus estudiantes. Aun que el 23.4% de los estudiantes no han tenido que enfrentarse a una situación de este tipo es una realidad que afecta a 3 de cada 4 estudiantes.

Pregunta 10:

¿Se siente satisfecho con el desempeño de sus docentes durante el periodo académico?

107 respuestas

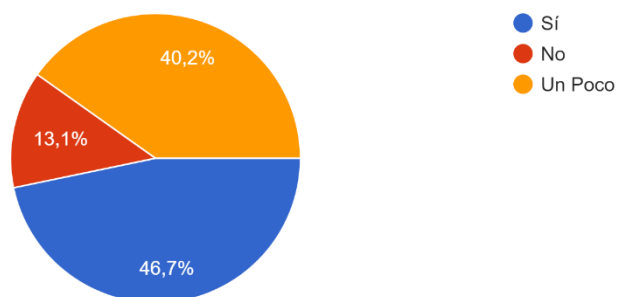


Figura 10. Tabulación de resultados pregunta 10.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la Figura 10, el 46.7% de los estudiantes está conforme con el desempeño de sus docentes, pero el 40.2% que afirma que está un poco satisfecho con sus docentes y el 13.1% de los estudiantes que afirman que no están satisfechos, sumados evidencian que más de la mitad de los estudiantes no están totalmente conformes con sus educadores, lo que si bien no está del todo mal, en una institución que tiene por objetivo ser una de las mejores de la región y el país es una cuestión que debe ser manejada.

Pregunta 11: Esta pregunta esta evaluada en la escala de 1= Poco a 5 = Mucho.

¿En que medida siente que la metodología de enseñanza de sus docentes afecta su salud mental?

107 respuestas

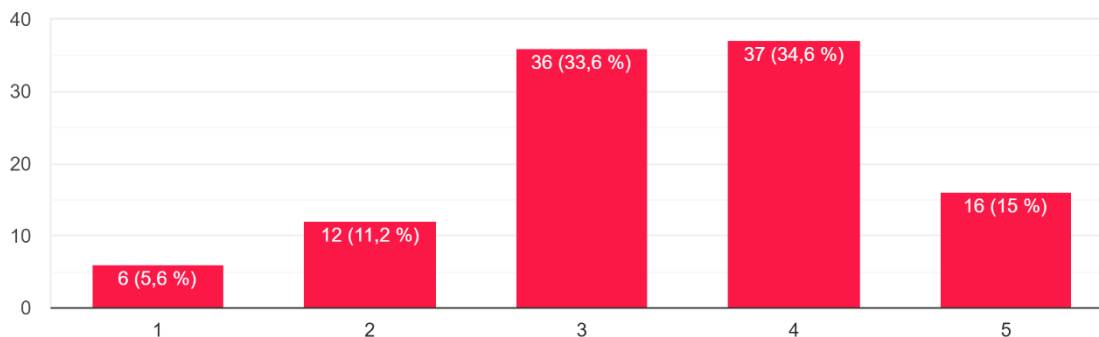


Figura 11. Tabulación de resultados pregunta 11.

Autor: David Constante

Análisis e interpretación de resultados

Según los resultados de la ilustración Figura 11 la tendencia está marcada a que la salud mental de los estudiantes se vea medianamente afectada a muy afectada y eso afectara de manera directamente relacional al nivel de aprendizaje dentro de las aulas de clase y por ende a la calidad de educación y formación de los futuros profesionales.

Resultados ficha de observación al Sistema de Evaluación Docente

Tabla 3. Resultados de la ficha de observación.

Autor: David Constante

N	DESCRIPCIÓN	1	2	3	4	5	OBSERVACIÓN
1	Facilidad de Acceso al sistema				X		El sistema se encuentra dentro sistema integrado de la Universidad por lo que se requieren varios clics para acceder al sistema.
2	Certificados se seguridad	X					La página no tiene certificado https, por lo que muchos navegadores la detectan como un sitio inseguro.
3	Renderizado personalizado de los elementos		X				Los componentes de Usuario son genéricos, no validan la cuenta autenticada.
4	Errores de consola		X				La página presenta advertencias en la consola del navegador, lo que puede suponer fallos de seguridad dentro de la misma, además existen elementos que no se pueden cargar adecuadamente.
5	Actualización del Sistema	X					Pese a haber iniciado un nuevo periodo académico las fechas con respecto a las evaluaciones

							pertenecen al periodo académico anterior.
6	Sistema Responsive				X		Solo algunos componentes no reaccionan adecuadamente al tamaño de la pantalla.
7	Efectividad del Sistema		X				A dos semanas de iniciado el periodo académico el sistema está restringido lo que dificulta la correcta evaluación docente
8	Líneas de diseño			X			El sistema no sigue las líneas de diseño del sistema integrado, tiene el aspecto de un sistema diferente.

El Sistema de Evaluación Docente podría pasar por un MVP (Mínimo Producto Viable), pero no como un sistema ya desplegado que cumple los requerimientos de una institución, además de las funcionalidades del sistema presentan redundancias o limitaciones presenta severas fallas de seguridad que podrían ser utilizadas para escalar privilegios y atacar a la totalidad del sistema integrado.

2.2.4. Procesamiento y Análisis de Datos

En base a la información recolectada mediante las encuestas y fichas de observación, una vez analizada objetivamente la información se determina que:

- Todos los estudiantes en su mayoría conocen el comportamiento y límites en los cuales se pueden manejar dentro de la Universidad Técnica de Ambato.
- Un número de estudiantes que es equiparable al número de estudiantes que conforman alrededor de una facultad de la Universidad Técnica de Ambato no conoce las

obligaciones de sus docentes, por lo que sus estudiantes no pueden exigir el comportamiento esperado de sus docentes.

- Los estudiantes en su mayoría no conocen el proceso ni objetivo por el cual se realiza la evaluación docente.
- La ruta para acceder al Sistema de Evaluación docente no es cómoda a nivel de Experiencia de Usuario.
- Los docentes no se ven involucrados en el proceso de aprendizaje de sus alumnos, más de la mitad de ellos no se adaptan a sus grupos de clase.
- Muchos estudiantes sienten cierto favoritismo hacia algunos de sus compañeros lo que lleva a cabo con mucha facilidad al no ser totalmente transparente la asignación de notas a los estudiantes.
- Más de la mitad de los estudiantes en la Universidad Técnica de Ambato no se sienten del todo satisfechos con sus docentes.
- La salud mental es un factor muy fuerte dentro de la vida de los estudiantes, el cual tiene una relación con el desempeño de los docentes.
- El actual Sistema de Evaluación docente no cumple con los principios de usabilidad, no sigue las líneas de diseño de los demás sistemas de la Universidad y presenta varios errores en su funcionamiento, e incluso algunos pueden representar fallos significativos de seguridad.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

Determinación de la metodología de Desarrollo

El desarrollo de software es un proceso que nace desde un lienzo en blanco, existen muchas maneras en las que se pueden organizar y gestionar los flujos de trabajo para que el producto final cumpla con todos los requerimientos iniciales, con el paso de los años se han ido acuñando diferentes metodologías de desarrollo como: Extreme Programming (XP), Scrum o Kanban que se han ido adaptando a las nuevas necesidades de los desarrolladores.

Existe una extensa cantidad de literatura que habla sobre las diferentes metodologías de desarrollo de software, en mencionados artículos existe una gran variedad de comparaciones sobre las diferentes metodologías. Popova [21] sugiere que es importante adaptar una metodología de desarrollo para controlar los diferentes aspectos del desarrollo del sistema, la planificación y monitoreo ayudan a dividir adecuadamente el trabajo para cumplir con los objetivos planeados.

Según el 13vo informe anual sobre el estado de las metodologías se afirma que: La mayoría de las empresas (95%) expresan que al menos uno de sus proyectos ágiles ha tenido éxito, mientras que, casi la mayoría (48%) afirman que la mayoría de sus proyectos ágiles han presentado mejoras. El informe también señala la tendencia de las empresas a adoptar una metodología de desarrollo ágil [22].

Una vez señalada la importancia de adoptar una metodología de desarrollo durante el proceso de desarrollo de un sistema, la incertidumbre se torna hacia la pregunta ¿Qué metodología de desarrollo de software utilizar? Las metodologías se adoptan en función del tamaño del proyecto, recursos, tiempo, etc. Donde la facilidad de interacción con otro u otros miembros del equipo de trabajo agiliza la entrega de un proyecto de software, siempre y cuando exista más de un miembro en el equipo de trabajo.

Entre las metodologías más populares se pueden destacar:

Extreme Programming

La metodología XP es usada por pequeños equipos que necesitan completar un proyecto en un periodo corto de tiempo, esta metodología se caracteriza por pequeñas entregas funcionales representadas por historias de usuario que pueden estar basadas en Test Driven Development o TDD, lo que garantiza que estos pequeños entregables sean parte de un todo funcional. El éxito depende de cuan bien este diseñado el proceso completo, la Planificación, Diseño, Codificación, Pruebas y Lanzamientos garantizan el éxito o fracaso a la hora de cumplir con un objetivo. Todas estas fases tienen un esquema cíclico, como la figura 12 señala una característica puede volver al punto de partida y reevaluar nuevos parámetros dentro de su implementación en el caso de ser necesario, para así poder tener una constante retroalimentación. Una de las desventajas de XP es que tiene un costo elevado a la hora de adaptar el proyecto en caso de una modificación del equipo de trabajo o requerimientos [21].

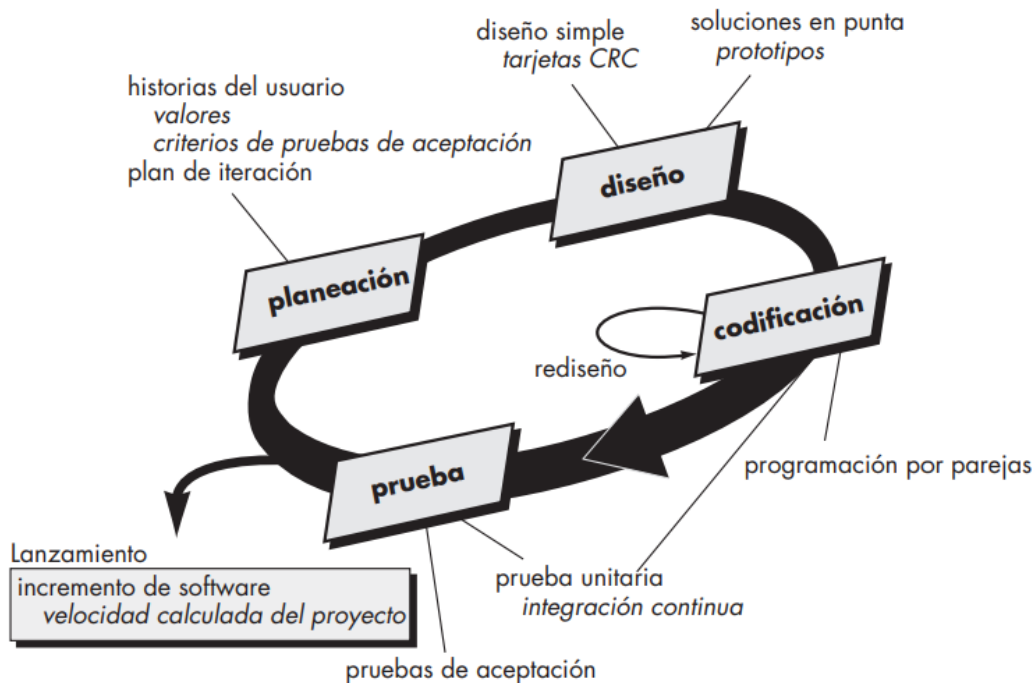


Figura 12. Flujo de la metodología xp

Autor: Leonardo Alvarado [23]

KANBAN

KANBAN por otra parte es una metodología de desarrollo que se basa en el principio “just in time” o justo a tiempo, que tiene como finalidad cumplir con los plazos acordados por el cliente y el equipo de desarrollo, a diferencia de otras metodologías de desarrollo Kanban tiene su origen en las instalaciones de Toyota en donde etimológicamente se traduce del japonés como tablero visual o tarjetas visuales [24]. El proceso KANBAN no tiene un flujo como tal por lo que su estructura se basa en tareas acomodadas en columnas dentro de un tablero que como tal no siguen un flujo de trabajo, más bien siguen un conjunto de seis reglas que se pueden visualizar en la figura 13 las cuales garantizan el correcto funcionamiento del proceso. [25]

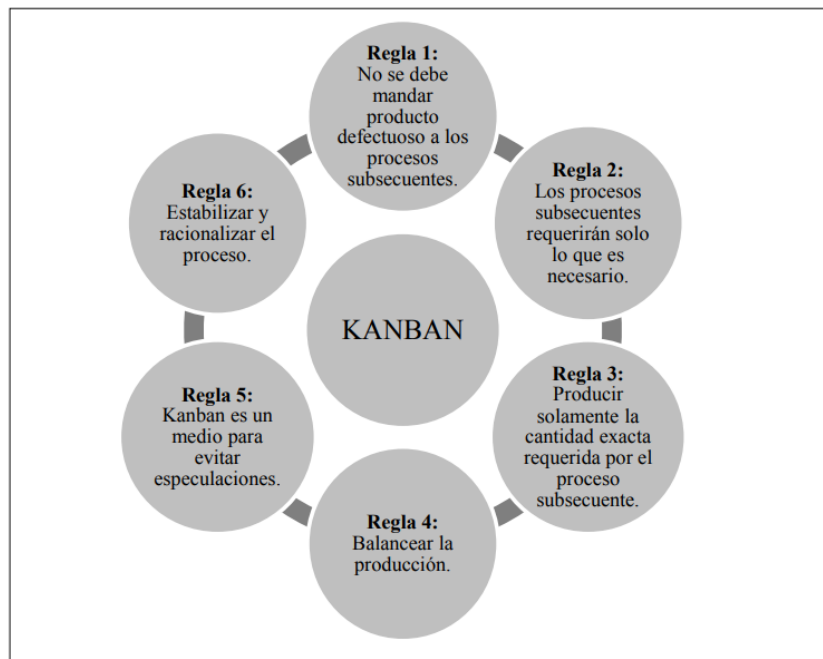


Figura 13. Reglas en las que se basa la metodología Kanban

Autor: Martin Arango et al.[25]

SCRUM

SCRUM es otra de las metodologías ágiles más utilizadas, utiliza un patrón de proceso llamado sprint que se puede definir como una unidad de trabajo que engloba subtarefas que convergen en una funcionalidad específica, cada sprint o tarea puede variar en función de la complejidad y tamaño del producto, tiene la facultad de adaptarse en tiempo real a medida que el proyecto se reestructure. Tiene un seguimiento constante mediante las reuniones SCRUM o Daily Meeting que tienen por finalidad tener un feedback sobre el equipo, las funcionalidades y problemas con los que se viesen involucrados los integrantes del equipo. [23]. Como se puede evidenciar en la figura 14 el flujo de trabajo de SCRUM al igual que XP también tiene una forma cíclica, pero en este caso los ciclos de trabajo son más extensos y concurrentes.

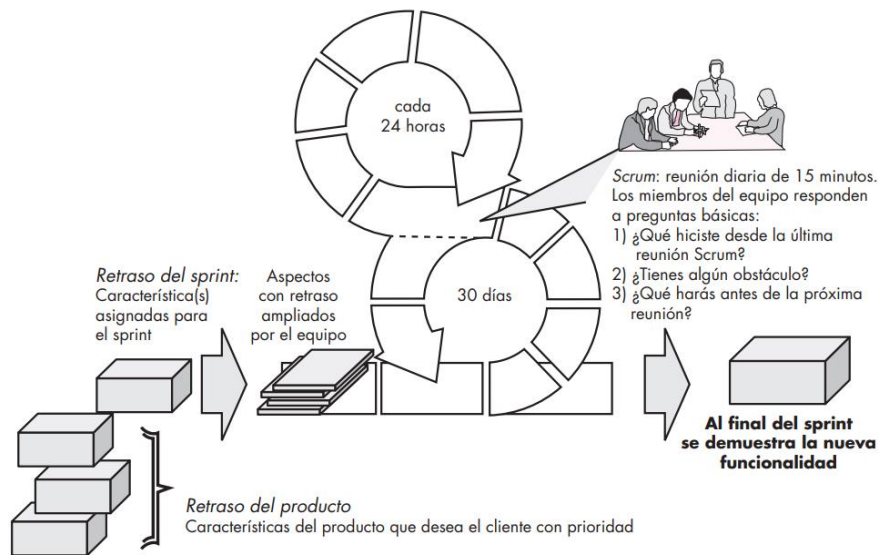


Figura 14. Flujo de la metodología SCRUM

Autor: Leonardo Alvarado [23]

Evaluación de Metodologías

Las metodologías señaladas anteriormente son las que han ido ganando popularidad con el pasar de los años dentro del mundo del desarrollo de software, y si bien se podría desarrollar la plataforma con cualquiera de estas, la mala planificación podría incurrir a un desperdicio de recursos durante toda la etapa de desarrollo de la plataforma. También resulta complicado seleccionar una metodología tradicionalmente definida, debido a que las mismas poseen individualmente características favorables para el desarrollo, por consiguiente, se realizó una tabla de criterios ponderados en donde se evalúan por separado las diferentes características de las diferentes metodologías de desarrollo de software para así poder seleccionar las más relevantes para los stakeholders.

Tabla 4. Criterios ponderados características metodologías

Autor: David Constante

Factores	Peso Relativo (%)	Alternativas		
		XP	SCRUM	KANBAN
Planificación	20	18	15	5
Test de Código	10	10	5	5
Despliegue	10	10	5	8
Organización	20	10	18	15
Informes	10	8	10	1
Manejo de Tareas	30	25	25	20
Puntuación Total	100	81	78	54

Si bien es cierto que la metodología de Extreme Programming es la que más se adapta a las necesidades del proyecto las metodologías SCRUM y KANBAN poseen características que pueden ser de un alto valor para el desarrollo del proyecto, por lo que la opción utilizada durante la elaboración del presente proyecto fue la utilización de una metodología de desarrollo híbrida que tomara como punto de partida la metodología XP a la cual se le añadieron las características más relevantes de SCRUM y KANBAN, como se detalla a continuación.

Definición de Arquitectura Híbrida

Una vez recolectada suficiente información se puede concluir que las metodologías no son estatutos que deben ser aplicados al pie de la letra, por el contrario, son convenciones de trabajo que deben ser adaptadas según las necesidades de la organización para crear el proyecto con los recursos que tiene disponible esta. Es así que se tomó como punto de partida la metodología de Extreme Programming, a la cual se le añadió características de Scrum como la implementación de un backlog como banco de tareas principal, las cuales se dividieron en grupos según la funcionalidad que se desarrolló, esto permitió planificar sesiones de trabajo cortas que permitían crear una funcionalidad completa que podía ser integrada a medida que esta fuese completada, de esta manera se mantiene del principio de integración continua. Además, la implementación del tablero de KANBAN hizo posible el monitoreo constante del desarrollo de la plataforma para poder seguir una secuencia que vaya de acuerdo con lo planificado en la hoja de ruta.

Algunas de estas características fueron sometidas a una adaptación de acuerdo con las necesidades y recursos disponibles, las mismas se detallan en la tabla a continuación.

Tabla 5. Metodología híbrida

Autor: David Constante

Característica	Origen	Descripción	Motivo	Adaptación
Etapas	XP	Planificación, diseño, codificación, pruebas, lanzamiento.	Planificación general del proyecto.	XP requiere de la participación del cliente, por lo que los entregables en la fase de lanzamiento no son tomados en cuenta. Se reemplaza el diseño y las tarjetas CRC ya que no se desarrolla bajo el paradigma POO, en su defecto se especifican los mockups con sus respectivas funcionalidades.
TDD	XP	Desarrollo basado en pruebas.	Dado que es una metodología ágil se requiere automatizar test para estresar el sistema y garantizar su funcionamiento en entornos no controlados.	Solo se realizan test en las funcionalidades completas.

CI	XP	Las características se integran a medida que se completan.	Desarrollo progresivo del producto final.	
Backlog	SCRUM	Lista general de todas las tareas pendientes.	Visualización general de los requerimientos.	
Sprint	SCRUM	Definir las actividades para completar una tarea en específico en un plazo de tiempo.	Cumplimiento de Objetivos claros en la fase de desarrollo.	Los Sprints tendrán una duración de entre dos días y tres semanas.
Hoja de ruta	SCRUM	Diagrama de tareas y fechas límite.	Visualización de tareas necesarias para el desarrollo de tareas posteriores.	

Tablero	KANBAN	Tablero visual de 3 columnas: Pendiente, en curso, finalizado.	Visualización general del curso de las tareas	Se añade una columna más al tablero para especificar las funcionalidades que están en etapa de Testing.
---------	--------	--	---	---

Todas las metodologías de desarrollo tienen por objetivo agilizar y gestionar de manera óptima los recursos utilizados, el personal y el tiempo son factores clave dentro del desarrollo de software, ya que la distribución de estos tendrá un impacto significativo en la entrega final del producto o aplicación. La idea de implementar una metodología de desarrollo en su mayoría nace de la dificultad de gestionar varios equipos de trabajo y que todos tengan un objetivo en común, en este caso la investigación y desarrollo del proyecto se realizó mediante un solo equipo de trabajo conformado por el Investigador y su docente tutor, es por ello que a pesar de que las metodologías de desarrollo convencionales pretendan agilizar y gestionar eficazmente el flujo de trabajo, dadas las circunstancias y naturaleza del proyecto estas por el contrario añadirían una carga de trabajo innecesaria y que sobre todo no genera algún tipo de valor agregado.

El planteamiento antes mencionado no desmerita el uso de metodologías de desarrollo, estas tienen técnicas y herramientas que pueden ser utilizadas a conveniencia y si es que las circunstancias lo ameritan. De esta manera generar una metodología de desarrollo híbrida permite recolectar las técnicas que brindan mejores resultados al desarrollo de la aplicación.

Como base de la metodología híbrida se toma a la metodología de Extreme Programming, esta presenta valores concisos a la hora de gestionar proyectos con un reducido margen de tiempo, uno de los inconvenientes encontrados dentro de esta metodología es su fase de diseño, ya que esta sugiere que el único producto de la fase de diseño son las tarjetas Clase Responsabilidad Colaboración o también conocidas como CRC. Como su nombre lo afirma dentro de la palabra “Clase” estas tarjetas son una herramienta utilizada dentro del desarrollo de software orientado a objetos, al utilizarse el paradigma de programación reactivo que está basado en eventos; no tiene sentido usar tarjetas CRC. En su defecto en la fase de diseño se generan mockups con la descripción de las funcionalidades y elementos más relevantes, así como las reglas de negocio para el funcionamiento de la aplicación.

Otra de las características tomadas de la metodología XP es el Test-Driven Development que es una característica que le añade robustez a la aplicación. Además, se utilizó la integración continua para probar recurrentemente la funcionalidad total de la aplicación.

Por parte de la metodología Scrum se utilizó el Backlog para generar las tareas necesarias que aplicación necesita para funcionar en su totalidad, tener un banco de tareas permite agrupar las tareas en Sprints, los cuales tienen como objetivo desarrollar una tarea en específico y por otra parte los Epics permiten agrupar Sprints que cumplan una funcionalidad, que a su vez permite generar la hoja de ruta que da una visión completa del estado del proyecto.

Para tener un correcto manejo de las tareas pendientes, en proceso o testeadas, se utilizó el tablero Kanban con una ligera adaptación, generalmente este tiene tres columnas, pero debido a que la metodología híbrida plantea el uso de TDD, el tablero Kanban tiene una columna más enfocada en el testing.

Flujo de Trabajo

Para el monitoreo y control de proyecto se utilizó la plataforma JIRA en su versión gratuita, esta plataforma permite manejar todo el proyecto de manera online y partir desde plantillas que pueden ser modificadas para generar un esquema de trabajo personalizado. Tomando en cuenta todas las características recolectadas el flujo de trabajo es el siguiente.

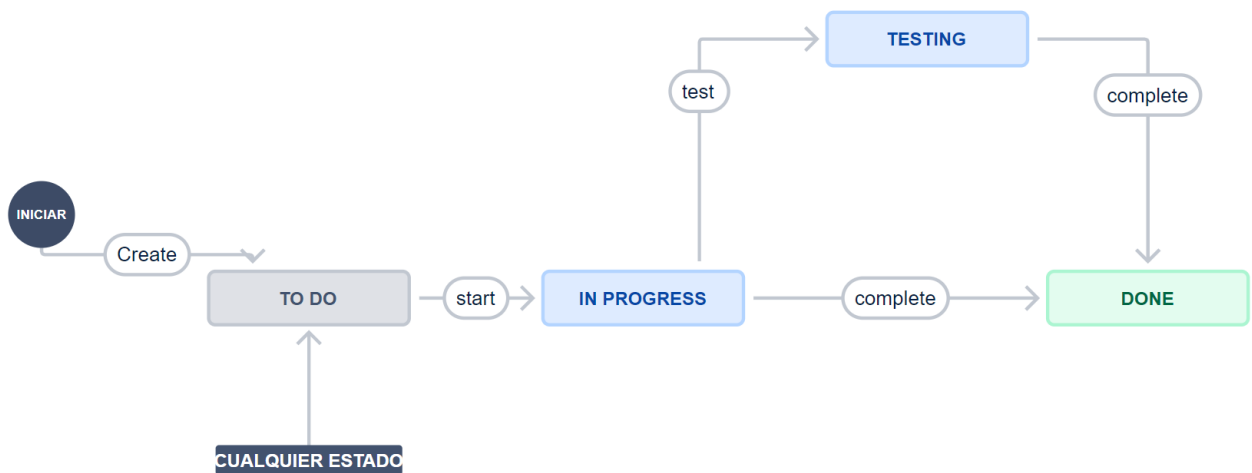


Figura 15. Flujo de trabajo

Autor: David Constante

El flujo de trabajo definido en la Figura 15, cumple con los parámetros establecidos anteriormente sobre la metodología de desarrollo de software híbrida. Las tareas en “TO DO” son las que se encuentran dentro del Backlog, una vez que se hagan agrupado dentro de un sprint estas pasaran al estado de “IN PROGRESS”, en caso de que sea necesario la tarea pasara al estado de “TESTING” donde se indica que se están realizando los test automatizados, una vez completados los test en caso de requerirse la tarea pasa a un estado de “DONE”, lo cual libera el tablero para añadir más Sprints y repetir el ciclo.

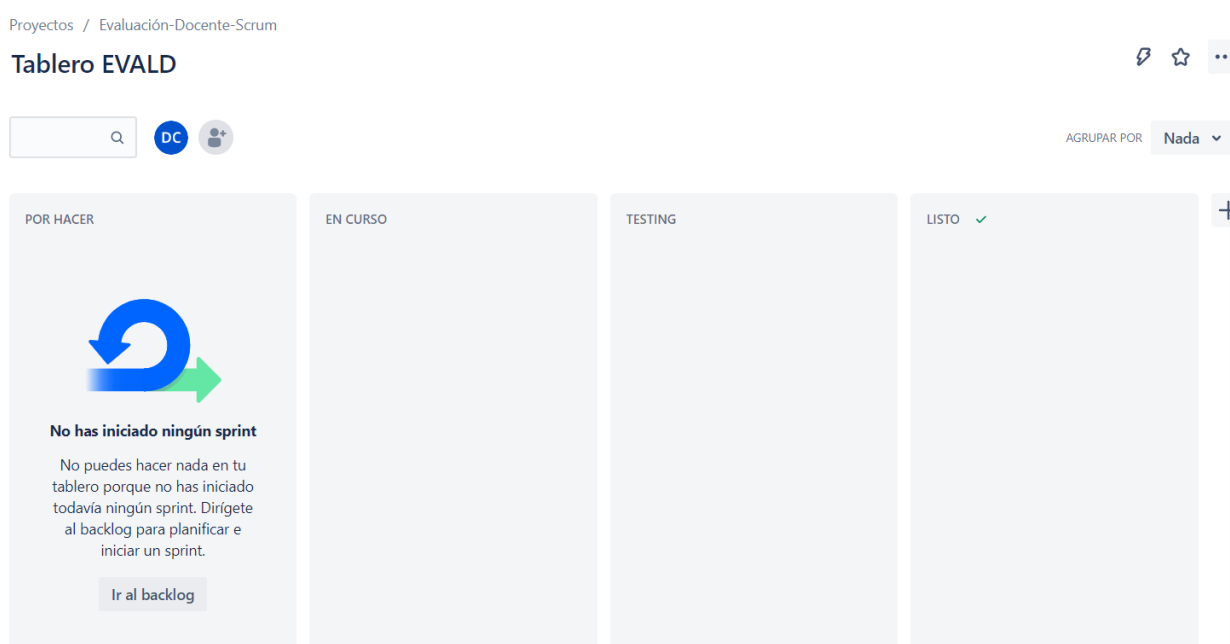


Figura 16. Tablero metodología híbrida

Autor: David Constante

Definición de la Arquitectura de Software

Una arquitectura de software es una representación de alto nivel de las definiciones de un sistema, en donde se evidencian las relaciones entre los elementos y sub-elementos con cierto nivel de granularidad [26]. Una arquitectura se puede representar mediante un diagrama que

permite abstraer el comportamiento de la aplicación con sus respectivos elementos. Esta práctica permite tener una vista panorámica de toda la solución planteada, para que en caso de que fuese necesario se puedan implementar las diferentes mejoras a nivel de escalabilidad si así es requerido.

Arquitectura de la Plataforma

Contenedores

Un contenedor se puede definir según Azure [27], como un paquete de software estándar que agrupa el código de una aplicación con las bibliotecas y los archivos de configuración asociados, además de las dependencias necesarias para que la aplicación de ejecute con normalidad. Esto permite implementar aplicaciones sin que el entorno de ejecución sea un problema.

La principal diferencia entre un contenedor y una máquina virtual (VM) es que: los contenedores se suelen medir en MB debido a que varios contenedores pueden interactuar directamente con la máquina host, mientras que las VM se miden en GB debido a que cada máquina virtual está controlada por un hypervisor, que es una capa adicional de software que consume recursos y es necesaria para que pueda gestionar cada sistema operativo que se encuentre dentro de cada VM [28].

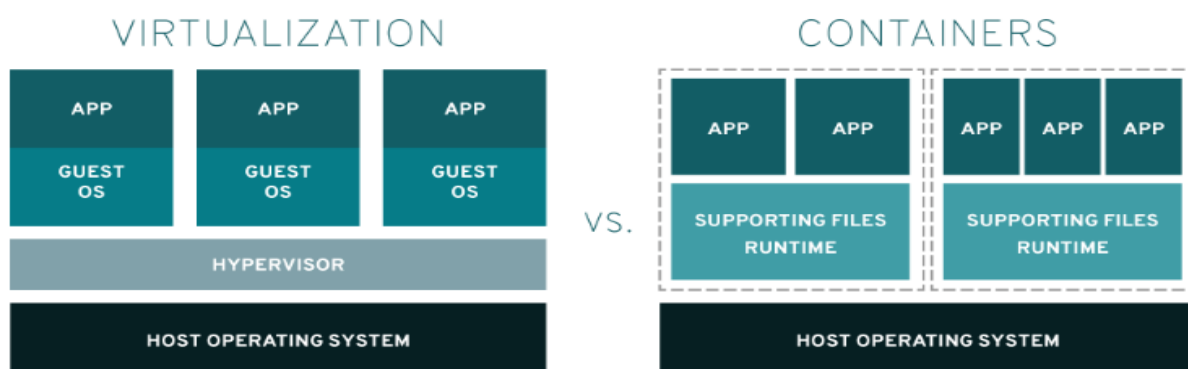


Figura 17. Diferencias entre VM y Contenedor.

Autor: Red Hat [28]

Los contenedores son ideales para aplicaciones nativas de la nube, aplicaciones con microservicios, DevOps y trasladar proyectos de TI escalables [28]. Esto permite que la aplicación sea escalable y pueda coexistir en el mismo servidor con elementos que pueden ser de utilidad y presenten una configuración diferente, y con sistemas y servicios que sean totalmente agnósticos a la plataforma, lo que evita el desperdicio de recursos.

Una vez definidas las características se decidió usar Docker como sistema de contenedores ya que este es libre y se puede usar sin depender de una suscripción u otro servicio de pago.

Base de Datos

Las bases de datos se pueden dividir en dos grandes grupos principalmente, SQL (Structured Query Language) y NoSQL (Not Only Structured Query Languages). En las siguientes comparaciones se muestran las principales diferencias entre estos dos tipos de bases de datos.

Tabla 6. Sql vs NoSql

Autor: Wang y Yang [29]

	SQL	NoSQL
Modelo	Relacional. Se almacenan datos en una Tabla.	No-relacional. Los datos se almacenan en objetos JSON, pares de clave-valor o grafos.
Modelo de datos	Datos Estructurados (Relaciones).	Datos semi estructurados (Colecciones).
Esquema	Fijo.	Dinámico o Flexible.
Consistencia y Disponibilidad	Alta consistencia, se prioriza sobre el rendimiento.	La consistencia, disponibilidad y rendimiento se adaptan a las necesidades de la aplicación.

Rendimiento	Depende del rendimiento del disco duro.	El rendimiento se puede maximizar reduciendo la consistencia.
Escalabilidad	Vertical (Características del Servidor).	Horizontal (Numero de Nodos de la BD).

Ningún sistema de base de datos es mejor que otro, pero un sistema de base de datos se puede comportar de mejor manera que otro dadas ciertas circunstancias, además de que dependerá de las características del sistema. Debido a que la población objetivo de la Universidad ronda los 18.000 estudiantes, se espera un alto tráfico de información, dentro de este tráfico de información la lectura de datos es uno de los factores más importantes, ya que los estudiantes podrán evaluar y comentar el desempeño de los docentes un número limitado de veces, pero los mismos usuarios y el público en general podrán leer los mismos datos un número superior de veces de los que son escritos. Lo que supone que las operaciones de lectura son las que deben tener más importancia.

En este caso, como se puede observar en la figura 18; el rendimiento en las operaciones de lectura coloca a BerkeleyDB como el sistema más rápido, a MongoDB en segundo lugar y MySql como tercero. Según la Developer Survey 2022 [16] BerkeleyDB no figura como uno de los lenguajes conocidos entre los desarrolladoras, y dado que la plataforma en general debe ser mantenida por varios años esta debe usar una tecnología que presente una comunidad contundente, documentación detallada y actualizaciones continuas. Por lo que se ha seleccionado MongoDB como motor de base de datos, ya que este presenta un manejo adecuado de las operaciones de lectura y además figura como el primer motor de bases de datos que soporten NoSQL en el Developer Survey 2022.

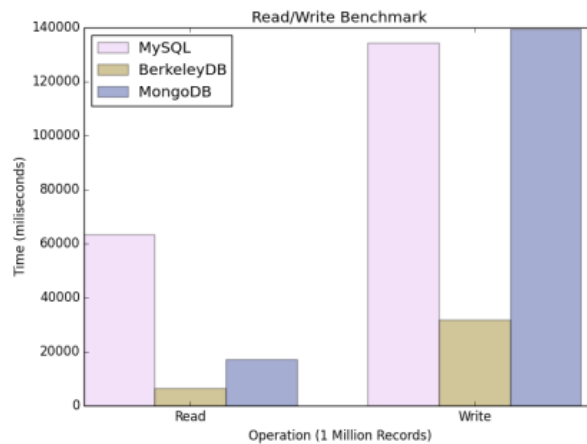


Figura 18. Operaciones de Lectura y Escritura (menos es mejor)

Autor: Wang y Yang [29]

Arquitectura del servidor de producción

La plataforma está alojada en un servidor privado virtual con el sistema operativo Ubuntu en su versión 22.1 con arquitectura x64, está de cara a internet con una dirección ip publica que permite a los usuarios acceder a la plataforma desde cualquier sitio. El VPS tiene tres contenedores: Contenedor 1, MongoDB; Contenedor 2, Backend API; Contenedor 3, Frontend. Estos contenedores estarán conectados con una red interna generada automáticamente por docker lo que permite tener 3 contenedores agnósticos entre sí, pero que están interconectados entre ellos.

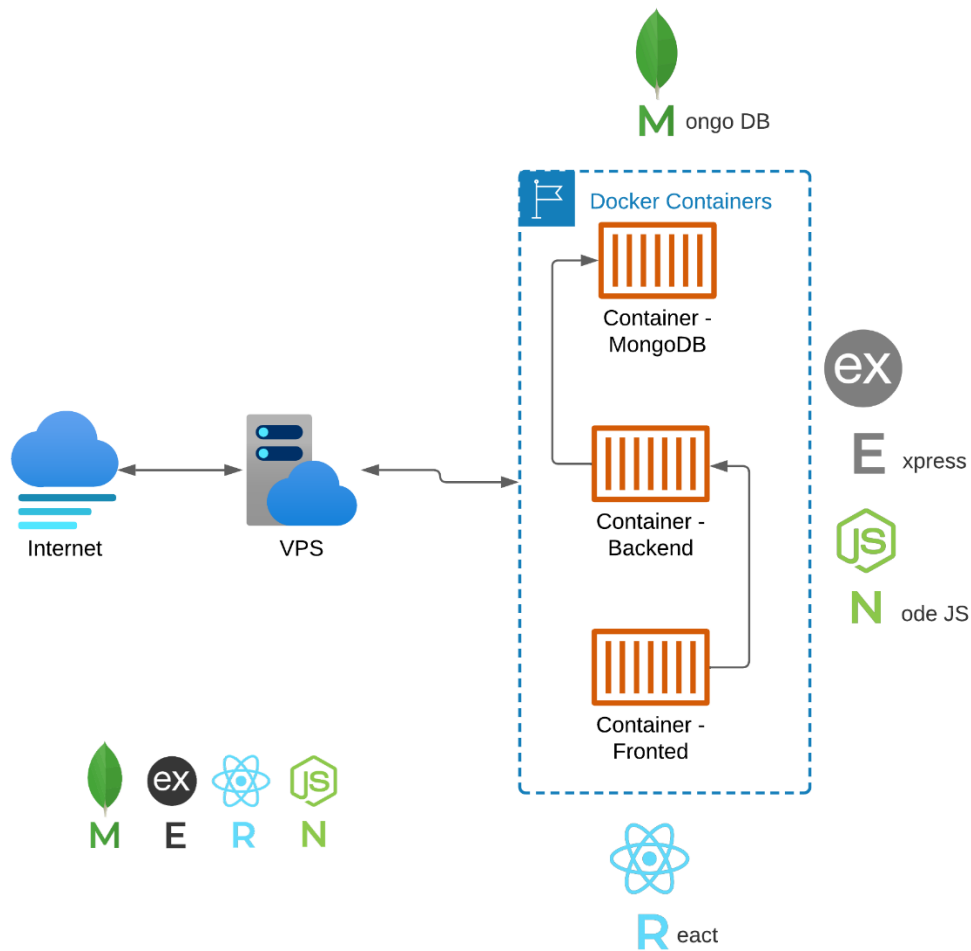


Figura 19. Arquitectura de la plataforma

Autor: David Constante

Docker

Existen diferentes módulos que componen a Docker y que permiten interactuar con la plataforma, los mismos permiten crear diferentes configuraciones según las necesidades de la aplicación, las mismas pueden utilizar imágenes base o contenedores ya configurados. Los módulos de Docker utilizados en el proyecto son los siguientes:

- **Demonio Docker**

Funciona como cualquier demonio dentro del concepto de sistemas operativos, dockerd interactúa con la API de Docker para poder administrar los objetos de la plataforma.

- **Ciente Docker**

Es la forma en la que se puede interactuar con Docker mediante líneas de comando en la terminal.

- **Docker Desktop**

Es otro cliente para Docker con una interfaz gráfica que se puede instalar en Windows, Mac y Linux. Para el desarrollo se utilizó la versión de Windows, pero en el servidor de producción solo se instaló el demonio de Docker.

- **Registros de Docker**

Un registro es un almacén de imágenes de Docker, Docker Hub es un registro público el cual se utilizó en la construcción de los contenedores necesarios para la aplicación. Docker Desktop también permite manejar el registro privado de imágenes de una manera más interactiva, esta característica se utilizó para construir contenedores e imágenes personalizadas.

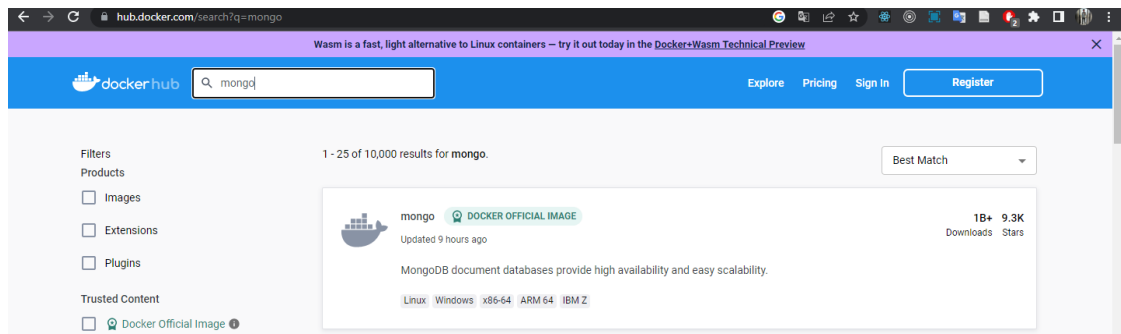


Figura 20. Docker Hub

Autor: David Constante

- **Imágenes**

Es una plantilla de lectura con instrucciones para crear un contenedor Docker, las imágenes pueden estar basadas en otras imágenes. Como se puede observar en la figura 19, existen imágenes oficiales de diferentes plataformas o servicios, que pueden ser configurados previamente antes de crear el contenedor.

- **Contenedores**

Un contenedor es la instancia ejecutable de una imagen, se puede crear, iniciar, detener, reiniciar, mover o eliminar. Un contenedor puede tener una o varias redes, o volúmenes de datos enlazados, los contenedores están aislados de los demás. Esta es la característica primordial utilizada dentro del desarrollo de la Aplicación, debido a que permite configurar diferentes entornos iniciales, como servidores independientes y estos tienen paquetes en diferentes versiones que no intervienen con los demás contenedores.

- **Docker Compose**

Es una herramienta de Docker que utiliza un archivo con extensión yml para construir varios contenedores con instrucciones específicas, esta característica se usó para agilizar el proceso de despliegue de la Aplicación.

MongoDB

Mongo puede ser instalado como un servicio sencillo, pero dado que la Aplicación necesita usar transaccionalidad, es necesario crear un clúster de MongoDB, que también son conocidos como replica sets. La replicación añade una capa de seguridad a los datos a manera de redundancia, en caso de que una base de datos falle, los datos están respaldados en otro servidor.

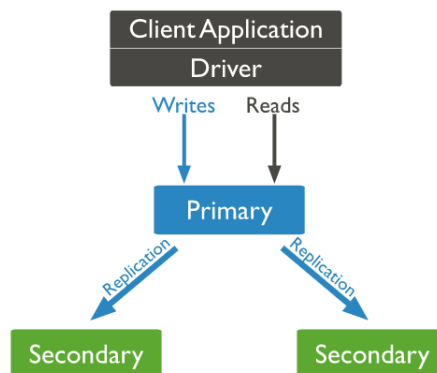


Figura 21. Diagrama replica set.

Autor: MongoDB [30]

Arquitectura del Backend

El backend de la Aplicación se desarrolló en base al principio de single responsibility, en donde cada módulo se encargará de una tarea específica. Las peticiones son gestionadas por un servidor general desarrollado en express, en el cual un router se encarga de gestionar las peticiones e interactuar con el módulo del controlador que le corresponda a cada entidad, para que así las interacciones con la base de datos sigan una línea de proceso fácil de manejar y escalar. También se utilizaron middlewares para gestionar las respuestas generadas en caso de éxito o fallo.

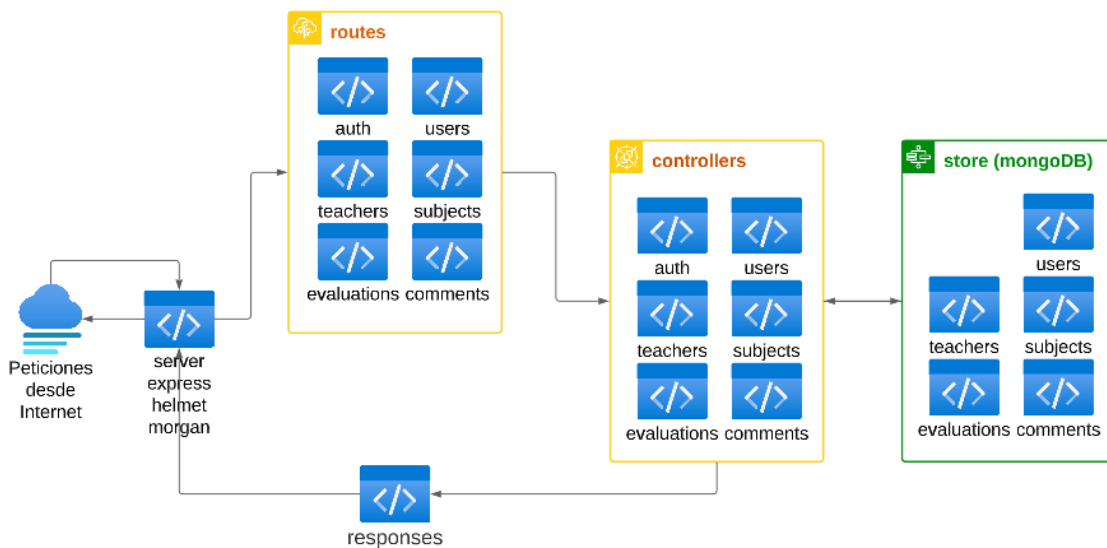


Figura 22. Arquitectura backend

Autor: David Constante

Debido a que la Aplicación está desarrollada sobre NodeJs, se utilizaron varios paquetes gestionados mediante npm, tales como:

- **Mongoose**

Es un ODM (Object Document Mapper) que permite interactuar con la base de datos de una manera asíncrona y sencilla sin necesidad de escribir queries. Está basado en Modelos de documentos, promesas y callbacks [31].

- **Morgan**

Es un middleware para NodeJs que sirve para monitorear las peticiones que llegan al servidor, y conocer sus estados. Se puede monitorear dentro del proceso de la terminal de comandos o como un archivo de log [32].

- **Helmet**

Es una dependencia que ayuda a asegurar el servidor Express añadiendo configuraciones para Https [33].

- **Boom**

Específicamente conocido como @hapi/Boom, boom hace parte de una serie de herramientas que forman el ecosistema de hapi web framework. Boom es utilizado como una herramienta aislada del framework de hapi que sirve para gestionar las respuestas de errores de una manera más sencilla, y así generar respuestas más simples de leer en la API [34] .

- **Bcrypt**

Es una dependencia que es utilizada para encriptar las contraseñas generadas por los usuarios, y así estas no sean almacenadas como texto plano dentro de la base de datos. Internamente implementa funciones de hash y una frase semilla que son utilizadas para encriptar la contraseña, y aun que esta sea generada varias veces con la misma entrada, el resultado siempre es diferente [35].

- **Jwt-node**

Es un paquete que permite implementar JWT (Json Web Token), estos tokens son utilizados para gestionar la identidad dentro de la aplicación web [36].

- **Joi**

Es un sistema de validación de esquemas de objeto, se utilizó para generar modelos y así validar las peticiones entrantes en el servidor, y así evitar la carga innecesaria del mismo [37].

- **Dot-env**

Es un paquete que permite crear variables de entorno dentro de la aplicación, para así añadir una capa de seguridad en caso de que la aplicación sea desplegada en otro sitio sin autorización [38].

Integración del stack Mern

El stack MERN es una pila de tecnología de software que consta de MongoDB, Express, React y Node.js. Es una pila popular para el desarrollo de aplicaciones web modernas, especialmente para aplicaciones a gran escala. Cada uno de los componentes de la pila MERN ofrece un conjunto único de características que se combinan para proporcionar una solución completa y escalable para el desarrollo de la plataforma.

MongoDB es una base de datos NoSQL que utiliza documentos para almacenar datos en lugar de tablas como en las bases de datos relacionales. MongoDB es escalable y flexible y se integra bien con Node.js y Express. La integración de MongoDB en el stack MERN permitió crear plataforma flexible que pueden manejar grandes cantidades de datos.

Express es un marco de aplicaciones web para Node.js. que proporciona un conjunto de herramientas para el desarrollo web escalable y robusto. Express simplifica la creación de aplicaciones web a nivel de backend al proporcionar una capa de abstracción sobre Node.js. La integración de Express en el stack MERN permitió crear una API backend flexible y escalable.

React es una biblioteca de JavaScript para la creación de interfaces de usuario (UI). React posee una variedad de herramientas propias que hacen posible manejar el estado de la plataforma de una manera reactiva, esto permite que la plataforma pueda actualizarse en tiempo real sin necesidad de actualizar la página.

Node.js es un entorno de tiempo de ejecución de JavaScript construido sobre el motor V8 de Google Chrome. Node.js, esto permitió unificar el lenguaje JavaScript para el desarrollo del

Frontend y backend de la aplicación y que exista una base sólida de código, la misma que puede ser escalada fácilmente debido a que se trata de una sola tecnología.

En resumen, la integración del stack MERN en el desarrollo de la plataforma proporcionó una solución completa y escalable para la creación de un sistema completo desde cero. Cada uno de los componentes de la pila MERN ofrece un conjunto único de características y librerías externas que se combinan para proporcionar una solución completa.

Software Utilizado

Diseño

El diseño es parte fundamental en el desarrollo del proyecto, por lo que usar un software que ayude a definir los estilos, colores y diseños de toda la interfaz de la aplicación. Existen muchas herramientas en el mercado, entre las cuales las más usadas en el mercado se pueden diferenciar en la siguiente tabla comparativa.

Tabla 7. Comparación software para mockup

Autor: David Constante

Característica	Adobe XD	Figma	Sketch
Precio	\$ 9.99/mes	Gratis o \$15/mes	\$ 9.99/mes
Plataforma	Escritorio	Web y escritorio	Web (visualizar) y Escritorio
Instalación	Requiere paquete de programas,	Instalación única.	Instalación única.

Sistema operativo	Windows, Mac OS, Android, IOS.	Windows, Mac OS, Android, IOS.	Mac OS.
Requisitos	Medios.	Bajos.	Bajos.

Todos los softwares tienen las características necesarias para esquematizar pantallas, botones o elementos, para el proyecto se utilizó Figma debido a que es el único que tiene versión gratuita con todas sus características más importantes.

Editor de Código

La naturaleza de la plataforma es web, eso quiere decir que HTML CSS y JavaScript son los elementos que en su mayoría le darán vida a la aplicación. Nativamente se puede codificar incluso en un bloc de notas, pero esto añadiría al proyecto una capa de complejidad que no está justificada, el desarrollo de software utiliza IDEs o Editores de código, los IDEs están enfocados a un lenguaje y proyecto, pero tienen la limitante de que consumen muchos recursos y en su mayoría permiten codificar únicamente en un solo lenguaje de programación. Por otro lado, los editores de código se pueden definir como un bloc de notas mejorado, a este se le pueden añadir o quitar herramientas según las necesidades, una de las características principales en un editor de código es la comunidad, ya que esta es la que se encarga de reportar errores o desarrollar extensiones que se integran nativamente al editor y agilizan en gran medida el desarrollo.

Aun así existen varios editores de código en el mercado, como: Sublime Text, Atom, Brackets o Visual Studio Code, siendo el último el más popular por la comunidad con un 74.48% de aceptación según la Developer Survey 2022 [16]. Este editor de código es propietario de Microsoft desarrollado para Windows, MacOS o Linux e incluso funciona en plataformas web. Además de que tiene una biblioteca de extensiones muy extensa, lo que brinda opciones para configurar el editor según las necesidades del proyecto. Por estas razones este Editor de código es el escogido para desarrollar la Aplicación.

Gestor de Peticiones

Una de las fases críticas en el desarrollo de la Aplicación es la gestión de las peticiones para testear los endpoints durante el desarrollo. Esto permite probar la API sin la necesidad de desarrollar el frontend y manejar la información de las peticiones como los headers para gestionar los Json web tokens, además de verificar de manera las simplificada las respuestas que genera la API. En la siguiente tabla comparativa se destacan las características de los gestores de peticiones más usados.

Tabla 8. Comparación gestores de peticiones

Autor: David Constante

Característica	Postman	Insomnia	Thunder Client
Precio	Gratis o \$12/mes.	Gratis o \$50/mes.	Gratis.
Tipo de Software	Aplicación Escritorio.	Aplicación Escritorio.	Extensión de Visual Studio Code.
Curva de aprendizaje	Alta.	Media.	Media.
UI – UX	Completa.	Simple.	Simple.
Opciones de configuración	Extensas.	Básicas.	Básicas.
Sincronización	Nube.	Nube.	No posee.

Gestor de Base de datos

Las bases de datos no relacionales también poseen sistemas SGBD, en este caso MongoDB posee diferentes paquetes que permiten monitorear, administrar y manejar las diferentes bases de datos. En el siguiente cuadro comparativo se muestran los sistemas gestores de bases de datos para MongoDB más utilizados.

Tabla 9. Comparacion SGBDS para MongoDB

Autor: David Constante

Característica	Mongo-express	MongoDB Compass	Robomongo 3T
Precio	Gratis.	Gratis.	Gratis.
Propietario	Open Source.	MongoDB.	3T.
Tipo de Software	Imagen de Docker - servidor web.	Aplicación.	Aplicación.
Shell	No integrada.	Integrada.	Integrada.
UI - UX	Simple.	Completa.	Anticuada.
Plataformas	Multiplataforma.	Windows, Linux, MacOS.	Windows, Linux, MacOS.

Todos los paquetes de software cumplen con su cometido, mongo-express al funcionar como una imagen de Docker añade una carga innecesaria a Docker, por otro lado, Robomongo y MongoDB Compass Tienen características similares, pero se decidió usar MongoDB Compass debido a que tiene una experiencia de usuario superior y es el software recomendado por MongoDB.

Redux Toolkit

Es una librería gratis que se puede implementar cualquier capa de UI o Framework, como Angular, Vue y como es el caso ReactJs. React Redux es la rama específica del proyecto que permite integrar los elementos y hooks de React con un completo manejador de estado, que se utilizó para almacenar de manera temporal como el usuario interactúa con el sistema web, antes de generar una petición a la API, reduciendo así la carga de trabajo y acelerando la respuesta de la Aplicación [39].

Tailwind CSS

Dado que el Framework tiene herramientas para manejar los media queries de manera mas práctica, todo el desarrollo del frontend tiene un enfoque de mobile first, es decir este es diseño principal y a partir del cual se gestionan los diferentes breakpoints para las demás dimensiones de pantalla. Y se toman las resoluciones más utilizadas por Google.

Tabla 10. Breakpoints según el tamaño de pantalla.

Autor: David Constante

Dispositivo	Abreviación	Resolución
Smartphone	sm	425 px
Tablet	md	768 px
Laptop	lg	1024 px
Computadora de Escritorio	xl	1440 px

3.2. Desarrollo de la propuesta

Planificación

Para gestionar el desarrollo de la aplicación se utilizaron cuatro categorías para la importancia de las tareas, que son: más baja, baja, media, alta y más alta. Dentro de Jira tienen la siguiente leyenda:

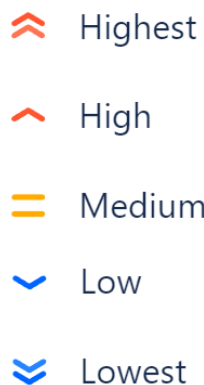


Figura 23. Leyenda de prioridad de tareas.

Autor: David Constante

Para el desarrollo de la Aplicación se definieron seis Sprints con sus respectivas fechas límite que están contenidos en cinco Epics. En donde cada Sprint tiene un Epic, a excepción del de Api que contiene Sprints para su desarrollo y para Testing.

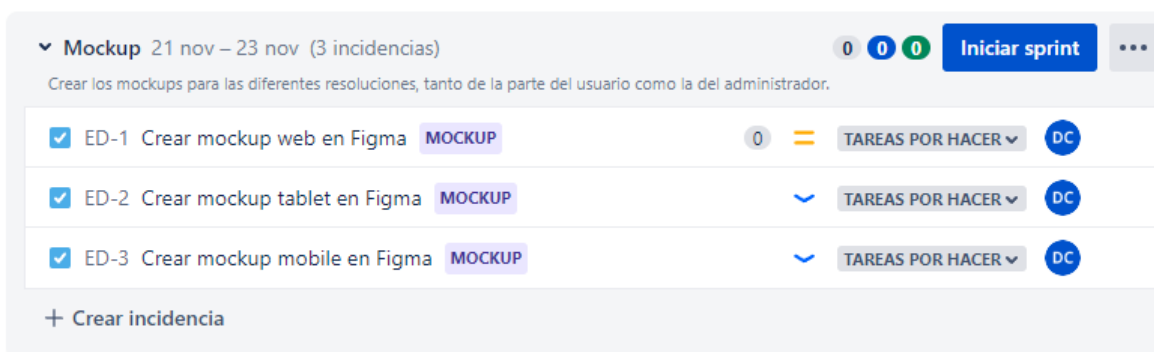


Figura 24. Sprint 1: Diseño del Mockup

Autor: David Constante

▼ **Server inicial** 26 nov – 29 nov (4 incidencias) 0 0 0 Iniciar sprint ...
 Instalar dependencias necesarias y configurar el servidor inicial para la API.

<input checked="" type="checkbox"/>	ED-4 Inicializar proyecto SERVER	▼	TAREAS POR HACER ▼	DC
<input checked="" type="checkbox"/>	ED-5 Configurar arquitectura backend SERVER	≡	TAREAS POR HACER ▼	DC
<input checked="" type="checkbox"/>	ED-6 Configurar contenedor MongoDB SERVER	≡	TAREAS POR HACER ▼	DC
<input checked="" type="checkbox"/>	ED-7 Config server SERVER	▼	TAREAS POR HACER ▼	DC

+ Crear incidencia

Figura 25. Sprint 2: Inicialización del servidor

Autor: David Constante

▼ **API** 1 dic – 15 dic (4 incidencias) 0 0 0 Iniciar sprint ...
 Crear los endpoints necesarios para la aplicación.

<input checked="" type="checkbox"/>	ED-8 Crear modelos para mongoose API	- ^	TAREAS POR HACER ▼	DC	...
<input checked="" type="checkbox"/>	ED-9 Crear controladores para cada entidad API	^	TAREAS POR HACER ▼	DC	
<input checked="" type="checkbox"/>	ED-10 Crear rutas para cada entidad API	^	TAREAS POR HACER ▼	DC	
<input checked="" type="checkbox"/>	ED-11 Crear middlewares para las respuestas API	▼	TAREAS POR HACER ▼	DC	

+ Crear incidencia

Figura 26. Sprint 3: Desarrollo Api

Autor: David Constante

▼ **Test API** 16 dic – 23 dic (6 incidencias) 0 0 0 Iniciar sprint ⋮

Test Automaticos

<input type="checkbox"/> ED-12 Test auth y login API	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-13 Test CRUD users API	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-14 Test CRUD subjects API	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-15 Test CRUD teachers API	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-16 Test CRUD evaluations API	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-17 Test CRUD comments API	⬆	TAREAS POR HACER ▾	DC

+ Crear incidencia

Figura 27. Sprint 4: Test de la Api

Autor: David Constante

▼ **Frontend** 24 dic – 14 ene (7 incidencias) 0 0 0 Iniciar sprint ⋮

Maquetar todas las pantallas y conectar con la API.

<input checked="" type="checkbox"/> ED-18 Maquetar paginas usuarios FONTEND	=	TAREAS POR HACER ▾	DC
<input checked="" type="checkbox"/> ED-19 Maquetar paginas administrador FONTEND	=	TAREAS POR HACER ▾	DC
<input checked="" type="checkbox"/> ED-20 Conectar paginas administrador con la API FONTEND	=	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-21 Probar CRUD desde la pantalla admin FONTEND	⬆	TAREAS POR HACER ▾	DC
<input checked="" type="checkbox"/> ED-22 Conectar paginas de usuario con la API FONTEND	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-23 Probar CRUD desde las pantallas de usuarios FONTEND	⬆	TAREAS POR HACER ▾	DC
<input type="checkbox"/> ED-24 Test de rendimiento FONTEND	=	TAREAS POR HACER ▾	DC

+ Crear incidencia

Figura 28. Sprint 5: Desarrollo del Frontend

Autor: David Constante

▼ **Despliegue** 15 ene – 18 ene (3 incidencias) 0 0 0 Iniciar sprint ...

Subir el proyecto a Producción.

<input checked="" type="checkbox"/>	ED-25 Empaquetar aplicación	DESPLIEGUE	=	TAREAS POR HACER ▼	DC
<input checked="" type="checkbox"/>	ED-26 Desplegar aplicación	DESPLIEGUE	=	TAREAS POR HACER ▼	DC
<input checked="" type="checkbox"/>	ED-27 Configurar HTTPS	DESPLIEGUE	=	TAREAS POR HACER ▼	DC

+ Crear incidencia

Figura 29. Sprint 6: Despliegue de la plataforma

Autor: David Constante

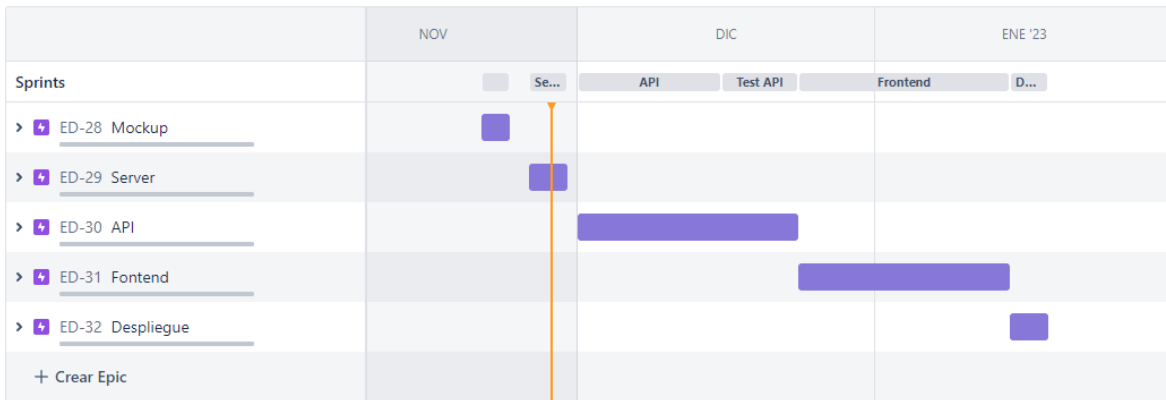


Figura 30. Hoja de Ruta

Autor: David Constante

Diseño

Epic 1 - Mockup

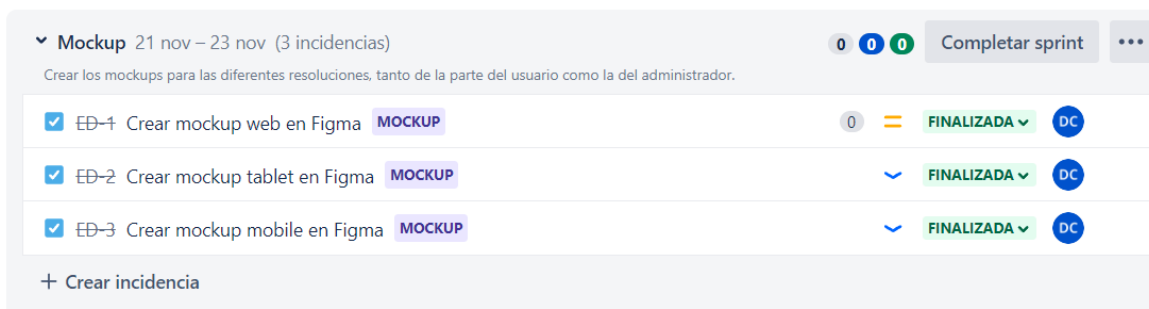


Figura 31. Epic 1, Sprint 1 – Diseño del Mockup

Autor: David Constante

La plataforma de evaluación docente pretende recolectar información por parte de los estudiantes para que esta sea más real a las circunstancias en las cuales se desenvuelve el periodo académico. Además, permitirá crear una comunidad universitaria en la que los estudiantes puedan expresarse de manera objetiva sin temor a las posibles represalias que se puedan generar en su contra. Según el reglamento de evaluación del desempeño del personal académico y apoyo académico de la universidad técnica de Ambato en su artículo 10 titulado Garantías de la evaluación integral del desempeño, señala que: “Para la realización del proceso de evaluación integral del desempeño del personal académico, la Universidad Técnica de Ambato garantizará la difusión de los propósitos y procedimientos, y la claridad, rigor y transparencia en el diseño e implementación del mismo, como el debido proceso, tutela efectiva y seguridad jurídica” [4]

Es por lo que la ruta principal esta visible para cualquier persona que requiera ver la información, y solo si tiene la facultad de poder interactuar con la página, el usuario puede autenticarse en la misma. Para acceder al login del sistema se muestra al usuario una fotografía con la temática de la Universidad Técnica de Ambato o alguna idea relacionada a la educación. Esta imagen será suministrada consecutivamente por cualquier integrante de la

comunidad universitaria y tiene como objetivo hacer que la misma genere interacciones de manera orgánica con la plataforma y así tener una experiencia de usuario más llamativa para todos los usuarios. Ya que uno de los objetivos de la plataforma es que los usuarios interactúen de manera voluntaria.

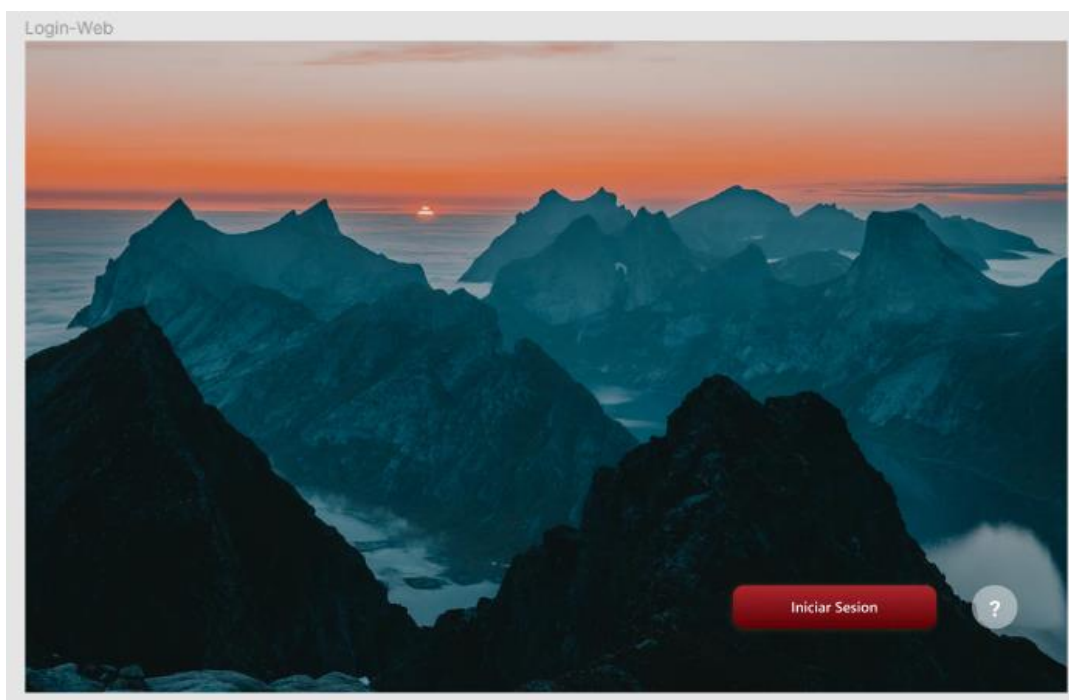


Figura 32. Login de Usuario (1/2)

Autor: David Constante

El usuario puede acceder al formulario de registro interactuando con el botón de iniciar sesión, el cual despliega una ventana pop-up que permite al usuario colocar sus credenciales de acceso o ingresar a su vez con su cuenta de Microsoft, que es el mecanismo utilizado para autenticarse en otros servicios de la Universidad y así seguir una línea de diseño más concisa.

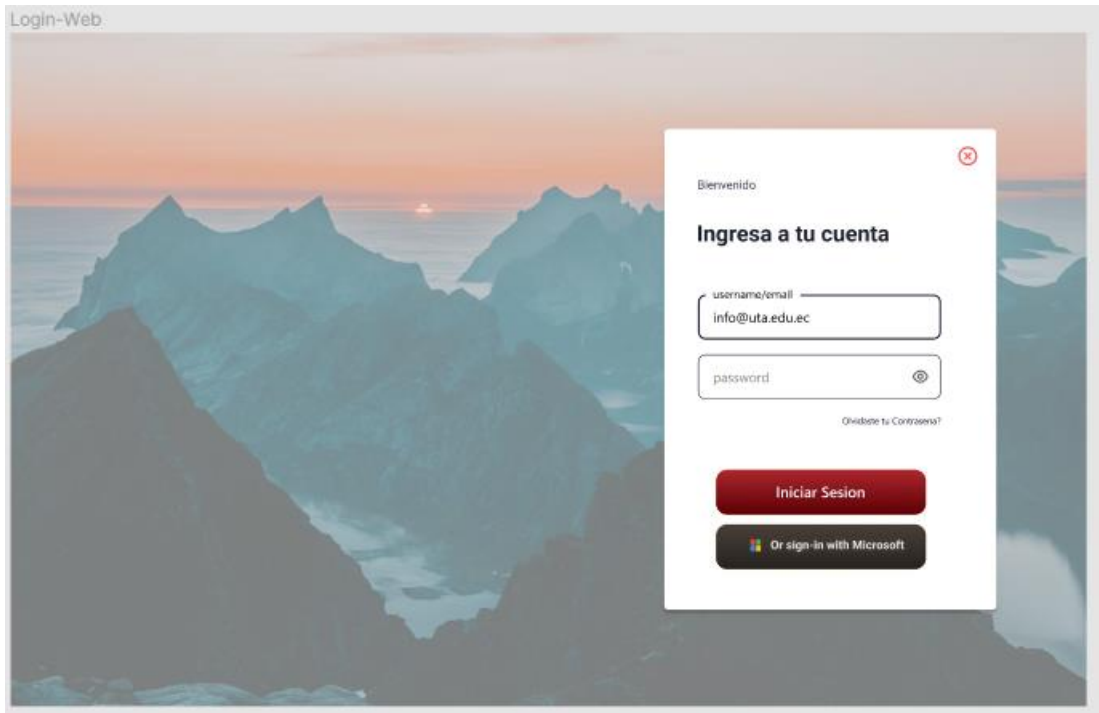


Figura 33. Login de usuario (2/2)

Autor: David Constante

Una vez las credenciales del usuario sean verificadas en el backend de la aplicación se le redirige al usuario hacia la pantalla correspondiente. Existen cuatro tipos de usuario con sus respectivos permisos que se pueden observar en la siguiente tabla.

Tabla 11. Tipos de usuarios y permisos

Autor: David Constante

Nombre	Permisos
Invitado	<ul style="list-style-type: none"> • Ver las estadísticas de los docentes.
Estudiante	<ul style="list-style-type: none"> • Ver las estadísticas de los docentes. • Realizar evaluaciones a los docentes. • Realizar comentarios de los docentes.

	<ul style="list-style-type: none"> • Reaccionar a comentarios. • Eliminar comentarios que le pertenecen.
Moderador	<ul style="list-style-type: none"> • Ocultar comentarios.
Coordinador	<ul style="list-style-type: none"> • Ver estadísticas de la Facultad Y Carrera.
Administrador	<ul style="list-style-type: none"> • Crear usuarios, materias, docentes, estudiantes.

El “hero” de la aplicación contiene un diagrama de araña que permite a los usuarios conocer las estadísticas generales de los docentes, el grafico permite tener una perspectiva más amplia de las características generales que presentan flaquezas y así tomar las medidas respectivas, como capacitaciones a los docentes en las áreas que más lo requieran. También existe un listado que permite conocer los docentes mejor puntuados, que pueden ser tomados como referente para difundir sus metodologías de enseñanza y como lo estipula el reglamento de la Universidad, reciban el reconocimiento respectivo. Además de que existe también un elemento de búsqueda que permite a los usuarios conocer las estadísticas de un docente o en su defecto realizar la evaluación respectiva.



Figura 34. Mockup Home

Autor: David Constante

Dentro de la pestaña de cada docente se puede visualizar otro grafico de araña que permite conocer las estadísticas generales del mismo, estas estadísticas se ponderaran según todas las

evaluaciones que realicen los estudiantes. Los estudiantes solo podrán evaluar a los docentes que les impartan clase. Cada estudiante tiene la facultad de realizar 5 evaluaciones y 5 comentarios por semana.

Estas evaluaciones y los comentarios serán anónimos para proteger la identidad de los estudiantes, pero en el caso de los comentarios se genera un nombre aleatorio que permite al administrador identificar al estudiante durante el periodo académico. Los moderadores tendrán la facultad de ocultar los comentarios ofensivos, para consecutivamente darles seguimiento en privado y de ser necesario eliminarlos de la plataforma y se tomen las medidas necesarias para con el estudiante.

Las opiniones pueden ser aceptadas o no por los estudiantes, por lo que los mismos tienen la facultad de estar de acuerdo o en desacuerdo con la opinión de uno de sus compañeros y así verificar si se trata de una opinión real sobre el desempeño del docente.



*Figura 35. Mockup Docente Evaluacion
Autor: David Constante*

La evaluación docente como tal se realiza mediante una escala lineal que va del uno al cinco, en donde uno es lo más bajo y cinco lo más alto. Se utilizan estrellas para interpretar estas escalas y así generar una experiencia de usuario agradable para el usuario.

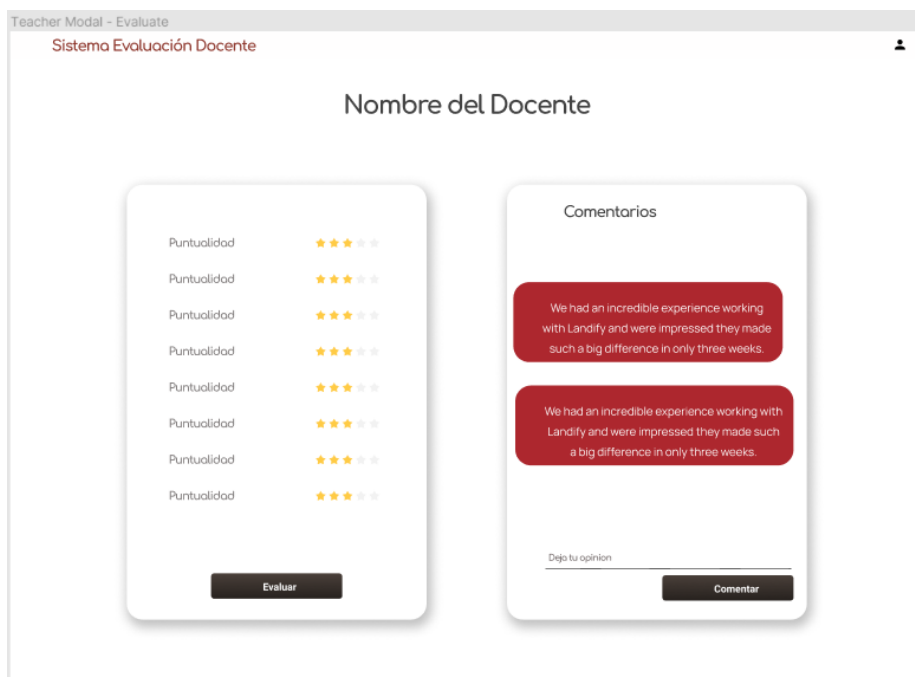


Figura 36. Mockup Docente Estadísticas

Autor: David Constante

Las evaluaciones se realizan bajo seis criterios a considerar los cuales reflejan el desempeño del docente. Para obtener el puntaje referencial de un docente se tomó como referencia el cálculo de la mediana de todas las evaluaciones del periodo actual en sus respectivas características, esto con la finalidad de obtener un valor mas real el cual no se vea afectado por estudiantes que hagan mal uso del sistema al no calificar objetivamente a sus docentes, los diferentes criterios de evaluación se detallan en la tabla a continuación.

Tabla 12. Criterios de evaluación

Autor: David Constante

Rubrica	¿Qué evalúa?
Puntualidad	En este apartado se evalúa la puntualidad del docente, así como el correcto uso de sus horas de clase
Conocimiento	En este apartado se evalúa la capacidad del docente para solventar las dudas de los estudiantes. (Domina la asignatura)
Metodología de enseñanza	En este apartado se evalúa la capacidad del docente para utilizar recursos interactivos en su clase y que de esta manera sus estudiantes sean capaces de dominar la asignatura.
Sílabo	En este apartado se evalúa si el docente cubre en profundidad todos los temas establecidos en el sílabo.
Transparencia con las notas	En este apartado se evalúa si el docente socializa con sus alumnos las notas obtenidas en los diferentes apartados (deberes, pruebas, exámenes). Y si hace un correcto uso de la rúbrica de calificación.
Feedback	En este apartado se evalúa la capacidad del docente para retroalimentar su clase, mediante tutorías o clases de refuerzo.

Si las credenciales del usuario son de administrador este puede visualizar un panel de administración en el cual puede realizar todas las operaciones CRUD sobre las entidades del sistema.

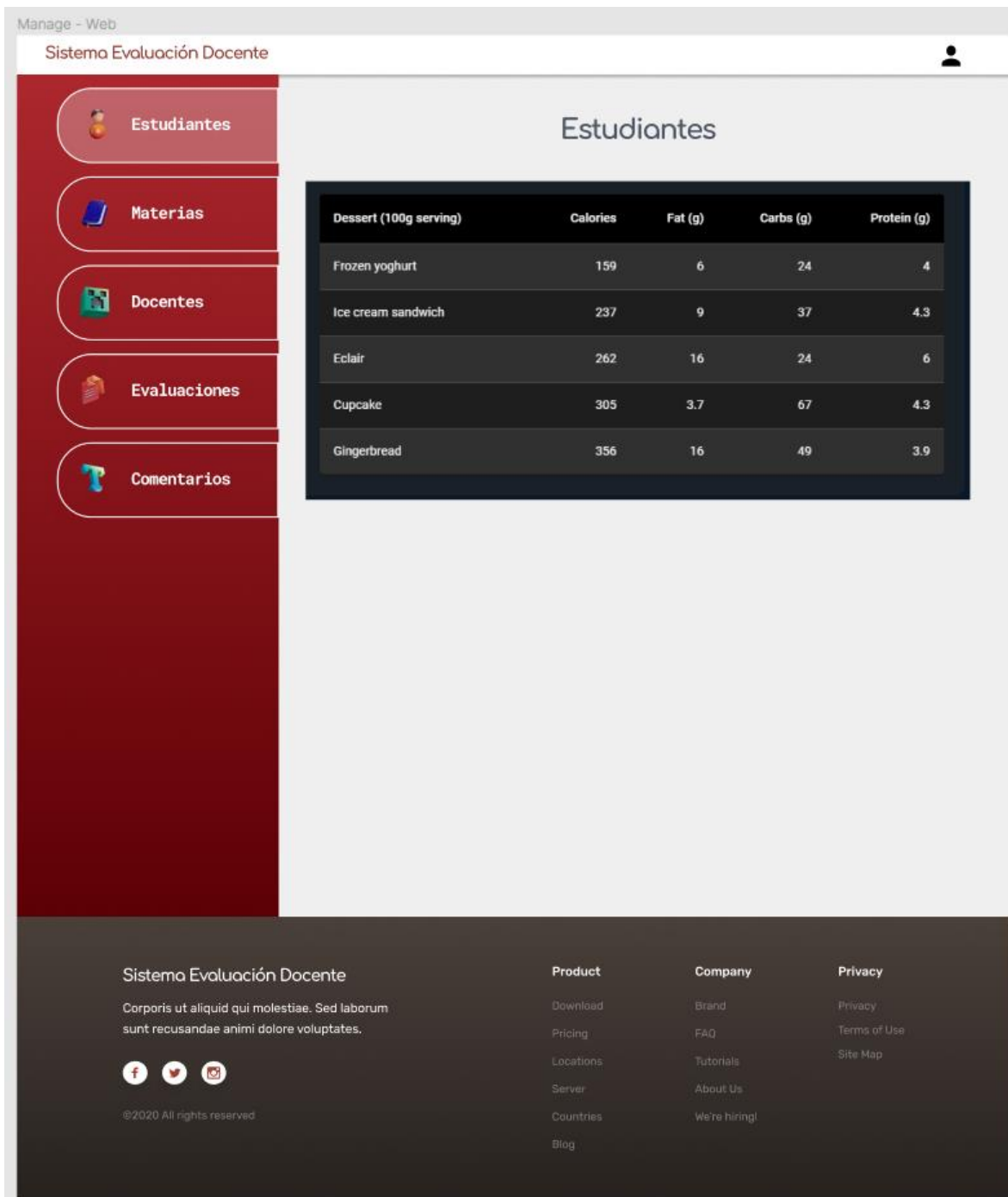


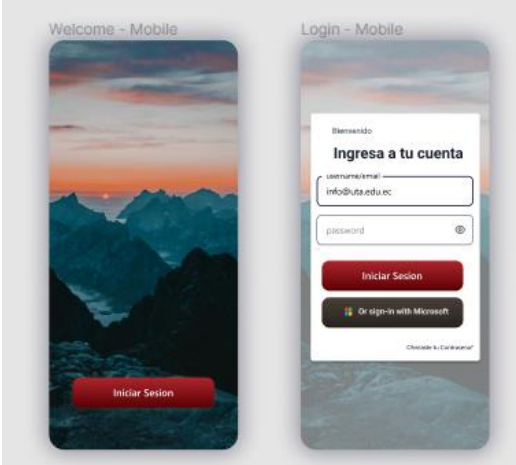
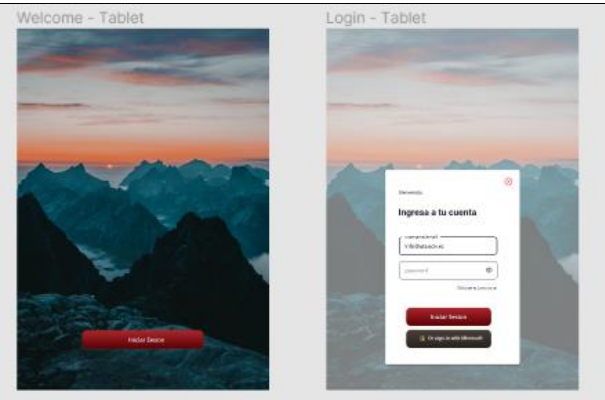
Figura 37. Mockup panel de administración

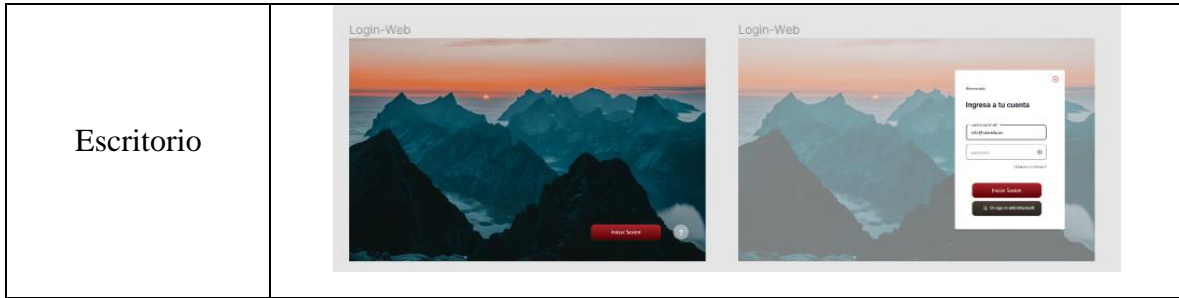
Autor: David Constante

Como se puede evidenciar en la figura 1 los usuarios que utilizan sus dispositivos móviles han aumentado en los últimos años y los mismos tienen una tendencia alcista, por consiguiente, debido a que uno de los objetivos del proyecto es mejorar la experiencia de usuario dentro de la plataforma los diferentes elementos que conforman el UI de la misma poseen características responsive, dichos elementos se adaptan al tamaño de la pantalla del dispositivo desde el cual acceda el usuario. Los mockups completos se pueden observar en el Anexo 3, Anexo 4 y Anexo 5.

Tabla 13. Diseño responsive.

Autor: David Constante

Tipo de pantalla	Mockup
Smartphone	
Tablet	



Codificación

Epic 2 - Server

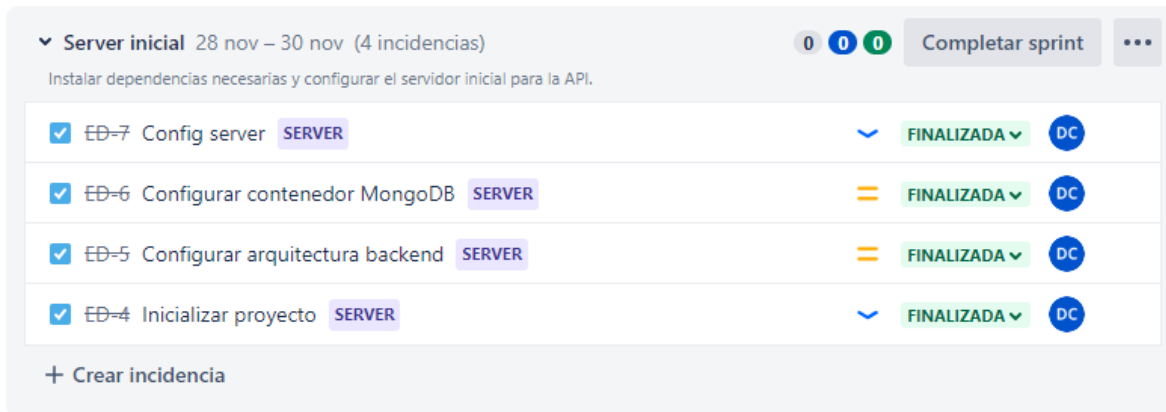


Figura 38. Epic 2, Sprint 2 – Servidor inicial

Autor: David Constante

Inicializar Proyecto

Para inicializar la parte del backend se crea un proyecto con el gestor de paquetes npm mediante el comando `npm init`, después de colocar toda la información necesaria se genera un archivo con extensión “.json”, al que posteriormente se le añaden todas las dependencias necesarias.

Para el control de versiones se utilizó Git con el servicio Github, la configuración inicial se realiza por medio del comando `git init`, una vez configurado el proyecto con las credenciales

del repositorio en la nube se sincronizan los cambios para probar el funcionamiento del servicio.

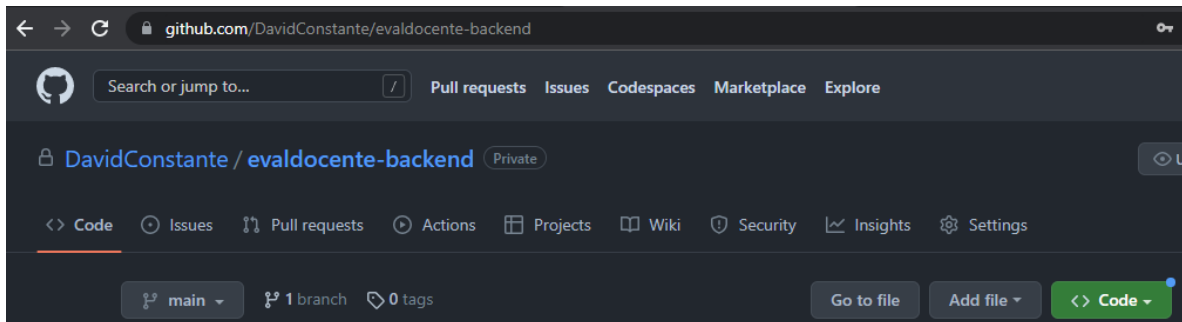


Figura 39. Repositorio en Github

Autor: David Constante

```

1  {
2  "name": "evaldocente-backend",
3  "version": "1.0.0",
4  "description": "Api for the Eval Docente",
5  "main": "index.js",
6  "scripts": {
7    "start": "node index.js",
8    "dev": "nodemon index.js",
9    "test": "echo \\\"Error: no test specified\\\" && exit 1"
10 },
11 "repository": {
12   "type": "git",
13   "url": "git+https://github.com/DavidConstante/evaldocente-backend.git"
14 },
15 "keywords": [
16   "node",
17   "express",
18   "api"
19 ],
20 "author": "David Constante <omileconstante@gmail.com>",
21 "license": "MIT",
22 "bugs": {
23   "url": "https://github.com/DavidConstante/evaldocente-backend/issues"
24 },
25 "homepage": "https://github.com/DavidConstante/evaldocente-backend#readme",
26 "dependencies": {
27   "@hapi/boom": "^10.0.0",
28   "bcryptjs": "^2.4.3",
29   "cors": "^2.8.5",
30   "dotenv": "^16.0.2",
31   "express": "^4.18.1",
32   "express-validator": "^6.14.2",
33   "joi": "^17.6.1",
34   "jsonwebtoken": "^8.5.1",
35   "mongoose": "^6.6.2",
36   "morgan": "^1.10.0"
37 },
38 "devDependencies": {
39   "standard": "^17.0.0"
40 },
41 "eslintConfig": {
42   "extends": "./node_modules/standard/eslintrc.json"
43 }
44 }
45

```

Figura 40. package.json

Autor: David Constante

Configuración del contenedor MongoDB

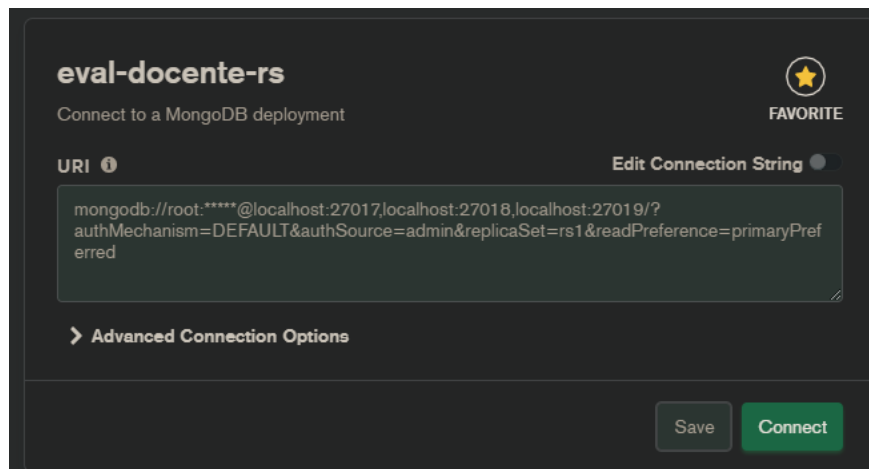
Para la base de datos se utilizó una imagen alojada en DockerHub con el nombre de “ *davidconstante/replicaset* ” esta contiene un cluster de MongoDB con 3 nodos interconectados mediante los puertos: 27017, 27018, 27019, el nodo principal arrancará por defecto en el puerto 27017. Para el almacenamiento se creó un volumen gestionado automáticamente por Docker, al igual que la red con un controlador de tipo bridge. Toda esta configuración está resumida en un archivo de *docker-compose.yml* .

```
1  version: "3.8"
2
3  # The services of your app
4  services:
5    # DB: Mongo replica set
6    db:
7      image: davidconstante/replicaset:1.0.0
8      container_name: mongors
9      hostname: mongo
10     restart: unless-stopped
11     ports:
12       - 27017:27017
13     volumes:
14       - "mongo-rs:/var/mongodb/db"
15     networks:
16       - eval-docente
17     entrypoint: [ "systemctl", "start", "mongors.service" ]
18 volumes:
19   mongo-rs:
20     driver: local
21 networks:
22   eval-docente:
23     driver: bridge
24
```

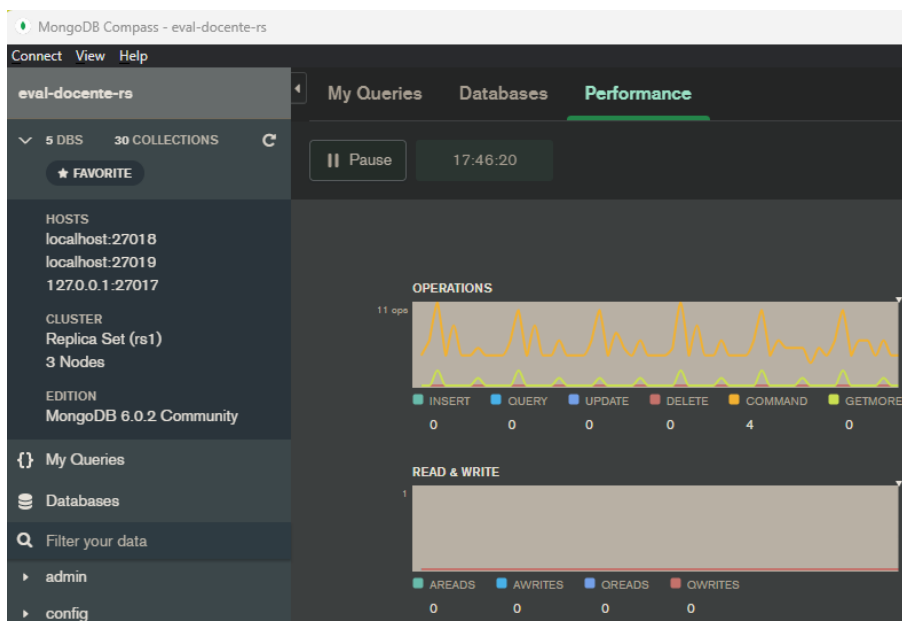
Figura 41. *docker-compose.yml* – Configuración MongoDB

Autor: David Constante

Con el contenedor levantado es posible conectarse a la base de datos, ya sea mediante un cliente con interfaz gráfica como un cliente desde la línea de comandos, en este caso para gestionar y monitorear la base de datos se utilizó MongoDB Compass.



*Figura 42. Conexión a MongoDB Compass
Autor: David Constante*



*Figura 43. Base de datos desde MongoDB Compass
Autor: David Constante*

Configurar Backend

Para dar consistencia a la arquitectura que se puede ver en la Figura 22, se creó un sistema de carpetas dentro de Visual Studio Code, este sistema contiene una carpeta para cada funcionalidad, como son: controladores, base de datos, configuración de los contenedores Docker, helpers o métodos auxiliares, middlewares y rutas de la aplicación. Para evitar las malas prácticas e incurrir en el “spanglish”, las funciones, nombres de variables o propiedades fueron declaradas con nombres en inglés, así como los commits de cada funcionalidad.

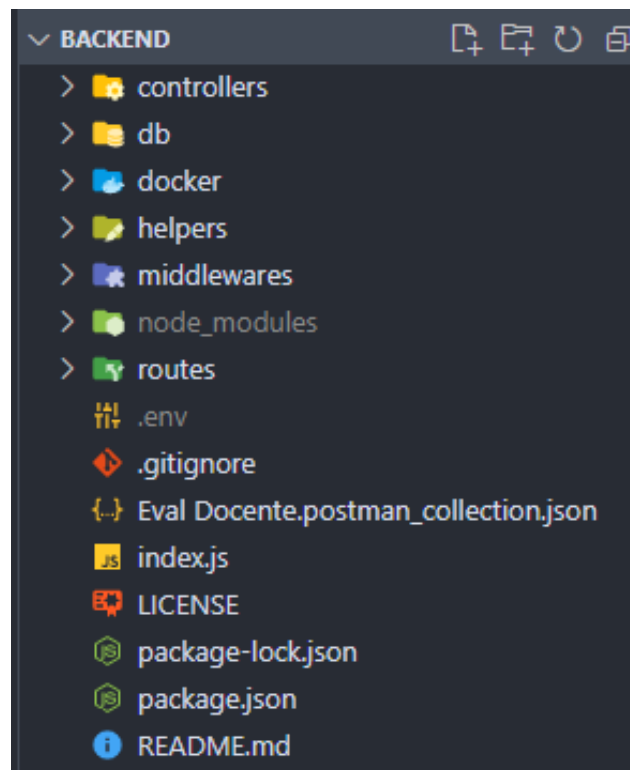


Figura 44. Backend de la Aplicación

Autor: David Constante

Configuración inicial del API

Para que la aplicación pueda enlazarse con la base de datos se creó un módulo que gestiona únicamente las conexiones, este usa dotenv para generar variables de entorno dentro del entorno de desarrollo y del sistema de producción, haciendo que las credenciales de acceso a la base de datos sean seguras y así evitar posibles fallos de seguridad relacionados a la autenticación. Ya que el Proceso de conexión a la base de datos en un proceso asíncrono se creó toda la conexión en base a *async* y *await*,

```
1 const mongoose = require('mongoose')
2
3 const DB_PORT1 = process.env.DB_PORT1 || 27017
4 const DB_PORT2 = process.env.DB_PORT2 || 27018
5 const DB_PORT3 = process.env.DB_PORT3 || 27019
6 const DB_HOST = process.env.DB_HOST || 'localhost'
7 const DB_USER = process.env.DB_USER
8 const DB_PASS = process.env.DB_PASS
9
10 const dbConnection = async () => {
11   try {
12     await
13     mongoose.connect(`mongodb://${DB_USER}:${DB_PASS}@${DB_HOST}:${DB_PORT1},${DB_HOST}:${DB_PORT2},${DB_HOST}:${DB_PORT3}/eval-docente?
14     authMechanism=DEFAULT&authSource=admin&replicaSet=rs1&readPreference=primaryPreferred`, {
15       useNewUrlParser: true,
16       useUnifiedTopology: true
17     })
18     console.log('Database online')
19   } catch (error) {
20     console.log(error)
21   }
22 }
23 module.exports = {
24   dbConnection
25 }
26
```

Figura 45. Conexión a la Base de Datos

Autor: David Constante

Para la gestión de errores de desarrollo un middleware que permite gestionar los errores mediante boom, y darles un formato de respuesta estandarizado.

```
1
2  const logError = (error, req, res, next) => {
3    // console.log(error)
4    next(error)
5  }
6
7  const boomErrorHandler = (error, req, res, next) => {
8    if (!error.isBoom) {
9      next(error)
10   }
11   const { output: { statusCode, payload } } = error
12   res.status(statusCode).json(payload)
13 }
14
15 const errorHandler = (error, req, res, next) => {
16   res.status(error.status || 500).json({
17     ok: false,
18     message: error.message,
19     stack: error.stack
20   })
21 })
22 }
23
24 module.exports = {
25   logError,
26   errorHandler,
27   boomErrorHandler
28 }
29 }
30
```

Figura 46. Middleware manejo de errores

Autor: David Constante

Dentro del servidor principal se incluyeron las dependencias y librerías necesarias para el funcionamiento de la API como tal, dotenv se utilizó para configurar el puerto, se creó un servidor mediante express el cual tiene en su ruta principal muestra información relacionada al API, este endpoint se utilizó para verificar que el backend este activo.

```

1  const express = require('express')
2  const morgan = require('morgan')
3  const cors = require('cors')
4  const pkg = require('./package.json')
5
6  require('dotenv').config()
7
8  const { dbConnection } = require('./db/config')
9  const routerAPI = require('./routes')
10 const { logError, boomErrorHandler, errorHandler } = require('./middlewares/error.handler')
11
12 /* Create a new express application instance
13 const app = express()
14
15 /* Connect to database
16 dbConnection()
17
18 /* Middlewares
19 app.use(express.json()) // Parse incoming requests data to JSON
20 app.use(morgan('dev')) // HTTP request logger middleware for node.js
21
22 /* Cors
23 app.use(cors())
24
25 /* Route principal (info)
26 app.set('pkg', pkg)
27 app.get('/', (req, res, next) => {
28   res.json({
29     name: app.get('pkg').name,
30     author: app.get('pkg').author,
31     description: app.get('pkg').description,
32     version: app.get('pkg').version
33   })
34 })
35
36 /* Routes
37 routerAPI(app)
38
39 /* Error handler
40 app.use(logError)
41 app.use(boomErrorHandler)
42 app.use(errorHandler)
43
44 /* Start server
45 app.listen(process.env.PORT, () => {
46   console.log(`Server on port ${process.env.PORT}`)
47 })
48

```

Figura 47. Servidor Inicial

Autor: David Constante

```
localhost:3000
1 // 20221207181724
2 // http://localhost:3000/
3
4 {
5   "name": "evaldocente-backend",
6   "author": "David Constante
7   "description": "Api for the Eval Docente",
8   "version": "1.0.0"
9 }
```

Figura 48. Servidor corriendo en express
Autor: David Constante

Epic 3 - API

The screenshot shows a Jira board for the 'API' epic. At the top, it indicates the sprint is from Dec 7 to Dec 15, with 4 incidents. The main goal is 'Crear los endpoints necesarios para la aplicación.' There are four tasks listed, each with a 'TAREAS POR HACER' status and assigned to 'DC':

- ED-8: Crear modelos para mongoose (API)
- ED-9: Crear controladores para cada entidad (API)
- ED-10: Crear rutas para cada entidad (API)
- ED-11: Crear middlewares para las respuestas (API)

A '+ Crear incidencia' button is visible at the bottom left.

Figura 49. Epic 3, Sprint 3 – Desarrollo Api
Autor: David Constante

Crear modelos para mongoose

Para interactuar con la base de datos se utilizó la biblioteca de mongoose, esta biblioteca es quien se encarga de interactuar con la base de datos mediante esquemas de datos conocidos como modelos, los modelos permiten establecer cuáles serán los atributos de cada colección y así una vez instanciado el modelo, mongoose se encarga de crearlo en la base de datos. Además de que dicha instancia es la que permite interactuar mediante métodos preestablecidos por la librería o queries nativos de MongoDB.

```
1  const { model, Schema } = require('mongoose')
2
3  const CommentSchema = Schema({
4    user: {
5      allowNull: false,
6      type: Schema.Types.ObjectId,
7      ref: 'Users'
8    },
9    teacher: {
10     allowNull: false,
11     type: Schema.Types.ObjectId,
12     ref: 'Teachers'
13   },
14   comment: {
15     allowNull: false,
16     type: Schema.Types.String
17   },
18   visible: {
19     allowNull: false,
20     type: Schema.Types.Boolean,
21     default: true
22   },
23   reactions: {
24     allowNull: true,
25     type: Schema.Types.Mixed,
26     default: {
27       likes: [],
28       dislikes: []
29     }
30   }
31 }, {
32   timestamps: true
33 })
34
35 module.exports = model('Comments', CommentSchema)
36
```

Figura 50. Esquema de modelo – comentarios

Autor: David Constante

Los modelos tienen todas las propiedades necesarias, así como las relaciones entre colecciones para mantener la integridad de la aplicación. Se crearon bajo el mismo esquema los modelos para las entidades de: comentarios, evaluaciones, materias, docentes y estudiantes.

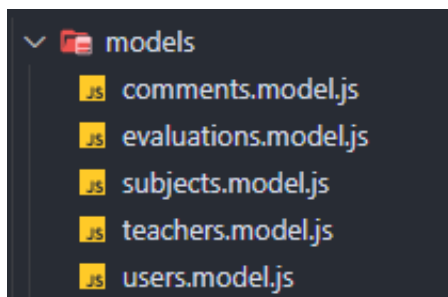


Figura 51. Modelos entidades
Autor: David Constante

Crear controladores para cada entidad

Los controladores de cada entidad abarcan toda la lógica de negocio de la aplicación, estos son la interfaz de comunicación entre los modelos y las rutas con sus respectivos endpoints. Cada controlador posee los métodos necesarios para realizar todas las operaciones CRUD con la entidad que le corresponde, además de que existe un controlador adicional encargado de verificar autenticación de los usuarios en la plataforma.

```
1  const getComments = async (req, res, next) => {
2    try {
3      const comments = await CommentSchema.find()
4      res.json(comments)
5    } catch (error) {
6      next(boom.badRequest(error))
7    }
8  }
9
```

Figura 52. Método obtener comentarios
Autor: David Constante

```

1  const getCommentsByTeacherId = async (req, res, next) => {
2    try {
3      const comment = await CommentSchema.find({ teacher: req.params.id })
4      if (!comment) return next(boom.notFound('Comment not found'))
5
6      res.status(200).json({
7        ok: true,
8        comment
9      })
10   } catch (error) {
11     next(boom.badRequest(error))
12   }
13 }

```

Figura 53. Método obtener comentarios mediante un id de docente

Autor: David Constante

```

1  const createComment = async (req, res, next) => {
2    try {
3      const comment = new CommentSchema(req.body)
4      const session = await startSession()
5
6      // Verify comments remaining
7      const remainingComments = await UserSchema.findById(comment.user, 'remainingComments')
8      if (remainingComments.remainingComments <= 0) next(boom.badRequest('no remaining comments'))
9
10     session.startTransaction()
11     await UserSchema.findByIdAndUpdate({ _id: req.body.user }, { $inc: { remainingComments: -1 } })
12     await comment.save()
13
14     session.commitTransaction()
15     session.endSession()
16
17     res.status(201).json({
18       ok: true,
19       msg: 'Comment created successfully',
20       comment
21     })
22   } catch (error) {
23     next(boom.badRequest(error))
24   }
25 }

```

Figura 54. Método crear comentario

Autor: David Constante


```

1  const likeComment = async (req, res, next) => {
2    let rta = null
3    try {
4      // Verify if comment exists
5      const comment = await CommentSchema.findById(req.params.id)
6      if (!comment) next(boom.notFound('Comment not found'))
7
8      // Verify if user already reacted the comment
9      if (comment.reactions.dislikes.includes(req.body.user)) next(boom.badRequest('You already reacted this comment'))
10
11     // Verify if user don't liked the comment
12     if (!comment.reactions.likes.includes(req.body.user)) {
13       rta = await CommentSchema.findByIdAndUpdate(req.params.id, {
14         $push: { 'reactions.likes': req.body.user }
15       }, { new: true })
16       res.status(200).json({
17         ok: true,
18         msg: 'Comment liked successfully',
19         rta
20       })
21     } else { // Remove reaction
22       rta = await CommentSchema.findByIdAndUpdate(req.params.id, {
23         $pull: { 'reactions.likes': req.body.user }
24       }, { new: true })
25       res.status(200).json({
26         ok: true,
27         msg: 'Comment unliked successfully',
28         rta
29       })
30     }
31   } catch (error) {
32     next(boom.badRequest(error))
33   }
34 }

```

Figura 55. Método like comentario

Autor: David Constante

```

1  const dislikeComment = async (req, res, next) => {
2    let rta = null
3    try {
4      // Verify if comment exists
5      const comment = await CommentSchema.findById(req.params.id)
6      if (!comment) next(boom.notFound('Comment not found'))
7
8      // Verify if user already reacted the comment
9      if (comment.reactions.likes.includes(req.body.user)) next(boom.badRequest('You already reacted this comment'))
10
11     // Verify if user don't disliked the comment
12     if (!comment.reactions.dislikes.includes(req.body.user)) {
13       rta = await CommentSchema.findByIdAndUpdate(req.params.id, {
14         $push: { 'reactions.dislikes': req.body.user }
15       }, { new: true })
16       res.status(200).json({
17         ok: true,
18         msg: 'Comment liked successfully',
19         rta
20       })
21     } else { // Remove reaction
22       rta = await CommentSchema.findByIdAndUpdate(req.params.id, {
23         $pull: { 'reactions.dislikes': req.body.user }
24       }, { new: true })
25       res.status(200).json({
26         ok: true,
27         msg: 'Comment unliked successfully',
28         rta
29       })
30     }
31   } catch (error) {
32     next(boom.badRequest(error))
33   }
34 }

```

Figura 56. Método dislike comentario

Autor: David Constante

```

1  const updateComment = async (req, res, next) => {
2    try {
3      // Verify if comment exists
4      const comment = await CommentSchema.findById(req.params.id)
5      if (!comment) {
6        return res.status(404).json({
7          ok: false,
8          msg: 'Comment not found'
9        })
10     }
11
12     // Update comment
13     await CommentSchema.findByIdAndUpdate(req.params.id, req.body)
14
15     res.status(200).json({
16       ok: true,
17       msg: 'Comment updated successfully'
18     })
19   } catch (error) {
20     next(boom.badRequest(error))
21   }
22 }

```

Figura 57. Método actualizar comentario

Autor: David Constante

```

1  const deleteComment = async (req, res, next) => {
2    try {
3      // Verify if comment exists
4      const comment = await CommentSchema.findById(req.params.id)
5      if (!comment) {
6        return res.status(404).json({
7          ok: false,
8          msg: 'Comment not found'
9        })
10     }
11
12     // Delete comment
13     await CommentSchema.findByIdAndDelete(req.params.id)
14
15     res.status(200).json({
16       ok: true,
17       msg: 'Comment deleted successfully'
18     })
19   } catch (error) {
20     next(boom.badRequest(error))
21   }
22 }

```

Figura 58. Método eliminar comentario

Autor: David Constante

Crear las rutas para cada entidad.

Las diferentes entidades: comentarios, evaluaciones, materias, docentes y estudiantes tienen su propio manejador de rutas, todos estos gestores de rutas convergen en un gestor general que es el que contiene distribuye las llamadas de los endpoints a su respectivo router. Además, se utilizó un esquema de versiones mediante la sintaxis: *api/v1*. Esta práctica está relacionada con la escalabilidad de la plataforma, ya que permitiría trabajar con dos versiones diferentes del backend en un mismo proyecto en caso de que los requerimientos institucionales cambien en algún punto del tiempo.

```
1  const express = require('express')
2
3  const userRouter = require('./user.routes')
4  const authRouter = require('./auth.routes')
5  const teacherRouter = require('./teacher.routes')
6  const evaluationRouter = require('./evaluations.routes')
7  const commentRouter = require('./comments.routes')
8  const subjectRouter = require('./subject.routes')
9
10 const routerAPI = (app) => {
11   const router = express.Router()
12
13   app.use('/api/v1', router)
14
15   router.use('/users', userRouter)
16   router.use('/auth', authRouter)
17   router.use('/teachers', teacherRouter)
18   router.use('/evaluations', evaluationRouter)
19   router.use('/comments', commentRouter)
20   router.use('/subjects', subjectRouter)
21 }
22
23 module.exports = routerAPI
24
```

Figura 59. Rutas generales

Autor: David Constante

```

1  const { Router } = require('express')
2  const {
3    getComments,
4    getCommentsByTeacherId,
5    createComment,
6    likeComment,
7    dislikeComment,
8    updateComment,
9    deleteComment
10 } = require('../controllers/comments.controller')
11
12 const { validateJWT } = require('../middlewares/validate.jwt')
13 const { isStudent } = require('../middlewares/validate.roles')
14
15 /*
16  host + /api/v1/comments
17  */
18 const router = Router()
19
20 router.get('/', getComments)
21
22 router.get('/:id', getCommentsByTeacherId)
23
24 router.post('/new', [
25   validateJWT,
26   isStudent
27 ], createComment)
28
29 router.post('/like/:id', [
30   validateJWT,
31   isStudent
32 ], likeComment)
33
34 router.post('/dislike', [
35   validateJWT,
36   isStudent
37 ], dislikeComment)
38
39 router.put('/:id', [
40   validateJWT,
41   isStudent
42 ], updateComment)
43
44 router.delete('/:id', [
45   validateJWT,
46   isStudent
47 ], deleteComment)
48
49 module.exports = router

```

Figura 60. Rutas comentarios

Autor: David Constante

Crear middlewares para las respuestas.

Para mantener el principio de “single responsibility” se crearon middlewares para manejar las respuestas de la API. Para gestionar las sesiones de los usuarios que se autentican en la plataforma se utilizó JWT, esto permite generar un token que se almacenara en la sesión de navegación y se envía en cada una de las peticiones para así garantizar la seguridad de la aplicación, y restringir las tareas de los diferentes tipos de usuarios.

```
1  const jwt = require('jsonwebtoken')
2
3  const SECRET_JWT_SEED = process.env.SECRET_JWT_SEED
4
5  const generateJWT = (username, email) => {
6    return new Promise((resolve, reject) => {
7      const payload = { username, email }
8
9      jwt.sign(payload, SECRET_JWT_SEED, {
10       expiresIn: '4h'
11     }, (err, token) => {
12       if (err) {
13         console.log(err)
14         // eslint-disable-next-line prefer-promise-reject-errors
15         reject('Could not generate token')
16       }
17       resolve(token)
18     })
19   })
20 }
21
22 module.exports = {
23   generateJWT
24 }
25
```

Figura 61. Generar Json Web Token

Autor: David Constante

```

1  const { response } = require('express')
2  const jwt = require('jsonwebtoken')
3  const boom = require('@hapi/boom')
4
5  const validateJWT = (req, res = response, next) => {
6    // x-token headers
7    const token = req.header('x-token')
8
9    if (!token) return next(boom.badRequest('There is no token in the request'))
10
11   try {
12     const { username, email } = jwt.verify(token, process.env.SECRET_JWT_SEED)
13
14     req.username = username
15     req.email = email
16   } catch (error) {
17     return next(boom.badRequest('Invalid token'))
18   }
19
20   next()
21 }
22
23 module.exports = {
24   validateJWT
25 }
26

```

Figura 62. Validación del Json Web Token

Autor: David Constante

Con la finalidad de añadir robustez a la API, además de reducir la carga de trabajo separando las peticiones que tengan inconsistencias, se realizaron esquemas de validación con Joi para cada una de las entidades, de esta manera se compara el cuerpo de cada petición entrante y en caso de no ser el adecuado la petición se rechaza inmediatamente. Estos esquemas se realizaron para todas las entidades: comentarios, evaluaciones, materias, docentes y estudiantes

```
1  const Joi = require('joi')
2
3  const user = Joi.string().min(3).max(30)
4  const teacher = Joi.string().min(3).max(30)
5  const date = Joi.date()
6  const comment = Joi.string().min(3).max(144)
7  const visible = Joi.boolean()
8  const reactions = Joi.object()
9
10 /* Validators
11
12 const createCommentFields = Joi.object({
13   user: user.required(),
14   teacher: teacher.required(),
15   date: date.required(),
16   comment: comment.required(),
17   visible: visible.required(),
18   reactions: reactions.required()
19 })
20
21 const updateCommentFields = Joi.object({
22   user,
23   teacher,
24   date,
25   comment,
26   visible,
27   reactions
28 })
29
30 module.exports = {
31   createCommentFields,
32   updateCommentFields
33 }
34
```

Figura 63. Esquema Joi comentarios

Autor: David Constante

```

1  const boom = require('@hapi/boom')
2
3  const validateReq = (schema, property) => {
4    return (req, res, next) => {
5      const data = req[property]
6      const { error } = schema.validate(data, { abortEarly: false })
7      if (error) {
8        next(boom.badRequest(error))
9      } else {
10       next()
11     }
12   }
13 }
14
15 module.exports = validateReq
16

```

Figura 64. Middleware validar petición

Autor: David Constante

Además, se crearon middlewares para verificar los roles de los diferentes usuarios de la aplicación.

```

1  const isGuest = async (req, res, next) => {
2    try {
3      const { username } = req
4      const user = await userSchema.findOne({ username })
5      if (!user) return next(boom.badRequest('User not found'))
6
7      if (user.roles.includes('guest')) return next()
8
9      next(boom.unauthorized('You are not authorized to perform this action'))
10   } catch (error) {
11     next(boom.badRequest(error))
12   }
13 }
14

```

Figura 65. Rol invitado

Autor: David Constante

Pruebas

Epic 3 – API Test

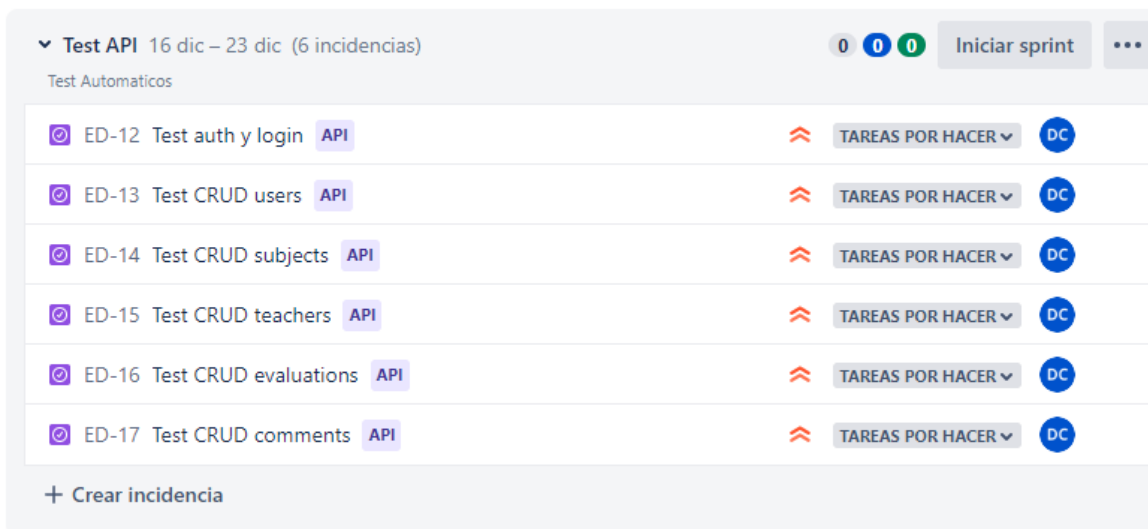


Figura 66. Epic 3, Sprint 4

Autor: David Constante

Para la verificación del estado y funcionamiento de la plataforma se incorporaron test automatizados en Jest, que se ejecutan según el entorno en el que se encuentre. Es importante definir las variables de entorno para que la plataforma pueda acceder, ya sea al servidor de producción como al servidor local de desarrollo.

```
1 describe('Test in evalDocenteApi', () => {
2   test('should return the port', async () => {
3     expect(evalDocenteApi.defaults.baseURL.split(':')[2].split('/')[0]).toBe('2000')
4   })
5 })
```

Figura 67. Test en el puerto de la API

Autor: David Constante

Test

```
1  /* eslint-disable no-undef */
2  import { evalDocenteApi } from '../api/evalDocenteApi'
3
4  describe('Test in evalDocenteApi', () => {
5    const userTest = {
6      username: 'david',
7      email: 'david@uta.com'
8    }
9
10   test('Test in authentication', async () => {
11     const rta1 = await evalDocenteApi.post('/auth/username', { username: 'david', password: '123456' })
12     const { token } = rta1.data
13     localStorage.setItem('token', token)
14
15     const userRta1 = {
16       username: rta1.data.user.username,
17       email: rta1.data.user.email
18     }
19
20     expect(userTest).toEqual(userRta1)
21
22     const rta2 = await evalDocenteApi.get('/auth/renew')
23
24     const userRta2 = {
25       username: rta2.data.user.username,
26       email: rta2.data.user.email
27     }
28
29     expect(userTest).toEqual(userRta2)
30   })
31
32   test('Test in renew', async () => {
33     const rta1 = await evalDocenteApi.post('/auth/username', { username: 'david', password: '123456' })
34     const { token } = rta1.data
35     localStorage.setItem('token', token)
36
37     const rta2 = await evalDocenteApi.get('/auth/renew')
38
39     const userRta2 = {
40       username: rta2.data.user.username,
41       email: rta2.data.user.email
42     }
43
44     expect(userTest).toEqual(userRta2)
45   })
46 })
```

Figura 68. Test login y renew session.

Autor: David Constante

El test verifica el inicio de sesión con un usuario de prueba, así como el token generado con el cual el usuario podrá restaurar su sesión automáticamente.

Esta acción se repite con todas las entidades de la aplicación, se utilizó un esquema general en donde los test conviven con los módulos dentro del mismo directorio, esto debido a que los test son los necesarios para evaluar la ruta crítica de funcionamiento.

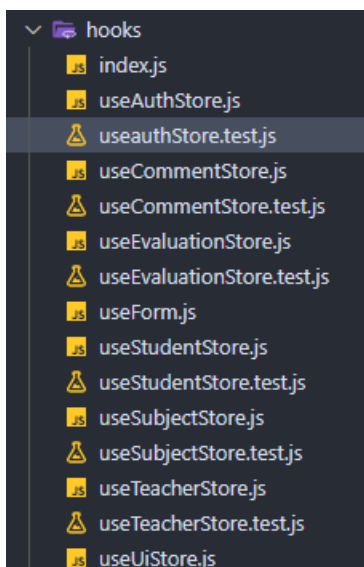


Figura 69. Sistema de archivos de módulos y test

Autor: David Constante

Las entidades users, subjects, teaches, evaluations y comments. Tienen su propio módulo de test que evalúa el funcionamiento de la API, esta práctica mejora el despliegue de la aplicación debido a que antes de ejecutar el sistema en el servidor de producción los test corren de manera automática y así, además de verificar el correcto funcionamiento estos evalúan el rendimiento del servidor.

```
C:\WINDOWS\system32\cmd.exe x PowerShell x + v
✓ Test in renew (188 ms) ... useauthStore.test.js x
PASS src/api/evalDocenteApi.test.js
  Test in evalDocenteApi
    ✓ should return the port (2 ms)
GROUP 2
PASS src/hooks/useStudentStore.test.js
  Test in useStudentStore
    ✓ Test in createStudent (712 ms)
    ✓ Test in getStudents (39 ms)
    ✓ Test in updateStudent (46 ms)
    ✓ Test in deleteStudent (37 ms)
    ✓ Test in performance (783 ms)
  > docker
PASS src/hooks/useTeacherStore.test.js
  Test in useTeacher
    ✓ Test in createTeacher (560 ms)
    ✓ Test in getTeacher (155 ms)
    ✓ Test in updateTeacher (56 ms)
    ✓ Test in deleteTeacher (55 ms)
    ✓ Test in performance Teachers (1087 ms)
  > helpers
PASS src/hooks/useSubjectStore.test.js (5.556 s)
  Test in useStudentStore
    ✓ Test in createSubject (705 ms)
    ✓ Test in getSubject (56 ms)
    ✓ Test in updateSubject (36 ms)
    ✓ Test in deleteSubject (42 ms)
    ✓ Test in performanceSubject (4158 ms)
  UseCommentStore.js
PASS src/hooks/useCommentStore.test.js (6.513 s)
  Test in useCommentStore
    ✓ Test in createComment (750 ms)
    ✓ Test in getComment (36 ms)
    ✓ Test in updateComment (40 ms)
    ✓ Test in deleteComment (114 ms)
    ✓ Test in performance comments (5011 ms)
  useStudentStore.test.js
PASS src/hooks/useEvaluationStore.test.js (7.204 s)
  Test in useEvaluation
    ✓ Test in createEvaluation (770 ms)
    ✓ Test in getEvaluation (37 ms)
    ✓ Test in updateEvaluation (141 ms)
    ✓ Test in deleteEvaluation (114 ms)
    ✓ Test in performance evaluations (5581 ms)
  useTeacherStore.test.js
Test Suites: 7 passed, 7 total
Tests: 28 passed, 28 total
Snapshots: 0 total
Time: 10.1 s
Ran all test suites.
```

Figura 70. Ejecución de los test

Autor: David Constante

Epic 4 - Frontend

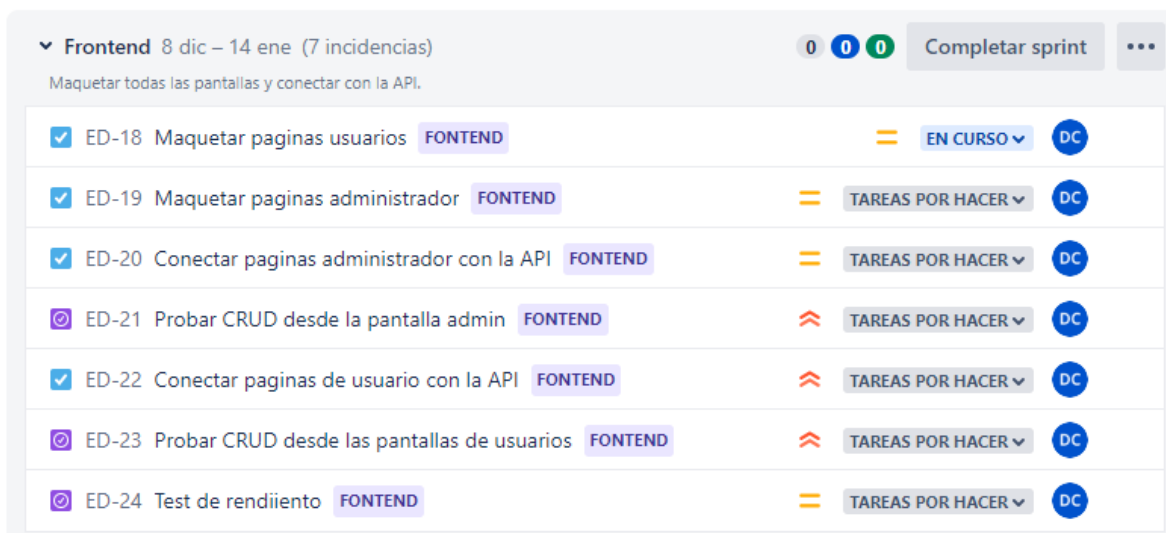


Figura 71. Epic 4, Sprint 5

Autor: David Constante

Como se puede verificar en el Anexo 3, la plataforma sigue una misma línea de diseño en donde lo único que cambia es el tamaño y disposición de los elementos, es por ello que el maquetado de la aplicación se realizó en base a componentes que son reutilizados y en algunos casos re-renderizados por el navegador según el tamaño del dispositivo del que se acceda a la misma.

La plataforma consta de cuatro páginas con las que el usuario puede interactuar, una Página Principal, una Página Docente, una Página Login y Una página de Administración. Los elementos de la página se desarrollaron bajo la modalidad mobile-first, y estos reaccionan automáticamente al cambio de tamaño de la pantalla, en donde los breakpoints utilizados están definidos en la tabla 9.

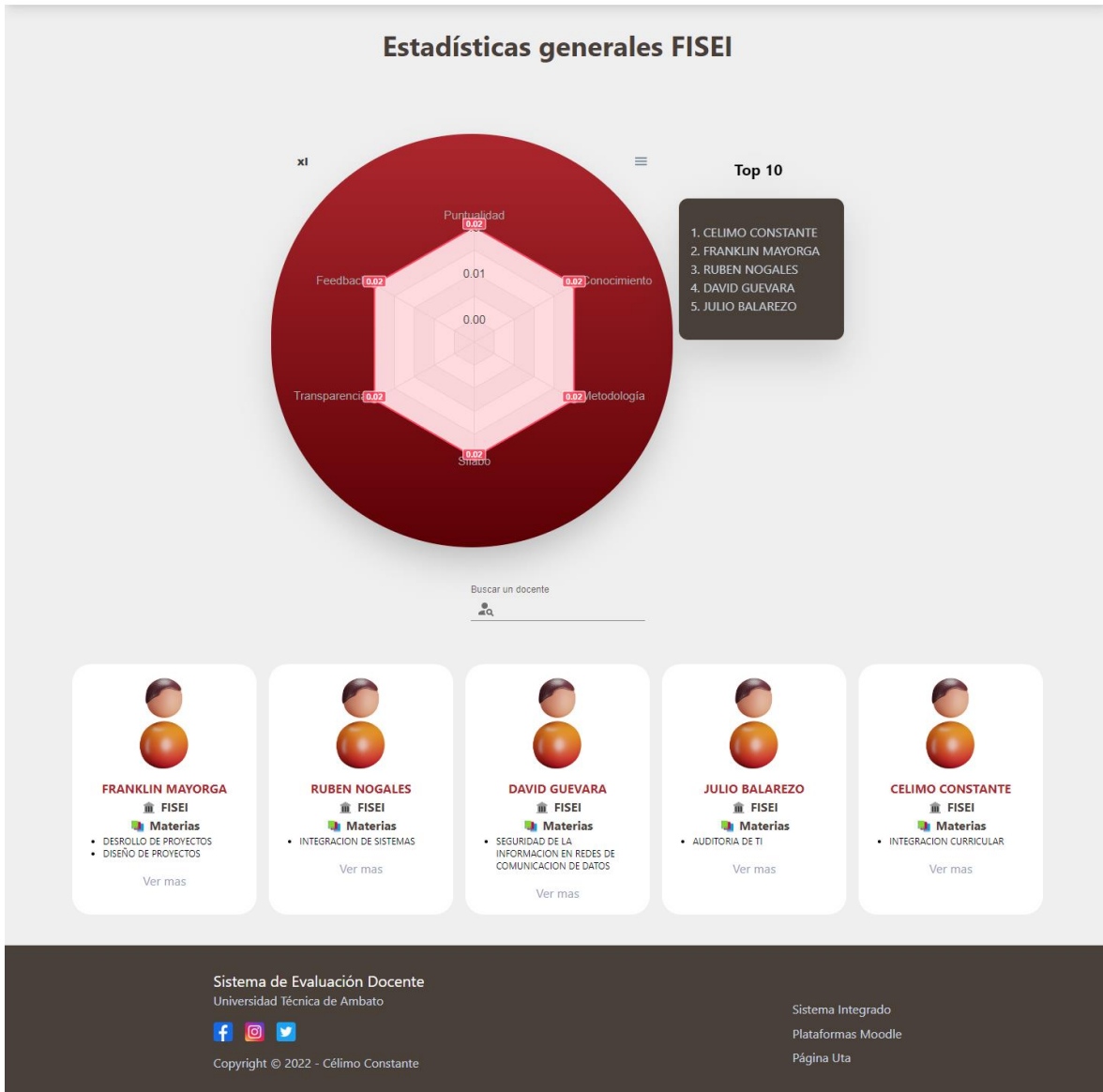


Figura 72. Maquetado Home Page

Autor: David Constante

CELIMO CONSTANTE

xl☰

Categoría	Puntuación
Puntualidad	0.4
Conocimiento	0.4
Metodología de enseñanza	0.4
Silabo	0.4
Transparencia	0.4
Feedback	0.4

Puntualidad	?
Conocimiento	?
Metodología de enseñanza	?
Silabo	?
Transparencia	?
Feedback	?

Evaluar

Comentarios

No hay comentarios
Se el primero en comentar

Comparte tu opinión con la comunidad universitaria...

Sistema de Evaluación Docente
Universidad Técnica de Ambato

Copyright © 2022 - Célimo Constante

Sistema Integrado
Plataformas Moodle
Página Uta

Figura 73. Maquetado Teacher Page

Autor: David Constante

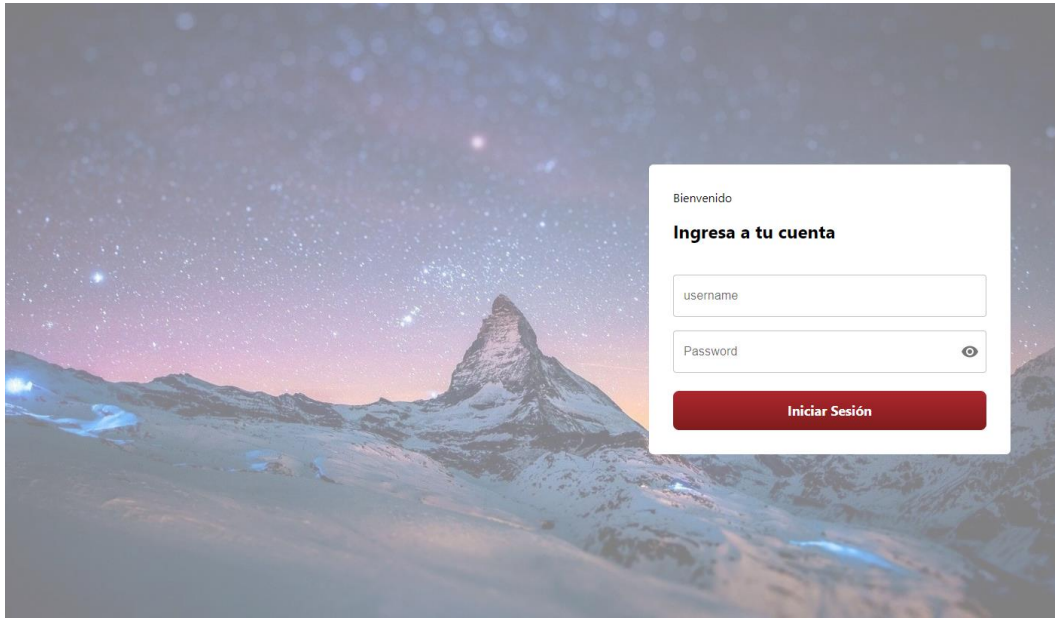


Figura 74. Maquetado Login Page

Autor: David Constante

Sistema de Evaluación Docente CC

Estudiantes

Materias

Docentes

Evaluaciones

Comentarios

ESTUDIANTES							🔍 📄 🖨️ 📊 ☰
ID	Nombre	Apellido	Facultad	Comentarios	Evaluaciones	Acciones	
6337849af9f51209080cf3bc	David	Constante Procel	FISEI	5	4	✎ 🗑️	
6337872231770bccdd6329682	Celimo	Constante Procel	FISEI	5	5	✎ 🗑️	
63380ca0659d3b56794d8731	Timmy	Constante Procel	FISEI	5	5	✎ 🗑️	
636d4935965629a942b39406	LUIS	SALAZAR	FISEI	5	5	✎ 🗑️	
636d4951965629a942b39409	FELIPE	MONAR	FISEI	5	5	✎ 🗑️	
636d4976965629a942b3940c	ANTHONY	PEREZ	FISEI	5	5	✎ 🗑️	
636d4982965629a942b3940f	ANTHONY	LOPEZ	FISEI	5	5	✎ 🗑️	
636d4997965629a942b39412	PAUL	GONZALES	FISEI	5	5	✎ 🗑️	
636d4ead43d682517dd4dd58	ESTEFANNY ARACELY	BOLAÑOS ARCOS	SALUD	5	5	✎ 🗑️	
63af7347705eb698c1860b2c	CELIMO DAVID	CONSTANTE PROCEL	IDIOMAS, FISEI	5	5	✎ 🗑️	

Rows per page: 10 ▾ 1-10 of 10 < >

REINICIAR COMENTARIOS
REINICIAR EVALUACIONES
AÑADIR ESTUDIANTE

Figura 75. Maquetado Manage Page

Autor: David Constante

Una vez enlazadas a la API de la plataforma, tanto la página de manejo como la de los usuarios convencionales pueden realizar todas las acciones permitidas por la plataforma, debido a que la página tiene acceso público, todos los usuarios podrán ver las estadísticas y comentarios, pero solo los usuarios autenticados (estudiantes reales) son quienes pueden comentar, reaccionar a comentarios o evaluar a los docentes. A medida que se evalúen docentes las estadísticas generales del docente y de todos los docentes también se verán afectadas.

Lanzamiento

Epic 5 - Despliegue

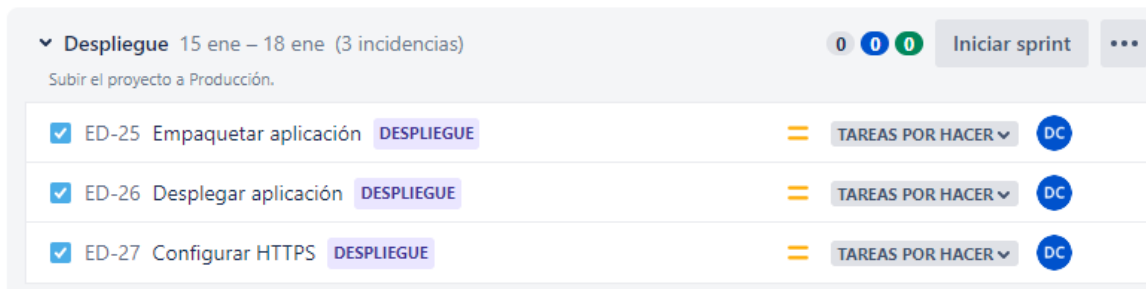


Figura 76. Epic 5, Sprint 6

Autor: David Constante

Debido se utilizó Git y Github como aplicación y servicio de control de versiones respectivamente, y como se especifica en la tabla 4; una de las características de la ya definida metodología de desarrollo híbrida adaptada al proyecto es la Integración continua o CI. Para llevar a cabo dicha tarea se configuraron dos actions runners dentro del servidor de producción enlazados a los repositorios del frontend y backend que se encuentran alojados en github, estos runners mantienen un socket que escucha todos los cambios que se han producido dentro del repositorio, y si es necesario se encargan de volver a clonar el repositorio y ejecutar las tareas ya configuradas para su empaquetado y despliegue automático. Esto permite tener un flujo de trabajo ágil, ya que para realizar un cambio dentro de la plataforma basta con publicar un cambio dentro de la rama principal del repositorio.

Así como la Integración continua puede ser una característica muy útil, esta también puede hacer que el servidor deje de funcionar. Por lo que durante el desarrollo se mantuvo un especial cuidado en la rama principal, ya que esa es un reflejo de lo que se encuentra en el servidor de producción, por lo que el desarrollo se realizó en ramas nombradas según la característica a desarrollar y estas se integraban secuencialmente en la rama dev.

```
1 name: Evaluacion Docente - Frontend CI
2
3 on:
4   push:
5     branches: [ "main" ]
6
7 jobs:
8   build-frontend:
9
10    runs-on: self-hosted
11
12    strategy:
13      matrix:
14        node-version: [18.x]
15
16    steps:
17      - name: Checkout Frontend
18        uses: actions/checkout@v3
19        with:
20          # path: main
21          node-version: ${ matrix.node-version }
22      - run: /
23          touch .env
24          echo VITE_API_URL_PROD=${ secrets.VITE_API_URL_PROD } >> .env
25
26      - run: yarn
27      - run: yarn build
28      - run: /
29          cd docker
30          docker compose -p eval-docente up -d --force-recreate
31
32
33
```

Figura 77. Action runner CI Backend

Autor: David Constante

Como parte del proceso de despliegue se utilizó un servidor nginx con un proxy inverso que redirige el tráfico puerto enlazado el contenedor correspondiente, el sitio está disponible en el dominio <https://omilec.com/> que fue adquirido por el investigador. La configuración https se realizó con la ayuda de certbot en su versión para nginx.

```
celimo@omilec:~$ cat /etc/nginx/sites-available/default
server{

    server_name omilec.com www.omilec.com;
    location / {
        proxy_pass http://localhost:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/omilec.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/omilec.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
# vim:ft=yaml
server{
    location / {
        proxy_pass http://localhost:2000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    listen 3000 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/omilec.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/omilec.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server{
    if ($host = www.omilec.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = omilec.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen *:80;

    server_name omilec.com www.omilec.com;
    return 404; # managed by Certbot
}
```

Figura 78. Configuración del proxy inverso nginx

Autor: David Constante

Gracias a la implementación de una metodología híbrida y el uso de herramientas para el manejo y administración de tareas el proyecto se desarrolló con normalidad, y se pudo completar una versión funcional en los plazos establecidos.

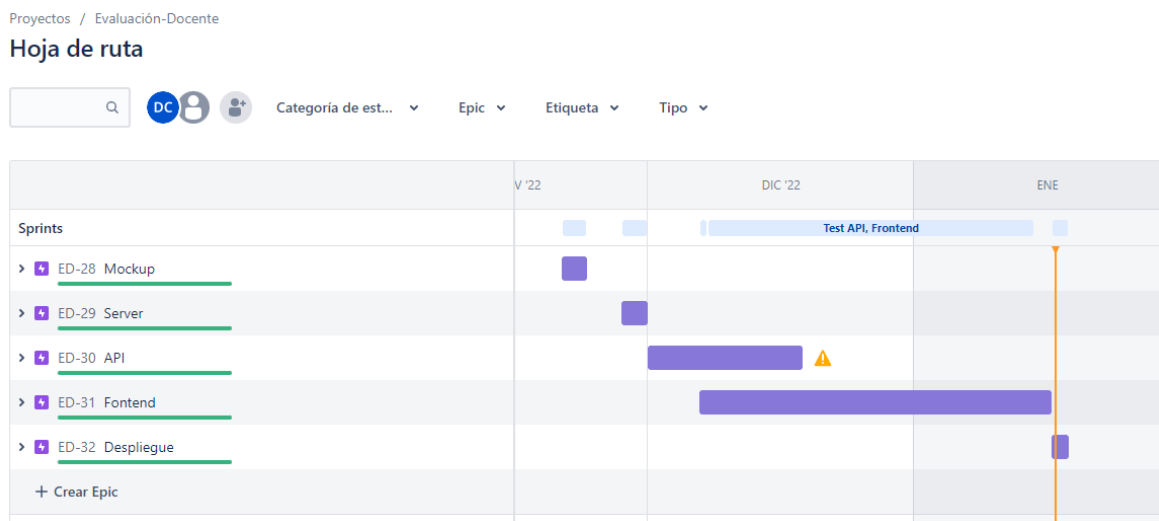


Figura 79. Hoja de ruta

Autor: David Constante

Pruebas de Aceptación

Dado que los estudiantes de la Universidad Técnica de Ambato son los principales usuarios de la plataforma desarrollada, como parte del proceso de desarrollo se realizó una reunión con 5 alumnos de los últimos semestres de la carrera de Tecnologías de la Información. Y una reunión con un docente de la misma carrera. Los estudiantes se vieron en la capacidad de usar el sistema aun sin leer el manual de usuario, lo que sugiere que el UI y UX tienen una integración sólida con los sistemas usados comúnmente por los mismos, a algunos estudiantes se les dificultó interpretar el gráfico inicial de la página, por lo que se adaptó el tamaño de los elementos principales dándole más realce al título principal y disminuyendo el tamaño del gráfico que muestra las estadísticas generales de la Universidad.

Por otra parte, el docente sugirió que sería útil un módulo en el cual se puedan comparar estadísticas de las diferentes facultades, es por ello que se decidió tomar en cuenta dicha observación y se desarrolló un nuevo módulo bajo el perfil de coordinador, el cual puede

acceder a una comparación de las diferentes facultades y utilizando dichos gráficos comparativos como mejor le convenga.

Es importante señalar que una de las métricas usadas para medir la mejora de la experiencia de usuario es la satisfacción del cliente, es decir de los usuarios finales de la aplicación. Los dos grupos seleccionados para las pruebas de aceptación dieron respuestas favorables al nuevo sistema de evaluación docente, a las funcionalidades y a la experiencia de usuario en general.

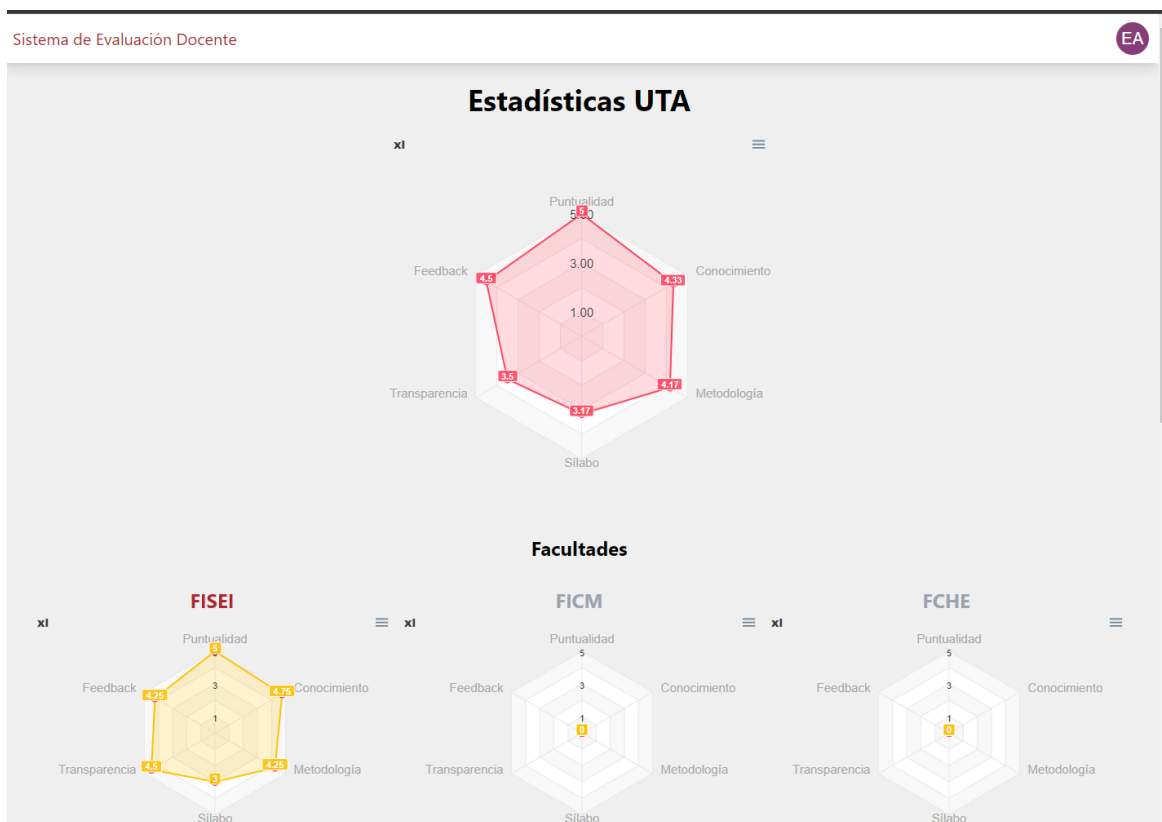


Figura 80. Página de estadísticas generales coordinador

Autor: David Constante

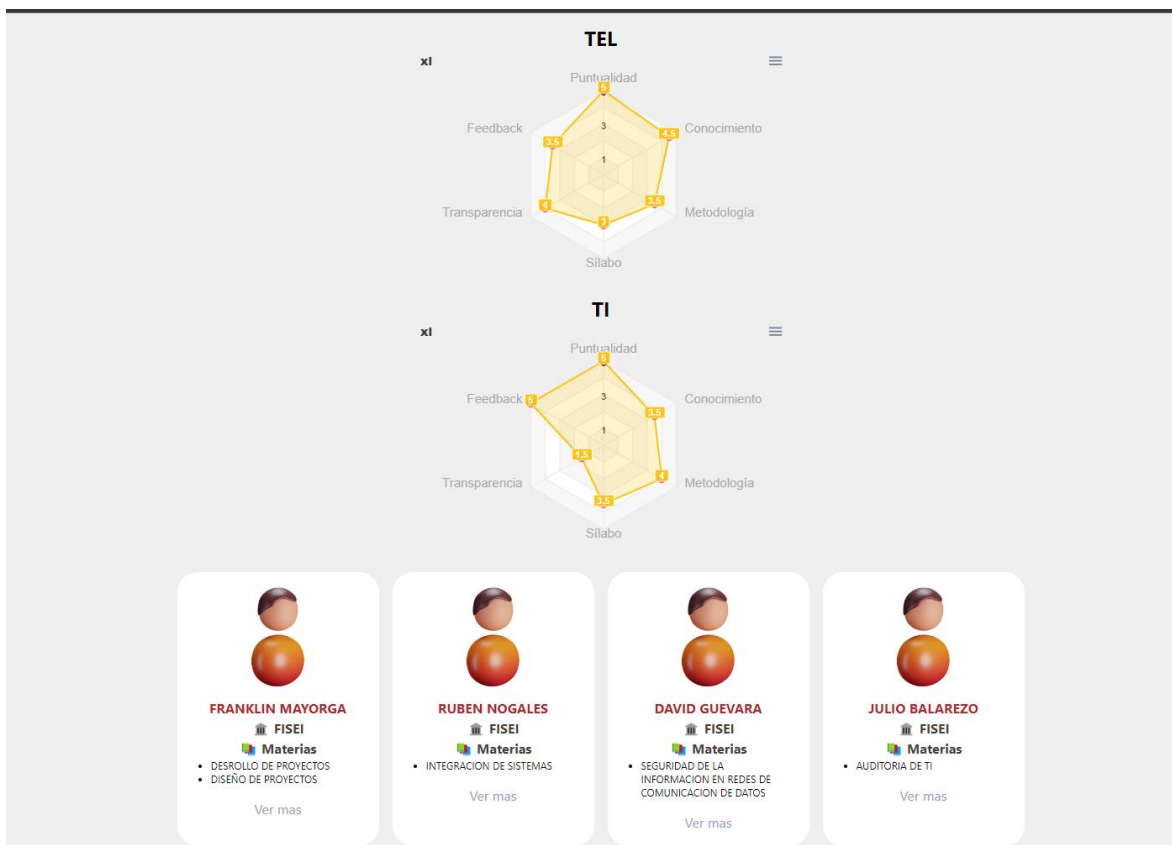


Figura 81. Página de estadísticas por carrera

Autor: David Constante

Mejoras en la Experiencia de usuario

Para evaluar la mejora de experiencia de usuario se tomaron en cuenta tres factores:

- Número de pasos para evaluar al docente.
- Dificultad de acceso al sistema (Escala de 1-5).
- Satisfacción de los usuarios (Escala de 1-5).

Las reuniones mantenidas con los estudiantes y docentes señalan que hubo una mejora significativa en el nuevo sistema de evaluación docente. Como se evidencia en la figura 82 el número de pasos que un estudiante debe realizar para evaluar a un docente se reducen a la mitad, lo que señala que el proceso se vio simplificado de manera significativa. Por otra parte, al hablar de la dificultad de acceso al nuevo sistema de evaluación docente es mucho más

simple, por lo que los usuarios se sienten más cómodos con el acceso a la plataforma. Por último, el nivel de satisfacción de los usuarios al manejar el nuevo sistema de evaluación docente es significativamente mayor al del sistema de evaluación docente antiguo. Debido a lo anteriormente señalado queda demostrado que si existió una mejora en el nuevo sistema de evaluación docente.

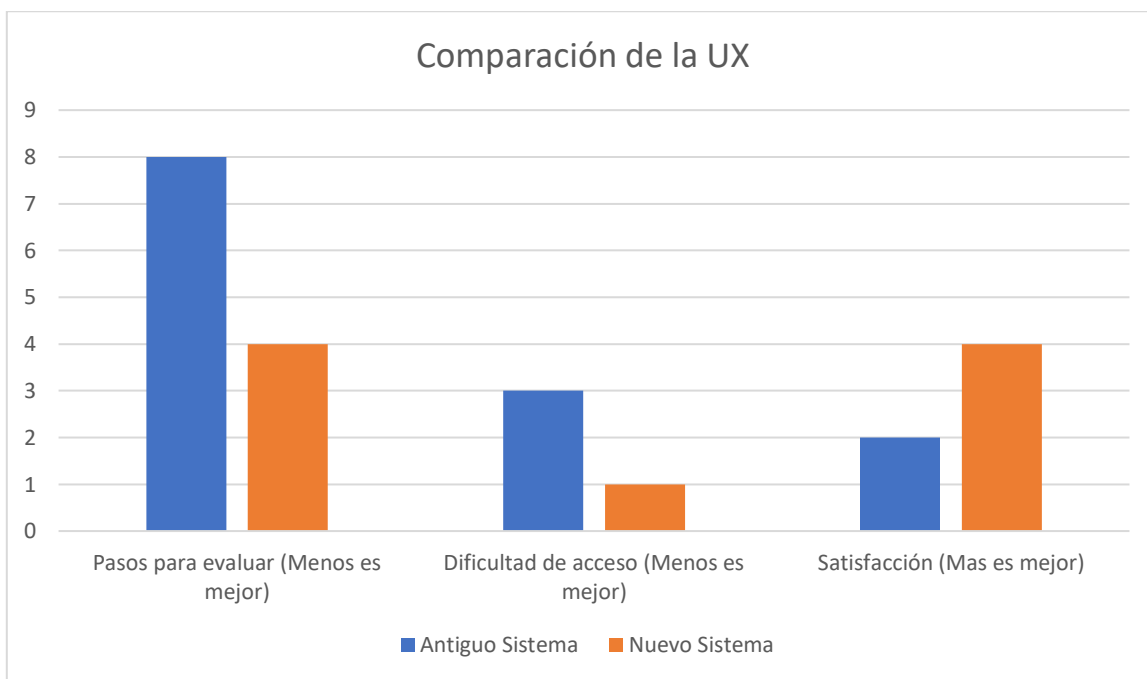


Figura 82. Comparación de la UX del sitio anterior vs nuevo

Autor: David Constante

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- El actual sistema de evaluación docente no posee una experiencia de usuario adecuada, los estándares de desarrollo van cambiando con el tiempo y es importante que la Universidad Técnica de Ambato esté a la vanguardia en cuanto a tecnologías, metodologías y patrones de desarrollo implementados en sus diferentes procesos.
- Las metodologías de desarrollo de software son esenciales para el manejo de equipos de trabajo, que no se tenga un equipo no significa que se debe desarrollar un proyecto sin ninguna metodología. Estas presentan convenciones que son utilizadas bajo determinados escenarios, es posible combinar lo mejor de diferentes metodologías según estas se adapten a la situación, necesidades y recursos del proyecto, y esto acelerará en gran medida el flujo de trabajo para tener como resultado lo antes posible el producto final.
- Usar el stack MERN facilitó el desarrollo del presente proyecto, ya que con la experiencia de otros desarrolladores y la suficiente documentación permitió crear una plataforma robusta, que al ser esta desplegada sobre una arquitectura basada en contenedores permite tener un sistema fácilmente escalable con diferentes servicios y plataformas que son agnósticas entre si.
- Las fases de planeación permiten tener una idea general del proyecto, es importante no solo cumplir las necesidades básicas, sino que también es necesario considerar la escalabilidad de la aplicación y desarrollar la misma con los cimientos necesarios para que esta pueda escalar de manera fácil sin la necesidad de mucha refactorización.
- Diseñar una plataforma con características a la vanguardia como en este caso, líneas de diseño y responsive, permite reducir la curva de aprendizaje para que los usuarios puedan manejar el sistema de manera intuitiva.
- La librería mongoose facilita la comunicación entre la base de datos y el backend, además react-redux permite mantener un estado general de la aplicación. Ambas librerías son una de las características principales a la hora de que la información manejada por el usuario se perciba como instantánea.

4.2. Recomendaciones

- Los entornos de desarrollo y producción son totalmente diferentes, es importante realizar monitoreos constantes para verificar el estado de la plataforma y si el caso lo amerita escalar el servidor.
- Se recomienda para una posterior implementación validar los puertos de desarrollo y producción a la hora de refactorizar código, es recomendable no ejecutar test en entornos de producción, en su defecto se puede configurar un nuevo servidor con características similares para evaluar el rendimiento.
- Se recomienda estandarizar el diseño de futuras aplicaciones en la misma línea de diseño de la universidad, para así tener una experiencia de usuario más consistente en todos los servicios de la universidad.
- Debido a que es un sitio de cara al internet es importante revisar constantemente la vigencia del dominio y la vigencia del certificado https.
- Es importante mantener especial cuidado con las ramas locales de trabajo, ya que estas están enlazadas al servidor principal y pueden hacer que el sitio deje de estar disponible. Debido a lo anteriormente señalado la seguridad se vuelve un factor primordial por lo que es importante configurar la seguridad de los repositorios de Github, así como las llaves ssh para acceder al servidor de producción.
- Se recomienda realizar pruebas de aceptación en grupos más grandes de usuarios para así poder obtener un feedback más adecuado y de ser el caso realizar las correcciones necesarias.
- En reuniones mantenidas con la dirección académica se recomienda escalar la presente aplicación a la evaluación de infraestructura y ayudar a la mejora de toma de decisiones.

BIBLIOGRAFÍA

- [1] R. L. Arreola Rico, «Formación y evaluación docente basada en un perfil por competencias. Una propuesta desde la práctica reflexiva», *Rev. Educ.*, vol. 43, n.o 2, p. 30, may 2019, doi: 10.15517/revedu.v43i2.30898.
- [2] J. L. Rodríguez Siu, R. E. Rodríguez Salazar, L. Fuerte Montaña, J. L. Rodríguez Siu, R. E. Rodríguez Salazar, y L. Fuerte Montaña, «Habilidades blandas y el desempeño docente en el nivel superior de la educación», *Propósitos Represent.*, vol. 9, n.o 1, ene. 2021, doi: 10.20511/pyr2021.v9n1.1038.
- [3] Presidencia de la República, «Ley Orgánica de Educación Superior». 2 de agosto de 2018. [En línea]. Disponible en: <https://www.ces.gob.ec/documentos/Normativa/LOES.pdf>
- [4] H. C. Universidad Técnica de Ambato, «0335-CU-P-2022», mar. 2022, Accedido: 30 de mayo de 2022. [En línea]. Disponible en: <https://redi.uta.edu.ec/jspui/handle/123456789/93345>
- [5] T. M. Naranjo Gamboa, «Sistema Web para el seguimiento del cumplimiento de los contenidos de los sílabos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial», ago. 2016, Accedido: 23 de mayo de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/23657>
- [6] S. A. M. Durán, X. G. C. Avila, J. P. Z. Proaño, y L. M. V. Jaramillo, «Desarrollo de un sistema web de retroalimentación continua para los docentes de la ESFOT.», p. 44.
- [7] Y. A. Olarte-Arias, C. N. Madiedo-Clavijo, y A. E. Pinilla-Roa, «Evaluación docente como factor de desarrollo profesional desde una pedagogía reflexiva», *Rev. Fac. Med.*, vol. 67, n.o 3, pp. 277-285, jul. 2019, doi: 10.15446/revfacmed.v67n3.62539.
- [8] L. F. Gómez López y M. G. Valdés, «La evaluación del desempeño docente en la educación superior», *Propósitos Represent.*, vol. 7, n.o 2, abr. 2019, doi: 10.20511/pyr2019.v7n2.255.

- [9] E. Gálvez Suarez y R. Milla Toro, «Evaluación del desempeño docente: Preparación para el aprendizaje de los estudiantes en el Marco de Buen Desempeño Docente», *Propósitos Represent.*, vol. 6, n.o 2, p. 407, ago. 2018, doi: 10.20511/pyr2018.v6n2.236.
- [10] «Bases de datos no relacionales | Bases de datos de gráficos | AWS», Amazon Web Services, Inc. <https://aws.amazon.com/es/nosql/> (accedido 20 de junio de 2022).
- [11] «¿Qué Es MongoDB?», MongoDB. <https://www.mongodb.com/es/what-is-mongodb> (accedido 20 de junio de 2022).
- [12] «ECMAScript® 2023 Language Specification». <https://tc39.es/ecma262/> (accedido 30 de octubre de 2022).
- [13] «JavaScript - Aprende sobre desarrollo web | MDN». <https://developer.mozilla.org/es/docs/Learn/JavaScript> (accedido 30 de octubre de 2022).
- [14] «Introducción a Express/Node - Aprende sobre desarrollo web | MDN». https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction (accedido 20 de junio de 2022).
- [15] «React – Una biblioteca de JavaScript para construir interfaces de usuario». <https://es.reactjs.org/> (accedido 20 de junio de 2022).
- [16] «Stack Overflow Developer Survey 2022», Stack Overflow. https://survey.stackoverflow.co/2022/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2022 (accedido 22 de agosto de 2022).
- [17] A. M. Potdar, N. D g, S. Kengond, y M. M. Mulla, «Performance Evaluation of Docker Container and Virtual Machine», *Procedia Comput. Sci.*, vol. 171, pp. 1419-1428, ene. 2020, doi: 10.1016/j.procs.2020.04.152.
- [18] Y. Baiskar, «MERN: A Full-Stack Development», *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, n.o 1, pp. 1029-1035, ene. 2022, doi: 10.22214/ijraset.2022.39982.

- [19] L. Filippova y R. Svidelskyi, «Research of key approaches to responsive website development and their practical application», *Technol. Audit Prod. Reserv.*, vol. 6, pp. 31-35, nov. 2016, doi: 10.15587/2312-8372.2016.86086.
- [20] «Estadísticas del Número de Estudiantes». <https://servicios.uta.edu.ec/SistemaIntegrado/Estadisticas> (accedido 6 de marzo de 2023).
- [21] O. Popova, «Adaptation of flexible project management models based on Scrum and Kanban technologies», *Technol. Audit Prod. Reserv.*, vol. 4, n.o 2(48), pp. 4-10, jul. 2019, doi: 10.15587/2312-8372.2019.180459.
- [22] C. E. S. de F. Formación, «13o informe anual del estado de los Métodos Agile», CESI, 3 de enero de 2020. <https://www.cesi-iberia.com/workshop-introduccion-al-scrum-13o-informe-anual-del-estado-de-los-metodos-agile/> (accedido 7 de noviembre de 2022).
- [23] L. Alvarado, «Ingeniería del Software. Un Enfoque Práctico 7ma edición», Accedido: 7 de noviembre de 2022. [En línea]. Disponible en: https://www.academia.edu/24308956/Ingenieria_del_Software_Un_Enfoque_Practico_7ma_edici%C3%B3n
- [24] Y. Sugimori, K. Kusunoki, F. Cho, y S. Uchikawa, «Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system», *Int. J. Prod. Res.*, vol. 15, n.o 6, pp. 553-564, ene. 1977, doi: 10.1080/00207547708943149.
- [25] Martín Darío Arango Serna, undefined, undefined, M. D. A. Serna, L. F. C. Zapata, y J. A. Z. Cortes, «Mejoramiento de procesos de manufactura utilizando Kanban», *Rev. Ing. Univ. Medellín*, vol. 14, n.o 27, Art. n.o 27, 2015, doi: 10.22395/rium.v14n27a13.
- [26] C. Avci, B. Tekinerdogan, y I. N. Athanasiadis, «Software architectures for big data: a systematic literature review», *Big Data Anal.*, vol. 5, n.o 1, p. 5, dic. 2020, doi: 10.1186/s41044-020-00045-1.

- [27] «¿Qué es un contenedor? | Microsoft Azure». <https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-a-container> (accedido 8 de noviembre de 2022).
- [28] «Los contenedores en comparación con las máquinas virtuales». <https://www.redhat.com/es/topics/containers/containers-vs-vms> (accedido 8 de noviembre de 2022).
- [29] R. Wang y Z. Yang, «SQL vs NoSQL: A Performance Comparison», p. 3.
- [30] «Replication — MongoDB Manual». <https://www.mongodb.com/docs/manual/replication/> (accedido 17 de noviembre de 2022).
- [31] «mongoose», npm. <https://www.npmjs.com/package/mongoose> (accedido 21 de noviembre de 2022).
- [32] «morgan - npm». <https://www.npmjs.com/package/morgan> (accedido 21 de noviembre de 2022).
- [33] «helmet», npm. <https://www.npmjs.com/package/helmet> (accedido 21 de noviembre de 2022).
- [34] «@hapi/boom», npm. <https://www.npmjs.com/package/@hapi/boom> (accedido 21 de noviembre de 2022).
- [35] «bcrypt», npm. <https://www.npmjs.com/package/bcrypt> (accedido 21 de noviembre de 2022).
- [36] «jwt-node», npm. <https://www.npmjs.com/package/jwt-node> (accedido 21 de noviembre de 2022).
- [37] «joi», npm. <https://www.npmjs.com/package/joi> (accedido 21 de noviembre de 2022).

[38] «dotenv», npm. <https://www.npmjs.com/package/dotenv> (accedido 21 de noviembre de 2022).

[39] «Why Use React Redux? | React Redux». <https://react-redux.js.org/introduction/why-use-react-redux> (accedido 28 de noviembre de 2022).

ANEXO 1.- ENCUESTA ESTUDIANTES

UNIVERSIDAD TÉCNICA DE AMBATO

Facultad de Ingeniería en Sistemas Electrónica e Industrial

Carrera de Tecnologías de la Información

Ficha de Observación del Sistema de Evaluación Docente

Autor: Célimo David Constante Prócel

Fecha: 20/10/2022

Objetivo: Recolectar opiniones por parte de estudiantes referente al Sistema de Evaluación Docente de la Universidad Técnica de Ambato.

Pregunta 1:

¿Está familiarizado con el código de ética de la Universidad Técnica de Ambato?

- Sí
- No
- Un Poco

Pregunta 2:

¿Sabía usted que el código de ética también le exige cierto comportamiento dentro de la Universidad a sus docentes?

- Sí
- No

Pregunta3:

¿Está usted familiarizado con el sistema de Evaluación docente?

- Sí
- No
- Un poco

Pregunta 4:

¿Le parece cómodo el acceso al sistema de evaluación docente?

- Sí
- No
- Un poco

Pregunta 5:

¿Ha perdido alguna materia por culpa de la metodología de enseñanza de un docente?

- Sí
- No

Pregunta 6:

¿Sus docentes adaptan la metodología de enseñanza a medida que avanza el periodo académico?

- Sí
- No
- Un poco

Pregunta 7:

¿Qué tan flexible son sus docentes con la metodología de enseñanza?

- 1 – Muy poco Flexibles
- 2
- 3
- 4
- 5 – Muy Flexibles

Pregunta 8:

¿Su docente es transparente con las calificaciones?

- Sí
- No
- Un poco

Pregunta 9:

¿Alguna vez ha sentido que sus docentes tienen cierto favoritismo hacia algunos estudiantes?

- Sí
- No
- Un poco

Pregunta 10:

¿Se siente satisfecho con el desempeño de sus docentes durante el periodo académico?

- Sí

- No
- Un poco

Pregunta 11:

¿En qué medida siente que la metodología de enseñanza de sus docentes afecta su salud mental?

- 1 – Poco
- 2
- 3
- 4
- 5 – Mucho

ANEXO 2.- FICHA DE OBSERVACIÓN

UNIVERSIDAD TÉCNICA DE AMBATO

Facultad de Ingeniería en Sistemas Electrónica e Industrial

Carrera de Tecnologías de la Información

Ficha de Observación del Sistema de Evaluación Docente

Autor: Célimo David Constante Prócel

Fecha: 24/10/2022

Objetivo: Realizar un análisis desde la perspectiva de un Estudiante al Sistema de Evaluación Docente de la Universidad Técnica de Ambato para verificar su funcionamiento.

Escala de Valoración:

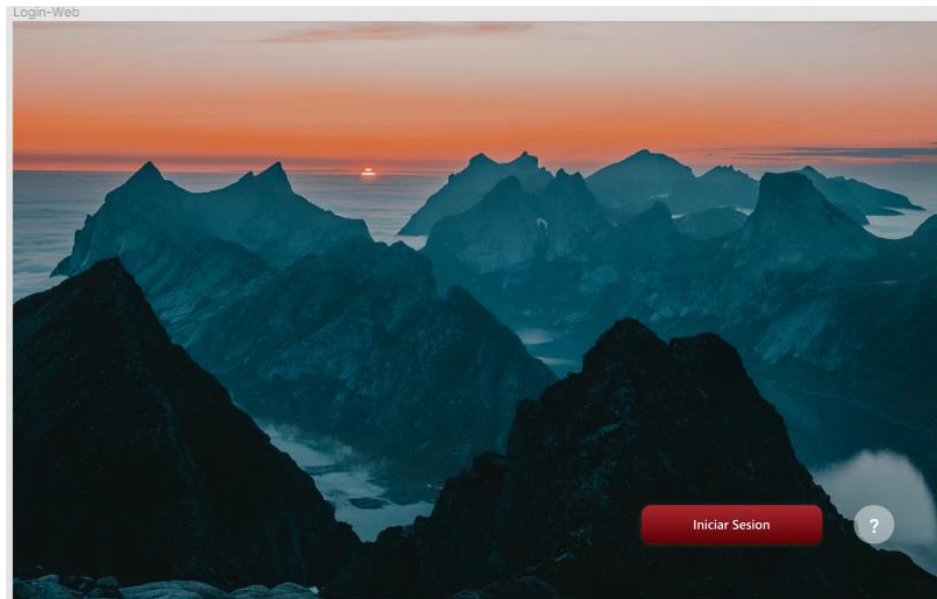
5: Excelente, 4: Muy Bueno, 3: Bueno, 2: Regular, 1: Insuficiente

N	DESCRIPCIÓN	1	2	3	4	5	OBSERVACIÓN
1	Facilidad de Acceso al sistema						
2	Certificados se seguridad						
3	Renderizado personalizado de los elementos						
4	Errores de consola						

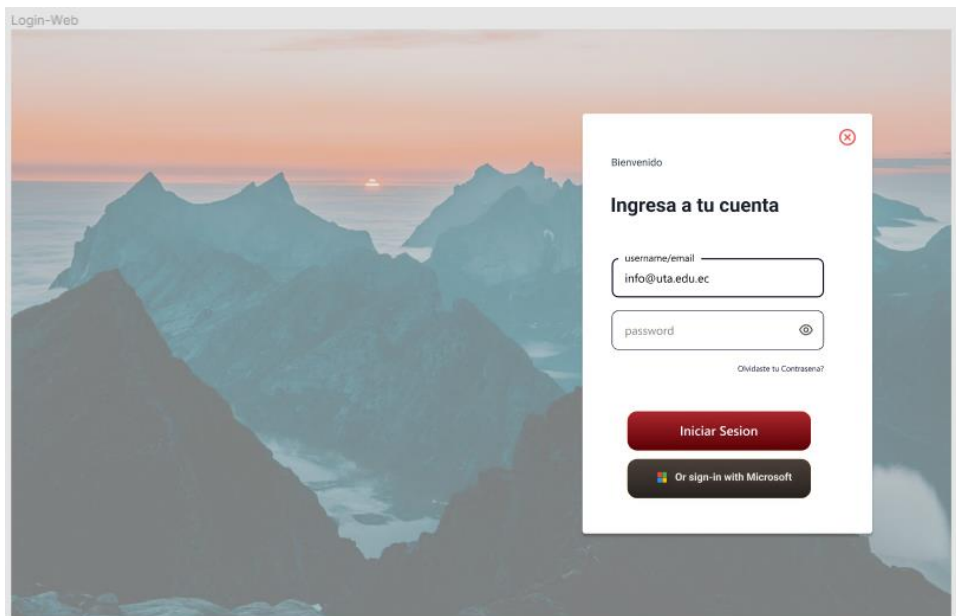
5	Actualización del Sistema						
6	Sistema Responsive						
7	Efectividad del Sistema						
8	Líneas de diseño						

ANEXO 3.- MOCKUP WEB

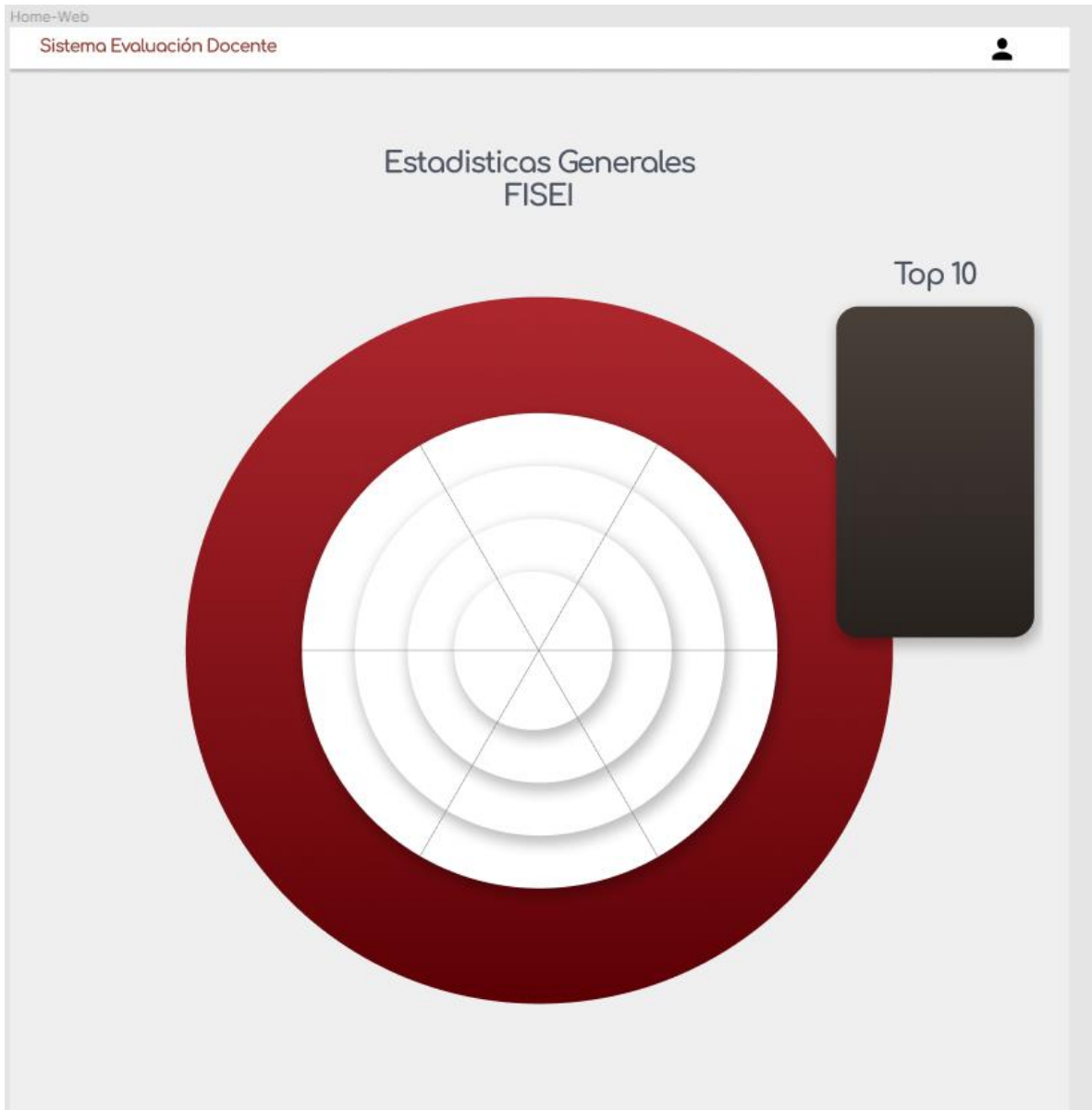
Ventana de bienvenida




Formulario de Acceso




Hero, estadísticas generales y top 10




Tarjetas docentes – footer




Vicky Russ
CTO FOR APPLICATION
DEVELOPMENT
[View profile](#)




Rot Mitt
CTO FOR COMMUNICATIONS
NETWORKS
[View profile](#)




Laura Hanz
CEO
[View profile](#)




Elena Wood
DEPUTY CHIEF EXECUTIVE
OFFICER
[View profile](#)




Vicky Russ
CTO FOR APPLICATION
DEVELOPMENT
[View profile](#)




Rot Mitt
CTO FOR COMMUNICATIONS
NETWORKS
[View profile](#)




Laura Hanz
CEO
[View profile](#)




Elena Wood
DEPUTY CHIEF EXECUTIVE
OFFICER
[View profile](#)




Vicky Russ
CTO FOR APPLICATION
DEVELOPMENT
[View profile](#)




Rot Mitt
CTO FOR COMMUNICATIONS
NETWORKS
[View profile](#)




Laura Hanz
CEO
[View profile](#)




Elena Wood
DEPUTY CHIEF EXECUTIVE
OFFICER
[View profile](#)




Vicky Russ
CTO FOR APPLICATION
DEVELOPMENT
[View profile](#)



Rot Mitt
CTO FOR COMMUNICATIONS
NETWORKS
[View profile](#)




Laura Hanz
CEO
[View profile](#)



Elena Wood
DEPUTY CHIEF EXECUTIVE
OFFICER
[View profile](#)

Sistema Evaluación Docente

Corporis ut alliquid qui molestiae. Sed laborum sunt recusandae animi dolore voluptates.



©2020 All rights reserved

Product

[Download](#)
[Pricing](#)
[Locations](#)
[Server](#)
[Countries](#)
[Blog](#)

Company

[Brand](#)
[FAQ](#)
[Tutorials](#)
[About Us](#)
[We're hiring!](#)

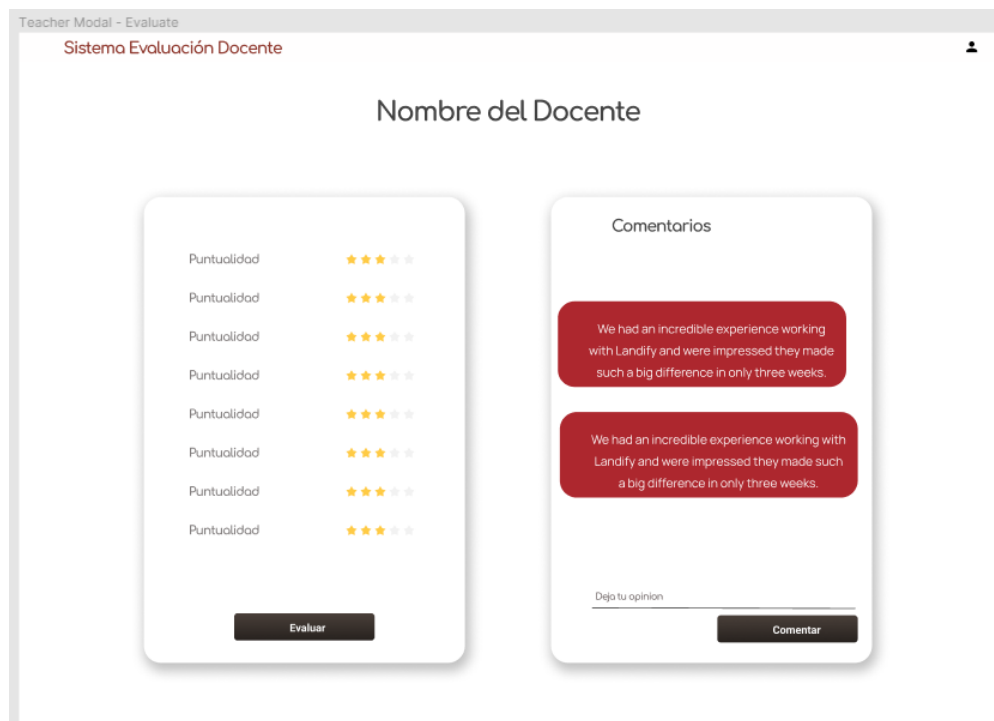
Privacy

[Privacy](#)
[Terms of Use](#)
[Site Map](#)

Ventana docente – estadísticas

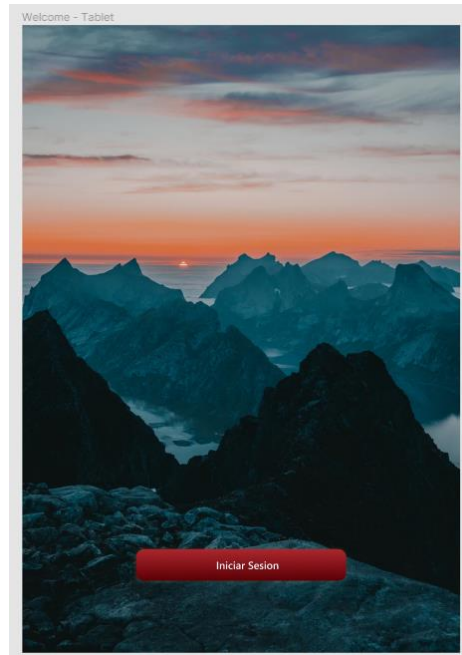


Ventana docente – evaluación

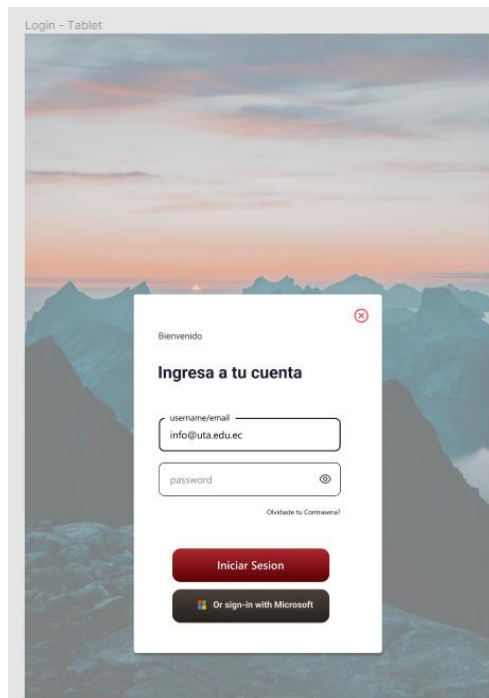


ANEXO 4.- MOCKUP TABLET

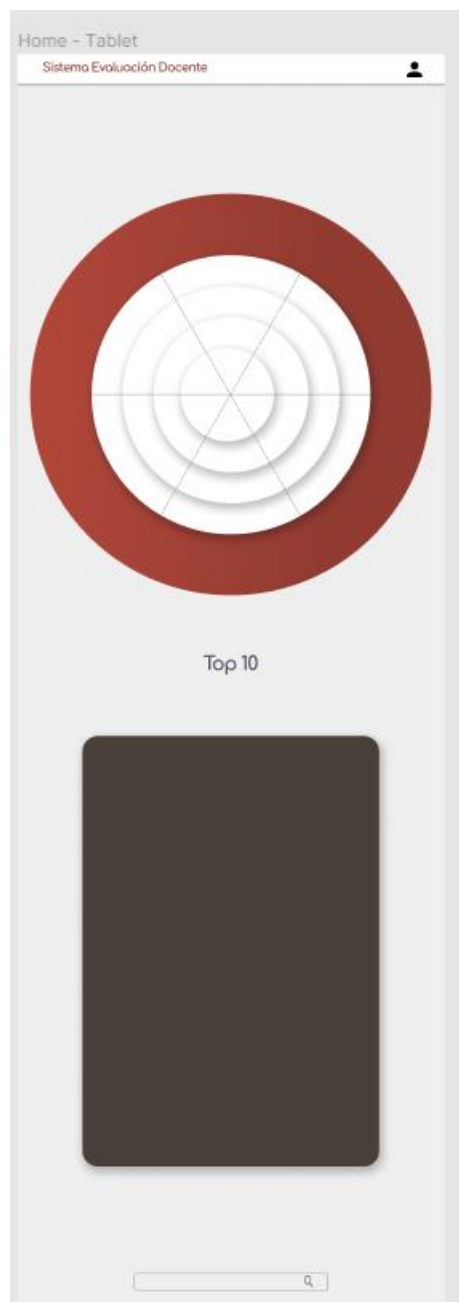
Ventana de bienvenida



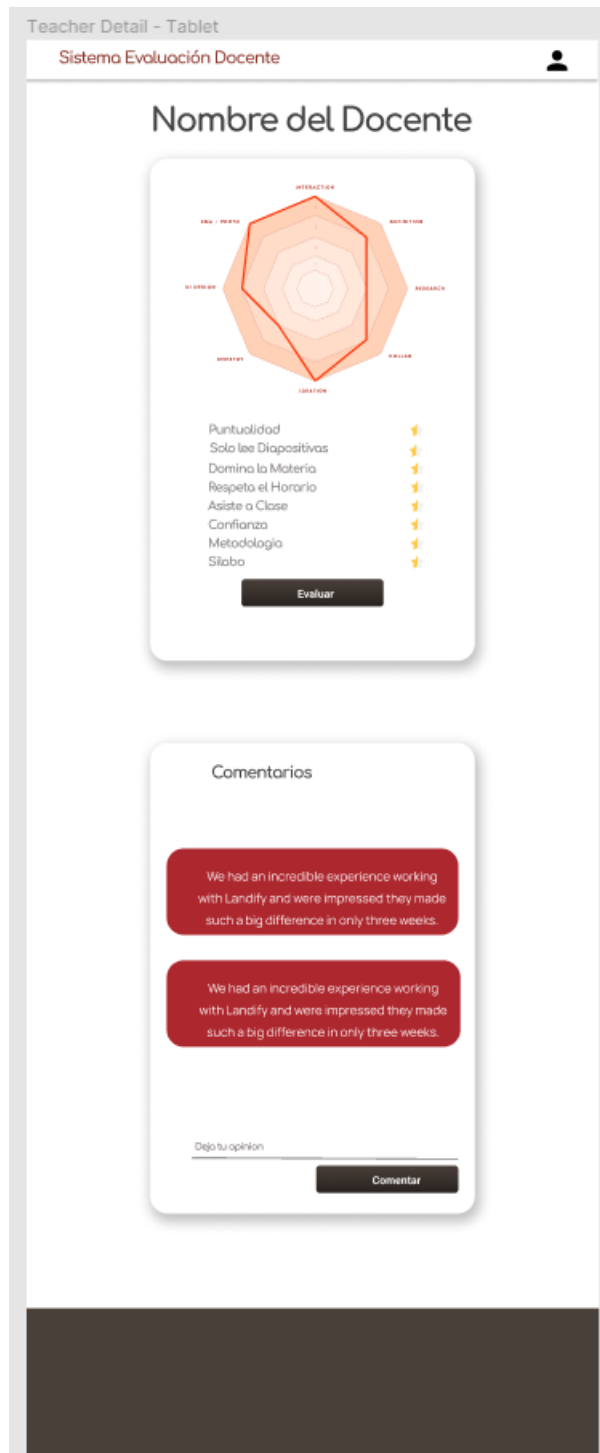
Formulario de acceso



Hero, estadísticas generales y top 10



Ventana docente – estadísticas



ANEXO 5.- MOCKUP PHONE

Ventana de bienvenida



Formulario de acceso

Login - Mobile

Bienvenido

Ingresar a tu cuenta

username/email

info@uta.edu.ec

password

Iniciar Sesión

 Or sign-in with Microsoft

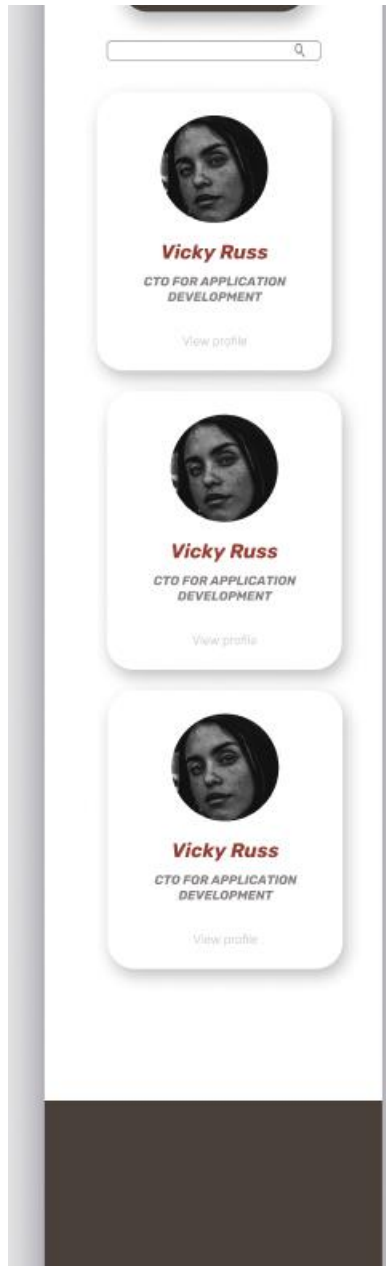
[Olvidaste tu Contraseña?](#)



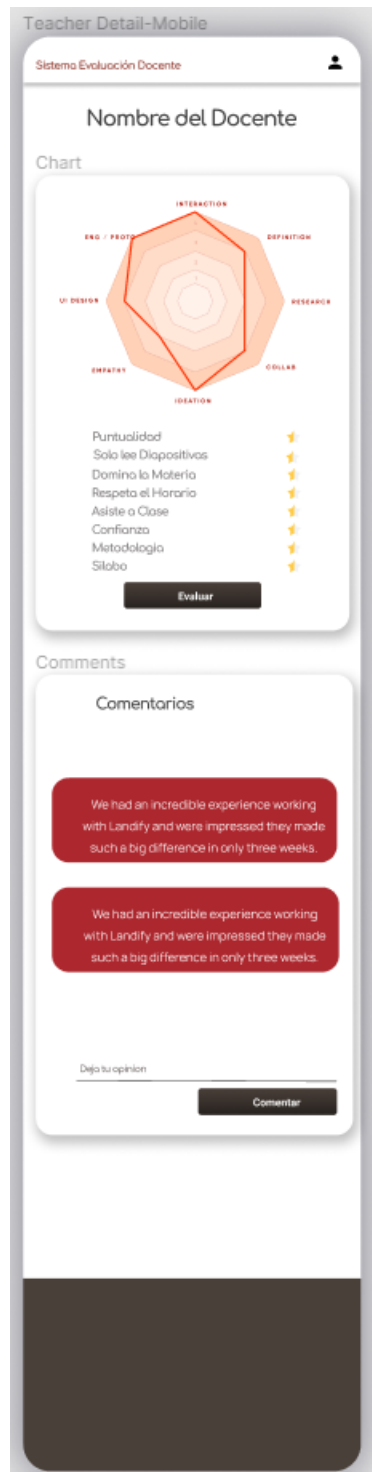
Hero, estadísticas generales y top 10



Tarjetas docentes – footer




Ventana docente – estadísticas





Ventana docente – evaluación


Teacher Detail-Mobile


Sistema Evaluación Docente 


Nombre del Docente


Puntuabilidad 


Puntuabilidad 


Puntuabilidad 

Puntuabilidad 

Puntuabilidad 

Puntuabilidad 

Puntuabilidad 

Puntuabilidad 

Evaluar

Comentarios

We had an incredible experience working with Landify and were impressed they made such a big difference in only three weeks.

We had an incredible experience working with Landify and were impressed they made such a big difference in only three weeks.

Deja tu opinion

Comentar

ANEXO 6.- MANUAL DE USUARIO

Manual de Usuario Estudiantes

<https://app.tango.us/app/workflow/Manual-de-Usuario-Estudiantes-7fd53f5fe08d4e96bcc3f8740e610db2>

Manual de Usuario Coordinador

<https://app.tango.us/app/workflow/Manual-de-Usuario-Coordinador-c1e27fa65b18498db56fca998d96a529>

Manual de Usuario Administrador

<https://app.tango.us/app/workflow/Manual-Administradores-d2cca071072445eb8de444c5a310d856>