

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA WEB QUE SIRVA COMO GUÍA DE ESTUDIO PARA ESTUDIANTES DE LA ESFOT

COMPONENTE FRONTEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

STEVEN ADRIAN CHICAIZA UGSHA

DIRECTORA: Mayra Alvarez Jiménez

DMQ, febrero 2023

CERTIFICACIONES

Yo, Steven Adrian Chicaiza Ugsha declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Steven Adrian Chicaiza Ugsha

steven.chicaiza@epn.edu.ec

adrianchicaiza3@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Steven Adrian Chicaiza Ugsha, bajo mi supervisión.



Ing. Mayra Alvarez Jiménez MSc.

DIRECTORA

mayra.alvarez@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



Steven Adrian Chicaiza Ugsha

DEDICATORIA

Dedico el actual proyecto a mi familia los cuales me ayudaron durante mi ciclo universitario dándome apoyo y motivándome para que siga adelante, también dedicó este trabajo a todos mis profesores y tutores que me ayudaron a llegar hasta aquí los cuales con su dedicación y paciencia han hecho que mi formación académica sea la mejor.

AGRADECIMIENTO

Doy gracias a mis padres por su apoyo incondicional y siempre estar a mi lado para motivarme hacer siempre el mejor.

También agradezco a la EPN por permitirme formar parte de esta gran familia en dónde compañeros y profesores me han enseñado que hay que esforzarse para conseguir nuestros sueños. También a los educadores de la Escuela de formación de tecnólogos que con su gran carisma, paciencia, dedicación y conocimientos han hecho que mi etapa universitaria sea un gran recuerdo cuando me desenvuelva en mi vida profesional.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
Objetivo general	2
Objetivos específicos	2
Alcance2	
Marco Teórico	3
1. METODOLOGÍA	8
Metodología de Desarrollo	8
Roles.....	8
Artefactos.....	9
Diseño de interfaces (mockups).....	12
Herramienta utilizada para el diseño.....	12
Diseño de la arquitectura	13
Patrón arquitectónico	13
Herramientas de desarrollo	14
Librerías.....	15
3. RESULTADOS.....	16
Sprint 0. Configuración del ambiente de desarrollo	16
Sprint 1. Inicio de sesión y recuperación de contraseña del usuario.....	18
Sprint 2. Registrar usuario y editar perfil de usuario	21
Sprint 3. Gestionar carreras y visualizar información	24
Sprint 4. Gestión y visualización de materiales educativos	27
Sprint 5. Búsqueda de materias y gestión de inquietudes.....	32
Sprint 6. Pruebas del sistema y despliegue.....	36
CONCLUSIONES.....	40
RECOMENDACIONES	41

REFERENCIAS BIBLIOGRÁFICAS	42
ANEXOS	46
ANEXO I.....	47
ANEXO II.....	48
ANEXO III.....	84
ANEXO IV	85

RESUMEN

Actualmente, en la comunidad politécnica los estudiantes no cuentan con una guía de estudios con contenidos de las materias de las carreras ofertadas por la ESFOT. Por tanto, como trabajo de Integración Curricular se propone el desarrollo de un sistema web que fomente la autoeducación, poniendo a disposición de los estudiantes material de estudio para comprender mejor los temas del PEA de las carreras de la ESFOT. El principal propósito de "EDUPOLI" es asistir a los estudiantes encontrando los temas de estudio de las distintas asignaturas de forma fácil, además de proporcionarles documentos de los temas tratados en cada semestre acorde a la carrera. Además, de proporcionar a los usuarios una forma de comunicarse a través de foros, fomentando la autoeducación en estudiantes tanto de nuevo ingreso como los ya en curso. Cabe mencionar que el actual proyecto se centra en el desarrollo del componente *frontend* EDUPOLI, el cual presenta una interfaz limpia y agradable para los estudiantes al momento de ingresar al sistema, de manera que encuentren los temas de cada asignatura que buscan.

La estructura del presente documento se divide en diferentes secciones como son: La sección I describe la problemática del proyecto, objetivos, alcance y marco teórico. En la sección II, se describe la metodología ágil *Scrum*, diseño de la arquitectura, interfaces y herramientas utilizadas para el componente *Frontend*. En la sección III, se describen los resultados que se obtienen de cada *Sprint*. Por último, se presentan conclusiones y recomendaciones que se han obtenido al desarrollar el componente *frontend*.

PALABRAS CLAVE: *Frontend*, Autoeducación, Interfaz, *Scrum*, ESFOT

ABSTRACT

Currently, in the polytechnic community, students do not have a study guide with content of the subjects of the careers offered by the ESFOT. Therefore, as Curricular Integration work, the development of a web system that promotes self-education is proposed, making study material available to students to better understand the PEA topics of the ESFOT careers. The main purpose of "EDUPOLI" is to assist students finding the study topics of the different subjects in an easy way, in addition to providing them with documents of the topics covered in each semester according to the career. In addition, to provide users with a way to communicate through forums, promoting self-education in both new students and those already in progress. It is worth mentioning that the current project focuses on the development of the EDUPOLI frontend component, which presents a clean and pleasant interface for students when moment of entering the system, so that they find the topics of each subject they are looking for.

The structure of this document is divided into different sections such as: Section I describes the problems of the project, objectives, scope and theoretical framework. In section II, the agile Scrum methodology, architecture design, interfaces and tools used for the Frontend component are described. In section III, the results obtained from each Sprint are described. Finally, conclusions and recommendations that have been obtained when developing the frontend component are presented.

KEYWORDS: Frontend, Self-education, Interface, Scrum, ESFOT

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Por el momento, el aprendizaje ha avanzado de manera que las personas ya no necesitan una biblioteca para encontrar temas de su interés, hoy en día cualquier información se puede encontrar más rápido sin importar la ubicación geográfica a través de Internet.

El internet ha permitido fomentar el aprendizaje autodidacta o la autoeducación, que no es más que la posibilidad de determinar nuestro propio ritmo de aprendizaje, por lo tanto, podemos acceder a información a cualquier hora y en cualquier lugar, solo necesitamos de una conexión a internet y un dispositivo electrónico [1].

En Ecuador, a raíz del brote del COVID-19, hizo que las diferentes instituciones educativas presenten alternativas al momento de educar a los estudiantes, de esta manera se presentaron plataformas las cuales podrían ayudar a fomentar la autoeducación en los estudiantes, aunque no con resultados satisfactorios, pues la mayoría de las instituciones se enfocaron en el envío de tareas y no en la autoeducación de los estudiantes [2].

De acuerdo a una sondeo online dirigido a 7678 estudiantes de distintos colegios en el territorio ecuatoriano matriculados en el año lectivo 2019 y 2020, el 67% de estudiantes no estaban conformes con la planificación para las clases, el 12% de escuelas y colegios se basaron únicamente al envío de tareas, el 42% de estudiantes indico que debe mejorar la organización de las clases virtuales, ya que más de la mitad de los encuestados manifestó que las clases virtuales no han sido claras, como conclusión obtenemos que las clases virtuales no se han planificado con una buena organización y mucho menos con una buena calidad, dando como resultado que los estudiantes se interesen menos por sus estudios [3]. De la misma manera, en el ámbito universitario debemos estar conscientes de la importancia de la autoeducación, pues es fundamental para la realización académica y profesional de los estudiantes que salen al mundo laboral una vez culminado sus estudios universitarios [4].

La autoeducación es una parte esencial de todos los estudiantes, incluyendo los universitarios, ya que fomenta el proceso de formación profesional y formación de aptitudes como el autoaprendizaje, a desarrollar su pensamiento crítico y ético sobre problemáticas a los que se enfrentan los estudiantes en la actualidad, habilidades para relacionarse adecuadamente en su entorno laboral [4].

Podemos hablar de un ejemplo de un sistema de autoeducación como es “MicroCosmos”, la cual es una herramienta tecnológica que se desarrolló para la autoeducación de niños, niñas y jóvenes mediante el uso de una aplicación móvil que promueve su curiosidad por

la ciencia, tecnología y el medio ambiente, este tipo de herramientas sirven para fomentar el aprendizaje autodidacta de manera que los estudiantes se sientan atraídos hacia el aprendizaje sobre un tema que les guste [5].

Es esencial implementar un sistema web que apoye y promueva el aprendizaje autodidacta de los estudiantes en Ecuador para mejorar la educación. En la Escuela de Formación de Tecnólogos de la comunidad Politécnica, este tema de integración curricular se enfoca en fomentar la autoeducación de los estudiantes, proporcionando temas y ejercicios de diferentes asignaturas de cada semestre de las carreras que ofrece la ESFOT. Por lo tanto, es importante que tanto los estudiantes nuevos como los que ya están cursando tengan acceso a un sitio web que les permita revisar los temas de sus asignaturas o adelantarse a temas que revisaran en futuras clases. Por tal razón se ha implementado un sistema web amigable e intuitivo, que pueda ser utilizado para buscar información y facilitar el aprendizaje autodidacta de los estudiantes en EDUPOLI, sería una solución efectiva.

Objetivo general

Desarrollar un componente *frontend* de manera que sea una interfaz llamativa para los estudiantes al momento de buscar los temas de estudio dentro del sistema.

Objetivos específicos

- Levantar requerimientos para el frontend del sistema con el fin de establecer los métodos dentro del sistema.
- Diseñar el componente *frontend* en función a los requerimientos adquiridos para la mejor visualización del contenido por parte de los estudiantes.
- Codificar los módulos del componente *frontend* en función a los requerimientos adquiridos para que los estudiantes realicen ciertas acciones dependiendo del módulo.
- Realizar pruebas del *frontend* y realizar el respectivo despliegue para el uso de los estudiantes.

Alcance

El presente proyecto trata de la realización del componente *frontend* la cual facilita al usuario final a realizar acciones dentro del sistema de manera más sencilla, cómoda y

contando con una interfaz amigable, esto con la ayuda del desarrollo *web* y con herramientas que facilita la implementación de este componente, el cual su objetivo principal es la de mejorar la experiencia del usuario.

El componente *frontend* ha asegurado que funcione correctamente, se adapte, integre y comunique adecuadamente con el *backend*. Diversas funcionalidades se han considerado para hacer uso del componente *frontend* más fácil, con interfaces que muestran información en tiempo real y compatibles con diversos navegadores web.

La gestión de información se ha simplificado mediante diversas funcionalidades, que incluyen detalles sobre carreras, asignaturas y material educativo. Herramientas de desarrollo del lado del cliente se han empleado para alcanzar estos objetivos, y se ha llevado a cabo una buena organización del código y una metodología de desarrollo para garantizar el cumplimiento de los requisitos del proyecto. Además, se haber realizado pruebas para asegurar el correcto funcionamiento del software.[6]. Por último, el acceso a la información dentro del sistema se basa en el perfil del usuario estos son: Administrador y Estudiante, ambos perfiles consumen métodos para mostrar distinta información en el sistema *web*.

Usuario administrador

- Consumir métodos para el *login*, recuperar contraseña y cerrar sesión.
- Consumir métodos para modificar la información del usuario.
- Consumir métodos para el gestionamiento de carreras, semestres, asignaturas y material educativo.

Usuario estudiante

- Consumir métodos para el *login*, recuperar contraseña y cerrar sesión.
- Consumir métodos para modificar la información del usuario.
- Consumir métodos para la visualización del material educativo.
- Consumir métodos para la gestión de sugerencias a través de foros.

Marco Teórico

En esta sección se da a conocer bases teóricas referente a temas tratados para la realización del proyecto, por tanto, se explicará que son cada uno de los temas a tratar como son las metodologías, componentes y herramientas a utilizar para su desarrollo.

Aprendizaje autónomo

Actualmente se ha planteado el desarrollo de competencias de aprendizaje autónomo, donde se espera que los estudiantes aprendan por sí mismos. Sin embargo, en los cursos presenciales no se ha logrado desarrollar este tipo de aprendizaje de manera adecuada debido a varios factores. Por ejemplo, los estudiantes suelen dejar la responsabilidad de su aprendizaje a los profesores y no suelen darle importancia al estudio autónomo. [7].

Metodologías ágiles

Las metodologías ágiles se enfocan en proporcionar pequeñas piezas de software en poco tiempo para aumentar la satisfacción del cliente. Para lograr esto, se utilizaron varios métodos para ofrecer mejoras constantes. La metodología ágil promovió que los procesos de desarrollo no fueran en cascada, sino que permitió que varias secuencias sucedieran al mismo tiempo. De esta manera, este tipo de metodología valoró las interacciones con las personas, el funcionamiento del software, la colaboración con el cliente y la respuesta ante el cambio [8]. De esta manera se presenta la **TABLA I**, con las metodologías ágiles más utilizadas:

TABLA I: Metodologías Ágiles

Metodologías	Descripción
<i>Extreme Programming XP</i>	Herramienta la cual fomenta la relación entre las personas involucradas en un desarrollo de proyecto, de manera que se planifican los pasos y estructuras de acuerdo con las peticiones del usuario [8]. A pesar de que es útil para proyectos a corto plazo, si hay errores o fallas, los costos de las comisiones para resolverlos son elevados. Además, requiere una estricta adhesión a los principios de XP.
<i>SCRUM</i>	Metodología basada en la elaboración de estrategias para desarrollar <i>software</i> de la mejor manera posible, asegurando la innovación, flexibilidad, competitividad y la productividad. Además de que se basa en reuniones para establecer cada etapa a desarrollar en el ciclo de vida del <i>software</i> , por tanto, utilizar la metodología <i>SCRUM</i> es recomendable para proyectos complejos o para aquellos en los que se quieran asegurar la flexibilidad y la rapidez al momento de desarrollar el <i>software</i> [9].

	De esta manera, se logra simplificar la planificación del proyecto, estableciendo fechas límites claras y permitiendo que el equipo de desarrollo aprenda rápidamente. También logra obtener un producto mínimo viable que se puede utilizar para demostrar el progreso del proyecto.
<i>Kanban</i>	Basado en la elaboración de una tarjeta visual en donde se planifican los pasos a realizar para facilitar y agilizar el desarrollo del software, por tanto, utiliza métricas visuales y establece plazos de entregas continuos [9]. Aunque los costos para la realización con esta metodología son altos y son limitados, por tanto kanba no es óptimo para todo tipo de proyectos, ya que no nos permite anticiparnos a cambios a futuro del proyecto.

La metodología Scrum posibilita la implicación del cliente en el proyecto, ya que puede seguir el progreso en cada una de las iteraciones, permitiendo que introduzca cambios en funcionalidades o prioridades en cada interacción sin dificultad. Como resultado, los beneficios de la metodología Scrum son numerosos, incluyendo el cumplimiento de las expectativas del usuario, la flexibilidad en los cambios, la mejora en la calidad del software, el aumento de la productividad, la capacidad de prever tiempos, entre otros [10].

Frontend

El desarrollo del *frontend* es la parte que el usuario ve por primera vez al ingresar a un sitio *web*, esto incluye la experiencia de usuario y las vistas del sistema *web*, de manera que requiere herramientas enfocadas en el lado del cliente, el *frontend* es la capa que se encuentra por encima del *backend* ya que se conforma de componentes que pueden ser manipulados por el usuario como son menús desplegables imágenes, iconos, colores, etc. Que dan como resultado una experiencia de navegación mejorada en nuestro sistema *web* [11].

Es por ello, que la implementación de un *frontend* es necesario para la mejor comprensión de los estudiantes dentro del sitio *web*, en este caso el sistema *web* y el componente *frontend* van enfocados hacia la comunidad de la ESFOT de la Escuela Politécnica Nacional, en donde el aprendizaje autónomo se ha vuelto una parte esencial para cada uno de los estudiantes y el componente *frontend* fomenta a seguir aprendiendo de manera autónoma con la ayuda de distintos materiales educativos. Para ello la creación de este

componente se utilizan tecnologías del lado del cliente para mejorar la comunicación del usuario con los distintos elementos que se encuentran dentro del sistema *web* como son: menús, botones, tarjetas, etc. La realización de cada una de estas características será a través de aplicaciones de desarrollo y lenguajes de programación como son: *JavaScript*, *CSS*, *HTML*, *frameworks* y librerías útiles para el desarrollo *web* como es *React* [12].

Backend

El desarrollo del *backend* es la arquitectura y la lógica detrás del sistema *web* se realiza a través de distintas herramientas para la mejor comunicación con él *frontend*, de esta manera se utilizan *frameworks* como *laravel* y arquitecturas como el *APIrest* las cuales facilitan la implementación de distintos métodos para realizar ciertas acciones por el lado del cliente en el componente *frontend* [13].

Dentro del proyecto el componente *backend* se hace por medio de métodos establecidos y consumidos por aplicaciones del lado del cliente o dispositivos móviles sin importar el lenguaje o arquitectura utilizado ya que se basa en la comunicación y envío de información a través de rutas creadas por el componente *backend*. además, proporciona métodos las cuales ayudan a la gestión de las diferentes materias, semestres y materiales educativos que se presentan en el sistema *web*, de manera que toda la información proporcionada en el *frontend* pueda ser visualizada y también gestionada a través de distintas acciones que puede realizar el cliente [11].

React

React es una librería de uso libre basado en *JavaScript* la cual se utiliza para desarrollar interfaces de usuarios, utilizado para el desarrollo de componentes dentro de una aplicación *web* para facilitar la interacción entre los distintos componentes que se pueden crear con *React* y así crear interfaces de usuario más complejas, ayudándose de otras librerías para desarrollar de mejor manera nuestra aplicación *web* [14].

Prototipado

En el ámbito del prototipado *web*, hay varias alternativas disponibles como *Figma*, que es una herramienta de prototipado *web* y editor de gráficos vectoriales que se puede alojar en la *web*. Lo interesante de *Figma* es su facilidad de uso, ya que es intuitivo y sencillo de utilizar. Además, permite la edición de gráficos vectoriales para crear una simulación de nuestro sistema *web* y ofrece una serie de *plugins* que se pueden añadir para ampliar su uso. También cuenta con una funcionalidad de compartir para que un grupo de trabajo

pueda editar el diseño del sistema *web* y se pueda presentar un primer diseño que se ajuste a las necesidades del usuario. [15].

Pruebas de *software*

La etapa de pruebas de *software* ha sido un proceso importante en el que se ha verificado que el sistema o *software* cumpla con los requerimientos del usuario, previniendo errores y reduciendo los costos de desarrollo. También se ha mejorado el rendimiento del *software*. La etapa de pruebas es crucial en el desarrollo de un sistema *web* ya que asegura la calidad del *software* antes de la producción y previene problemas futuros. Las pruebas que se han realizado incluyen pruebas unitarias, pruebas de carga y pruebas de compatibilidad [16].

El despliegue de aplicaciones es una etapa común dentro del desarrollo de una aplicación, donde compartimos nuestra aplicación a distintos usuarios que se encuentren en internet, ya que en la actualidad existen diferentes sitios de alojamiento como servidores externos en los cuales podemos desplegar nuestras aplicaciones para que puedan ser utilizados en cualquier dispositivo con acceso a Internet, algunas plataformas de alojamiento son *Heroku*, *Netlify*, *Vercel*, entre otros [17].

1. METODOLOGÍA

La metodología empleada en este proyecto se basó en la recopilación de información a través de entrevistas, encuestas, entre otros, con el objetivo de obtener resultados precisos para realizar un análisis adecuado y, posteriormente, mejorar el componente *frontend*. El estudio de caso se utiliza comúnmente como metodología de investigación en una variedad de entornos, como sociales, educativos y empresariales. [18].

Es por ello que el presente proyecto maneja un estudio de caso para la recopilación de información en las historias de usuarios y los requerimientos para desarrollar de mejor manera el componente *frontend*, ya que la información recopilada se basa en el filtrado de información, ayudándonos de entrevistas y encuestas que dan como resultado que muchos de los estudiantes no tienen un incentivo para su aprendizaje autónomo o no encuentran la forma de hacerlo, dado que en internet existe mucha información errónea y a veces difícil de encontrar, por esta razón el desarrollo del componente *frontend* puede ayudar a los estudiantes de la ESFOT a que encuentren toda la información requerida de sus respectivas carreras de una manera rápida e intuitiva.

Metodología de Desarrollo

Las metodologías de desarrollo en el ámbito de *software* constan de métodos y técnicas que organizan a un equipo de trabajo al momento de desarrollar un *software*, de manera que evita problemas futuros y planifica de mejor manera las distintas fases en las que se desarrolla el *software*, teniendo en cuenta todos los aspectos al desarrollar un *software* como son los costes, la planificación, la dificultad o las herramientas utilizadas [19].

El uso de las metodologías ágiles permite controlar de mejor manera el ciclo de vida del *software* y así asegurar su calidad al finalizar el proyecto, por ello el desarrollo del componente *frontend* se desarrolla implementando la metodología agile *Scrum*, la cual permite controlar cada una de las interacciones planificadas antes de desarrollar el *software* de manera que el equipo de trabajo tiene un conjunto de roles los cuales tienen una función que desarrollar dentro del desarrollo del *software*, facilitando así su calidad de manera ordenada y ágil [20].

Roles

Scrum permite la incorporación de roles en donde cada uno de estos tiene una actividad a realizar dentro del desarrollo del *software*, lo cual asegura la calidad de este al finalizar el proyecto, de esta manera en el presente proyecto se presenta a continuación los distintos roles que son:

Product Owner

Este rol dirige y guía el desarrollo del *software*, de esta manera él es el responsable del producto final y el encargado de la mejora constante del producto, de manera que garantiza su rentabilidad y también de ajustar características y prioridades que tendrá cada una de las interacciones en el ciclo de vida del *software* [21]. Por consiguiente, en la **TABLA II** se presenta al encargado de este rol.

Scrum Master

Este rol se encarga de facilitar la creación del *software*, de manera que elimina dificultades en el desarrollo para poder alcanzar la fase final y así alcanzar los objetivos del equipo de trabajo, es por ello que este rol se encarga de seguir valores y practicas recomendadas de la metodología *Scrum* planificando los *Sprints*, organizando las tareas, eliminando impedimentos o brindando apoyo al grupo de desarrolladores[22]. Por consiguiente, en la **TABLA II**, se presenta al encargado de este rol.

Development Team

Este rol se encarga de cumplir los requerimientos designados por el *Product Owner*, cumpliendo con los objetivos al desarrollar el *software*, generalmente conformado por un grupo pequeño de desarrolladores los cuales cada uno de ellos sabe cuáles son sus actividades en cada una de las etapas del desarrollo [23]. Por consiguiente, en la **TABLA II**, se presenta al encargado de este rol.

Tabla II: Designación de Roles

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Mayra Alvarez
<i>Scrum Master</i>	Ing. Mayra Alvarez
<i>Development Team</i>	Adrian Chicaiza

Artefactos

Dentro de la metodología agile *Scrum* intervienen varios métodos para mejorar la calidad del *software* al finalizar su ciclo de desarrollo, de manera que garantiza el rendimiento y la eficiencia del proyecto en cada una de las interacciones que se entregarán al usuario como producto final, como resultado se presenta cada uno de estos artefactos utilizados para el desarrollo del componente *frontend*:

Recopilación de Requerimientos

La recopilación de requerimientos consiste en la documentación y definición de las necesidades que requiere el usuario final para realizar distintas acciones, ayudando de esta manera para tener en cuenta la duración y la dificultad de desarrollar el proyecto para así llevar de la mejor manera el producto final [24].

En el desarrollo del componente *frontend*, este artefacto se lleva a cabo en base a encuestas a estudiantes para así recabar información y tomar en cuenta los diferentes módulos que se implementarán en el sistema. Se presenta un ejemplo en la **TABLA III**, de la recopilación de requerimientos de distintos estudiantes a través de la encuesta realizada. No obstante, la tabla completa con todos los requerimientos se ubica en el **ANEXO II**.

TABLA III: Requerimientos que se han recopilado

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID-RR	ENUNCIADO DEL ITEM
FRONTEND	RR001	El usuario administrador y estudiante necesitan estar registrados para acceder al sistema.
	RR002	El usuario Estudiante, debe llenar correctamente todos los campos solicitados en el registro, para ingresar a los diferentes módulos permitidos para el usuario Estudian

Historias de Usuario

Son constancias de ideas o características que necesita un cliente acerca de un sistema, de manera que se lo hace desde su perspectiva para realizar ciertas acciones, por ello se responden preguntas de: como usuario que voy a realizar, quiero algún tipo de acción en el sistema y para qué quiero hacer esas acciones [25]. Por consiguiente, en la **TABLA IV**, se presenta un ejemplo de estas historias. Los demás requerimientos se encuentran en el **ANEXO II**, del presente documento.

TABLA IV – HU001 - Iniciar sesión, cerrar sesión y recuperar contraseña

HISTORIA DE USUARIO	
Identificador: HU001	Usuario: Administrador y estudiante
Nombre historia: Iniciar sesión, cerrar sesión y recuperar contraseña.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 1	
Responsable: Adrian Chicaiza	
Descripción: El administrador y estudiante pueden:	

1. Iniciar sesión
2. Cerrar sesión
3. Recuperar contraseña

Observación: El administrador y estudiante deben registrarse para acceder a los métodos mencionados.

Product Backlog

Se trata de un conjunto de tareas a realizar durante el ciclo de desarrollo del *software*, contiene todas las tareas que el equipo de desarrollo va a realizar, además de ello cuenta con un campo de iteraciones la cual representa la dificultad al desarrollar cada uno de los requerimientos establecidos en el *Product Backlog* [26]. Un ejemplo del formato del *Product Backlog* utilizado se presenta en la **TABLA V**, en donde se plasman los requerimientos solicitados, se puede revisar con más detalle en el **ANEXO II**, del actual documento.

TABLA V: Product Backlog

ELABORACION DEL PRODUCT BACKLOG				
ID-HU	HISTORIA DE USUARIO	ITERACION	ESTADO	PRIORIDAD
HU001	Iniciar sesión, cerrar sesión y recuperar contraseña.	1	Finalizado	Alta
HU002	Registrarse a través de un formulario.	2	Finalizado	Media

Sprint Backlog

Conformado de elementos o tareas que están dentro de cada una de las interacciones al momento de desarrollar el *software*, cada uno de estos organizados dependiendo de su iteración en donde el equipo de desarrollo debe completar cada una para realizar una entrega de cada uno de los requerimientos [27]. Por consiguiente, en la **TABLA VI**, muestra el formato utilizado para desarrollar el *Sprint Backlog* con distintas tareas que son necesarias antes de realizar el componente *frontend*. La tabla se muestra con más detalle en el **ANEXO II**.

TABLA VI: Sprint Backlog

ELABORACION DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MODULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo	-----	-----	-----	<ul style="list-style-type: none"> - Definir los requerimientos funcionales y no funcionales para el componente <i>frontend</i> - Configuración de herramientas - Configuración del ambiente de desarrollo - Roles de usuarios 	20 horas

Diseño de interfaces (mockups)

Un *mockup* es un esquema de un diseño inicial que va a tener nuestro sistema o aplicación web, de manera que se pueden estilizar para realizar distintas acciones que dan como resultado respuestas que son controladas por la persona, es por ello que el diseño de la interfaz debe asegurar una interacción fácil e intuitiva para la persona, ya que necesita saber si las acciones que realiza tienen algún efecto en el sistema [28].

Herramienta utilizada para el diseño

Diseñar prototipos para el competente *frontend* es fundamental pues de ello depende las vistas que requiere el usuario, por ello es aconsejable la utilización de *Figma*, la cual es una potente herramienta de prototipado en el cual se pueden visualizar los diferentes módulos que se van a implementar en el componente *frontend*. Una prueba de ello se presenta en la **Fig. 1** con el formulario de login, campos para insertar los datos del usuario, la distribución de los campos para las imágenes e iconos y por último los colores designados para el sistema. En él **ANEXO II**, del actual documento se presenta los demás prototipos diseñados.

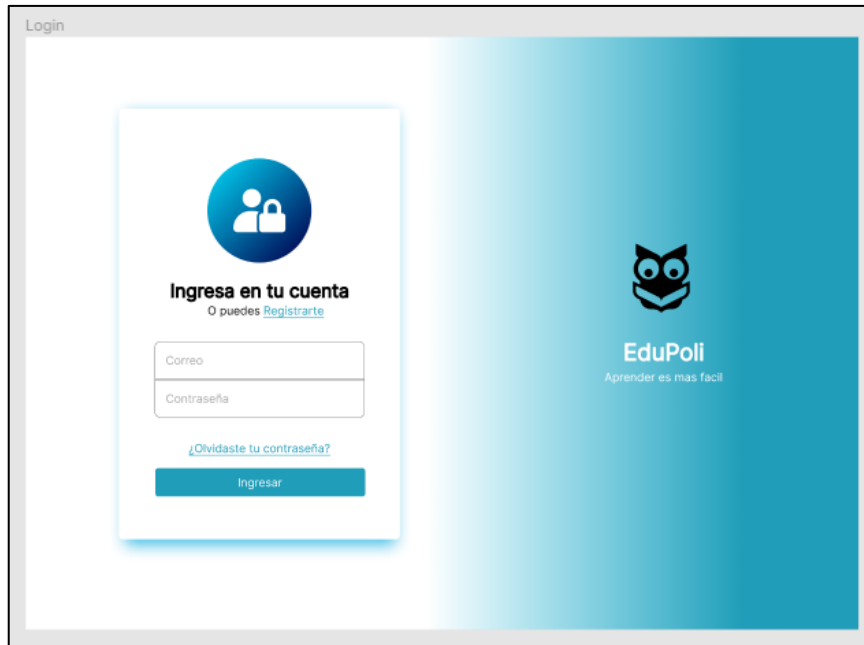


Fig. 1: Prototipo de inicio de sesión del sistema *web*

Diseño de la arquitectura

La importancia de establecer una arquitectura para nuestro *software* es uno de los pasos críticos antes de realizar un programa, puesto que de esto depende cómo se va a planificar y realizar todos los módulos que conlleva el ciclo de vida del *software*, por tanto, a la hora de elegir un patrón de arquitectura hay que tomar en cuenta el uso final que tendrá nuestro programa o sistema *web* [29].

Patrón arquitectónico

El modelo arquitectónico utilizado para el desarrollo del componente *frontend* es la del Modelo, Vista y Controlador, que consiste en la división de 3 capas con diferentes componentes que realizan tareas concretas y ofrecen beneficios diversos. La arquitectura MVC fue diseñada inicialmente para sistemas en donde se requiere el uso de interfaces, de manera que se crea un *software* robusto con prácticas recomendadas, aprovechando el código y el mantenimiento del mismo [30].

En este sentido se presenta la **Fig. 2**, con el patrón arquitectónico utilizado para el desarrollo del *frontend* basado en el MVC y la compatibilidad que existe entre las librerías y herramientas a utilizar.

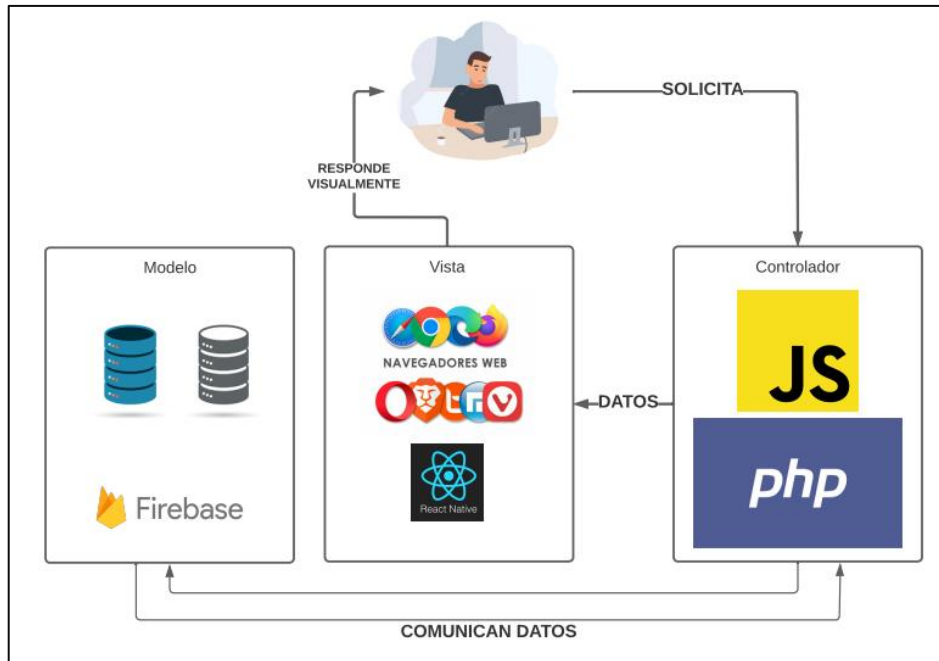


Fig. 2: MVC del Sistema Web

Herramientas de desarrollo

Existen diversas herramientas que ayudan a los desarrolladores a construir un software, en este caso desarrollar el componente *frontend*, por ello las herramientas a utilizar han sido seleccionadas en base a los requerimientos reunidos, de manera que facilita la creación de los diversos componentes y manteniendo una buena optimización para cada uno de ellos, la **TABLA VII** presenta las herramientas utilizadas :

TABLA VII: Herramientas para el desarrollo del componente *frontend*

Herramienta	Justificación
<i>React</i>	React es una librería de código abierto desarrollado por Facebook para facilitar la creación de componentes reutilizables dentro de un sistema <i>web</i> , de manera que facilita la creación de estos de una manera más fácil y elegante [31].
<i>ViteJs</i>	Se trata de una herramienta para el desarrollo <i>frontend</i> , creado por Evan You, con el fin de crear aplicaciones de <i>JavaScript</i> usando algún tipo de <i>framework</i> como <i>React</i> de manera que facilita el desarrollo de sistemas <i>webs</i> [32].
<i>Bootstrap</i>	Se trata de un <i>framework</i> CSS el cual facilita la integración de modelados y estilos para diferentes componentes como en

	<i>React</i> , por tanto, ofrece una mejor experiencia de usuario cuando navega por nuestro sitio web [33].
--	---

Librerías

Existen distintas librerías para el desarrollo del *frontend*, las cuales aportan una serie de ventajas al momento de desarrollar *software*, de manera que no es necesario establecer funciones cuando ya existen librerías que las proporcionan, ahorrando así tiempo y esfuerzo. La **TABLA VIII** muestra las librerías empleadas en el proyecto.

TABLA VIII: Herramientas para el desarrollo del componente *frontend*

LIBRERÍA	DESCRIPCIÓN
<i>react-router-dom</i>	Se trata de una librería la cual proporciona funciones para la navegación entre páginas dentro de red de manera que establecemos una carpeta de rutas y el nombre al componente al que queremos ir [34].
<i>axios</i>	Se trata de una librería la cual proporciona funciones pasadas en promesas de clientes <i>HTTP</i> , de manera que podemos insertar, consultar, borrar y ver datos extraídos de una base de datos [35].
<i>hooks-react</i>	Se trata de componentes que realizan funciones específicas y que facilitan así el desarrollo de componentes dentro de <i>React</i> , de manera que se basa en estados o acciones dependiendo de lo que importamos en los <i>Hooks</i> [36].

3. RESULTADOS

En esta sección se presentan los resultados de acuerdo con cada uno de los *Sprints*, planteados anteriormente para una mejor distribución y realización de cada uno de ellos

Sprint 0. Configuración del ambiente de desarrollo

En el Sprint 0 nos enfocamos en:

- 1 Definir requerimientos funcionales y no funcionales del *frontend*.
- 2 Configuración de herramientas.
- 3 Organización del *frontend*
- 4 Roles de usuarios.

Definir requerimientos funcionales y no funcionales del frontend.

Se definen los requerimientos para cada historia de usuario, de manera que establecemos los métodos a realizar en cada uno de los *Sprints*, un requerimiento es funcional o no dependiendo de la capacidad necesitada por nuestro usuario para que solviente un problema o lleve a cabo un objetivo, por ejemplo, los más esenciales como la gestión de carreras que son requerimientos funcionales, como las no funcionales de validación de campos en cada una de las páginas.

Configuración de herramientas

Se configuran las herramientas a utilizar dentro del desarrollo del componente *frontend*, tanto como el entorno de desarrollo como es *Visual Studio Code*, las librerías a utilizar como las dependencias de *React* y *Hooks* para la realización de los componentes que se utilizaran en las diferentes vistas del sistema, cada una de estas se configuran con sus respectivos comandos o instalaciones de sus páginas oficiales.

Organización del frontend

Existen herramientas como *Visual Studio Code* que ayuda a la mejor organización de los módulos, funciones y carpetas del proyecto, en base a ello se han creado distintas carpetas que separan los componentes que se utilizaran en el proyecto, como la carpeta *pages* que contendrá las vistas de nuestro sistema web, la carpeta *components* que contendrá los componentes reutilizables para el proyecto o la carpeta *routes* que contiene las rutas de una vista a otra, por tanto la estructura se define a través de la utilidad y facilidad de encontrar cada uno de los archivos o vistas del sistema como se indica en la **Fig 3**.

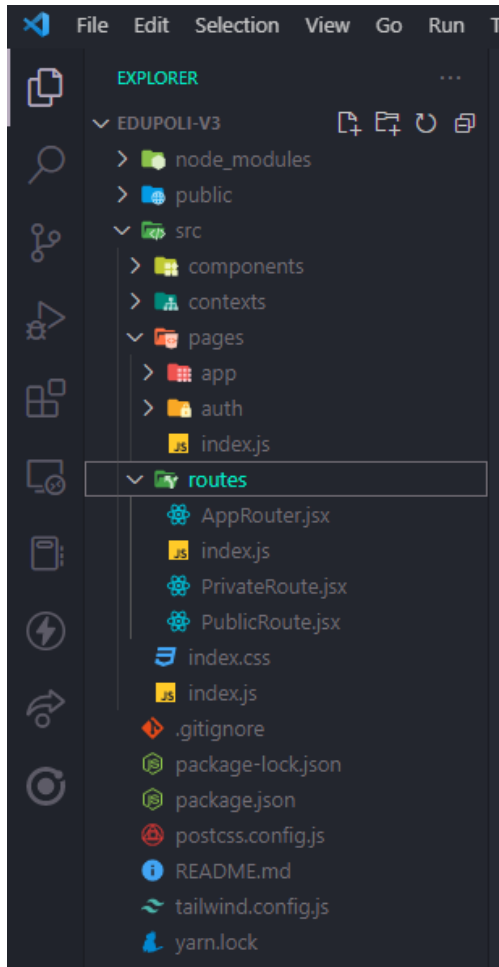


Fig. 3 Organización de carpetas y archivos del *frontend*

Roles de usuario

Se presenta en la **Fig. 4** los distintos roles que puede tener el usuario y sus funcionalidades dentro del sistema *web*.

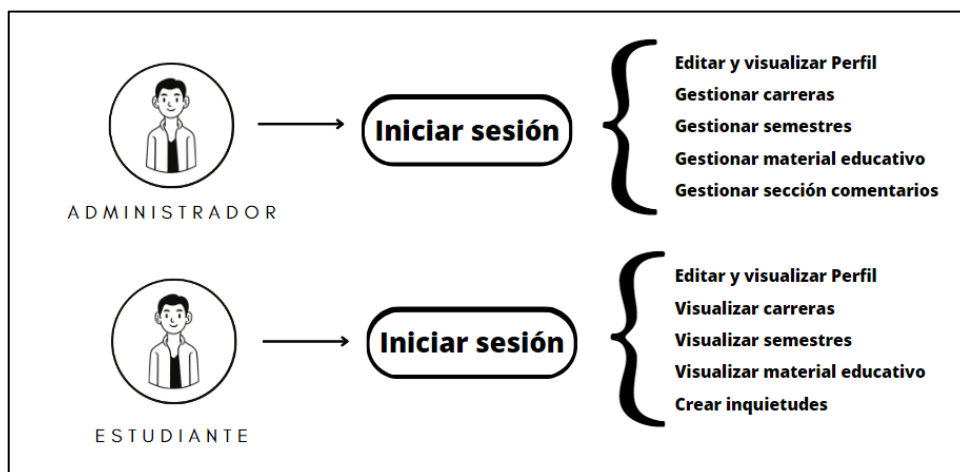


Fig. 4 Roles de usuario

Sprint 1. Inicio de sesión y recuperación de contraseña del usuario

Tareas asignadas para el *Sprint 1* son:

- Definición de campos y creación de funciones para el *login* y recuperación de contraseña.
- Validación de los campos en *login* y recuperar contraseña.
- Prueba unitaria *Sprint 1*.

Definición de campos y creación de funciones para el *login* y recuperación de contraseña

Comenzamos a desarrollar prototipos para el apartado del *login* y recuperación de contraseñas, por tanto, establecemos las distintas funciones para poder iniciar sesión con la ayuda del *backend*, además establecemos los colores, componentes y campos que se mostraran al usuario al momento de iniciar la página. Se muestra en la **Fig. 5**, la vista del *login*, la **Fig. 6** muestra la pantalla de recuperar contraseña, en la cual enviamos al correo la activación para que pueda cambiar su contraseña, de manera que le presenta la **Fig. 7**, de la pantalla de confirmar contraseña. Dentro de las funciones se establecen la llamada a solicitudes mediante *axios*, por tanto, presentan la información necesaria para realizar cambios o ingresar al sistema.

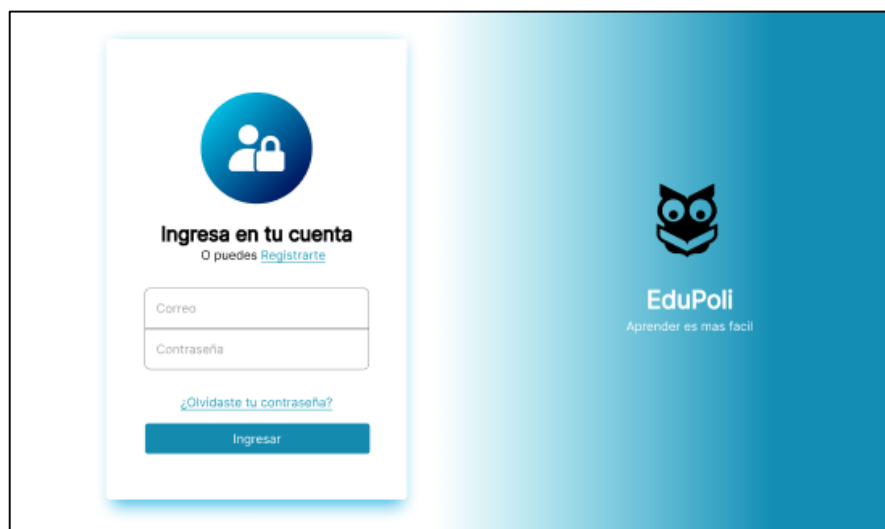


Fig. 5 Pantalla de iniciar sesión

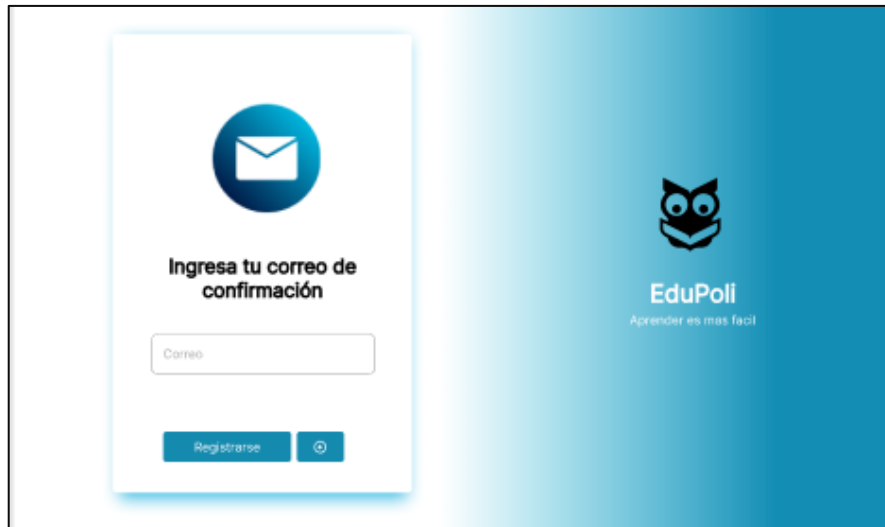


Fig. 6 Pantalla de recuperar contraseña

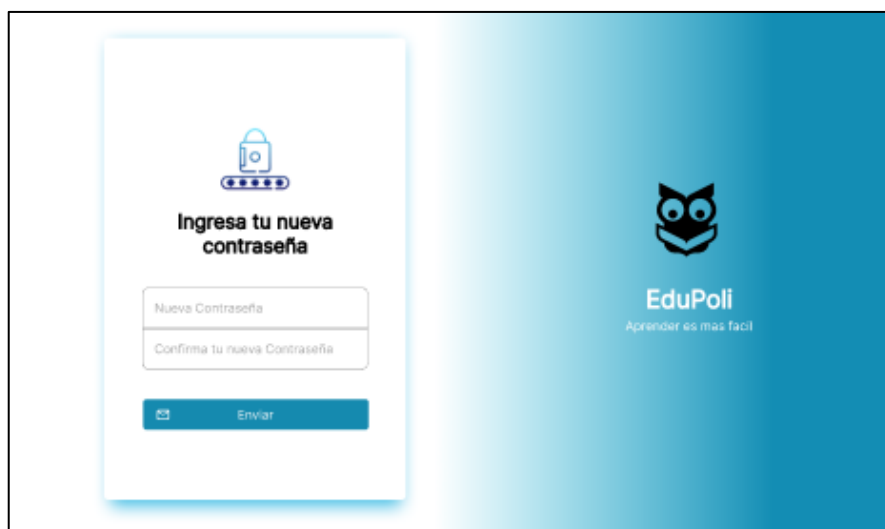


Fig. 7 Pantalla de confirmar contraseña

Validación de los campos en *login* y recuperar contraseña

En las diferentes pantallas tanto en iniciar sesión como en recuperar contraseña se validan si los campos son correctos y si existen dentro de la base de datos, de manera que si existe un error presentara un mensaje de que las credenciales están incorrectas, similar a la **Fig. 8**. De otra manera se ingresará a la página principal en donde se presentarán la ventana del *dashboard* como en la **Fig. 9**.

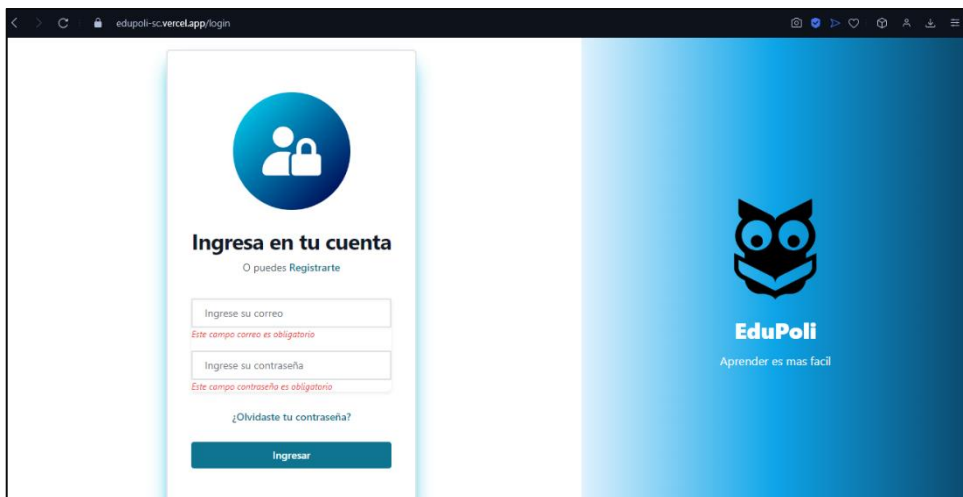


Fig. 8 Validación de usuario en iniciar sesión

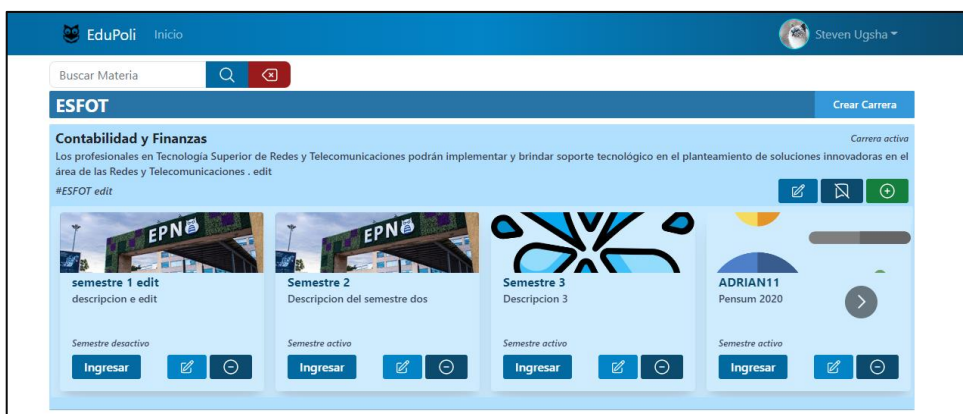


Fig. 9 Validación de usuario en iniciar sesión

Prueba unitaria *Sprint 1*

Comprobamos las funciones y componentes utilizados para el desarrollo de la pantalla del *login* de los usuarios, con ayuda de *Cypress* la cual es una herramienta para realizar pruebas unitarias de manera fácil y rápida mientras se puede observar en tiempo real los resultados de las peticiones de al *Frontend* por parte del usuario, a continuación, se presenta la prueba unitaria para el *login* en la **Fig.10** y su resultado con las peticiones y métodos utilizados para el ingreso al sistema en la **Fig. 11**. Este apartado de pruebas unitarias se explica mejor en el **ANEXO II**.

```
cypress > e2e > spec.cy.js > ...
1 describe("Inicio de sesion", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000");
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com");
5     cy.get('input[name="password"]').type("Adrian123@");
6     cy.get("button").contains("Ingresar").click();
7     cy.get("button").contains("Crear Carrera").click();
8   });
9 });
```

Fig. 10 Código prueba unitaria

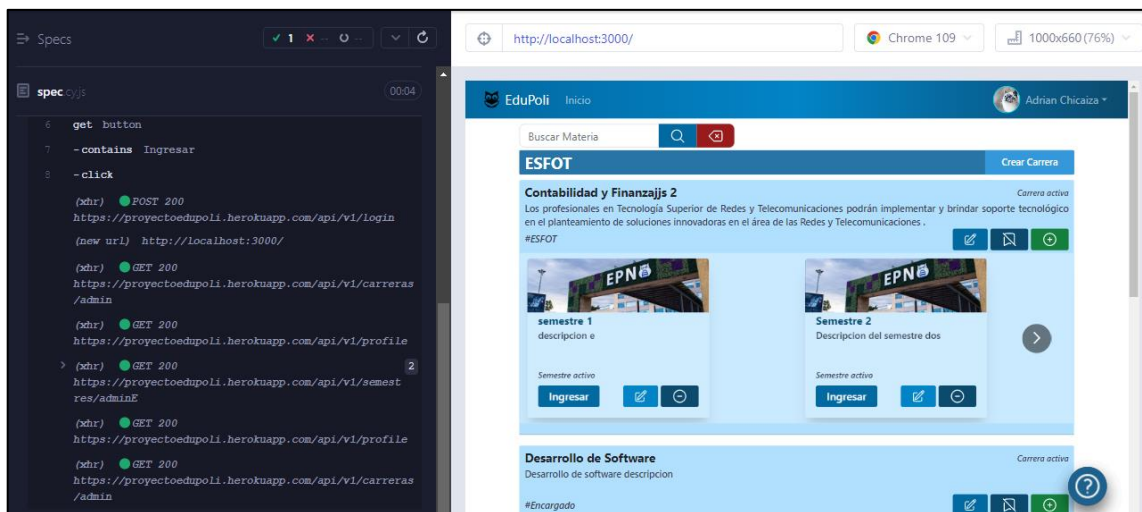


Fig. 11 Resultado de peticiones

Sprint 2. Registrar usuario y editar perfil de usuario

Tareas asignadas para el *Sprint 2* son:

- Definición de campos para el registro de usuarios y edición del perfil.
- Creación de funciones y validaciones para el registro de usuarios.
- Prueba unitaria *Sprint 2*.

Definición de campos para el registro de usuarios y edición del perfil.

En esta sección se establecen los estilos, componentes y campos que tendrán los formularios para el registro de usuario y la edición de su perfil como se muestra en la **Fig.12** y también en la **Fig.13**.

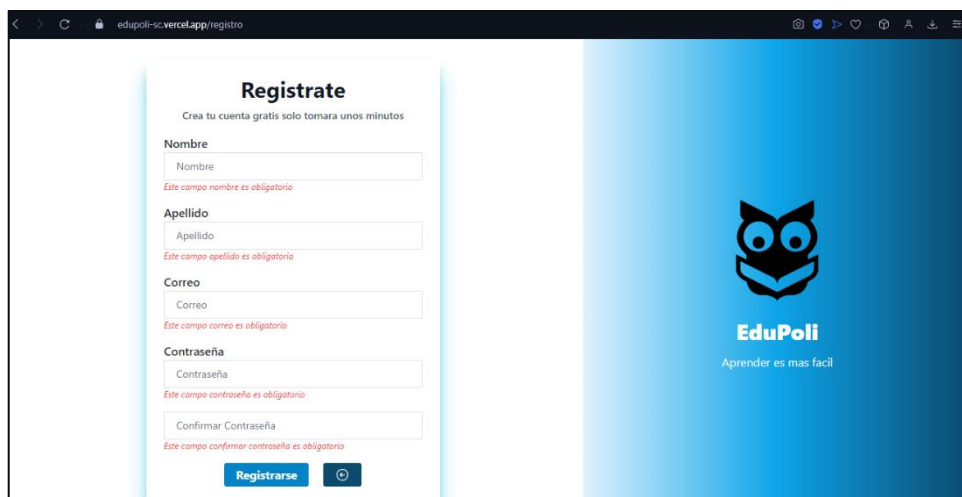


Fig. 12 Pantalla de registro de usuario y validaciones

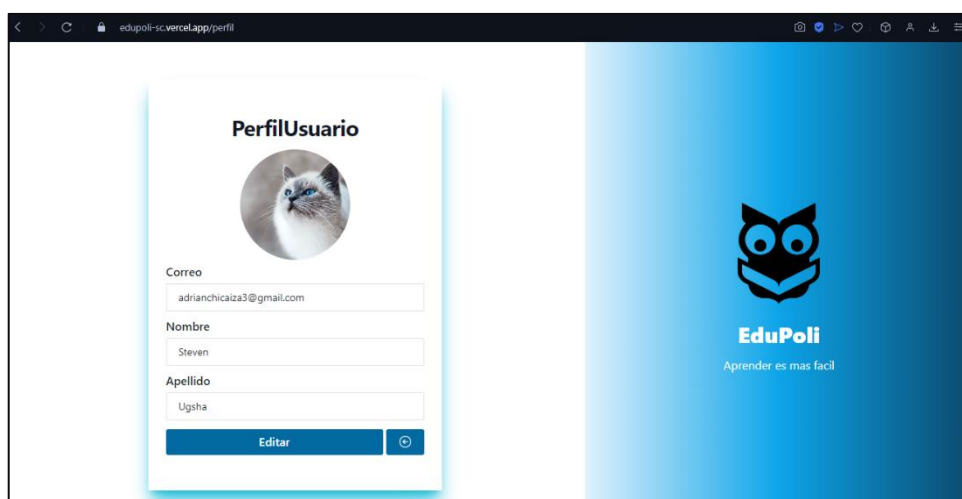


Fig. 13 Pantalla perfil de usuario

Creación de funciones y validaciones para el registro de usuarios.

Posteriormente, se establecen los métodos para el registro a partir de *endpoints* del *backend* y con la ayuda de *axios* para poder enviar nuestro registro a través del formulario, y también con la utilización del método *get* y *post* para editar y ver los datos del usuario. Se muestra en la **Fig. 14** la prueba de la traída de datos del usuario de manera que los llena en campos como nombre, apellido y correo, además de proporcionar un botón para que edite su información.

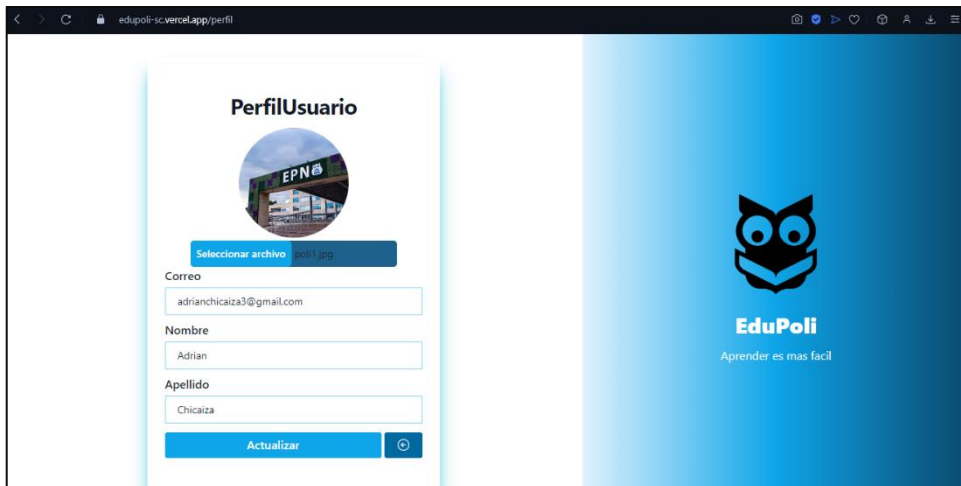


Fig. 14 Pantalla de perfil usuario editado

Prueba unitaria *Sprint 1*

Comprobamos los métodos utilizados para la vista de perfil de usuario con la ayuda de pruebas unitarias con la herramienta de Cypress, a continuación, un ejemplo del código de prueba unitaria presentado en la **Fig. 15** y su resultado del mismo en la **Fig. 16**. Este apartado de pruebas unitarias se explica con más detalle en el **ANEXO II**.

```
cypress > e2e > pVerPerfil.cy.js > ...
1 describe("Ver Perfil del Usuario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // ver perfil
9     cy.get('a[id="collapsible-nav-dropdown"]').first().click()
10    cy.get('a[id="EditPerfil"]').first().click()
11  });
12 }
13
```

Fig. 15 Código de ver datos de usuario

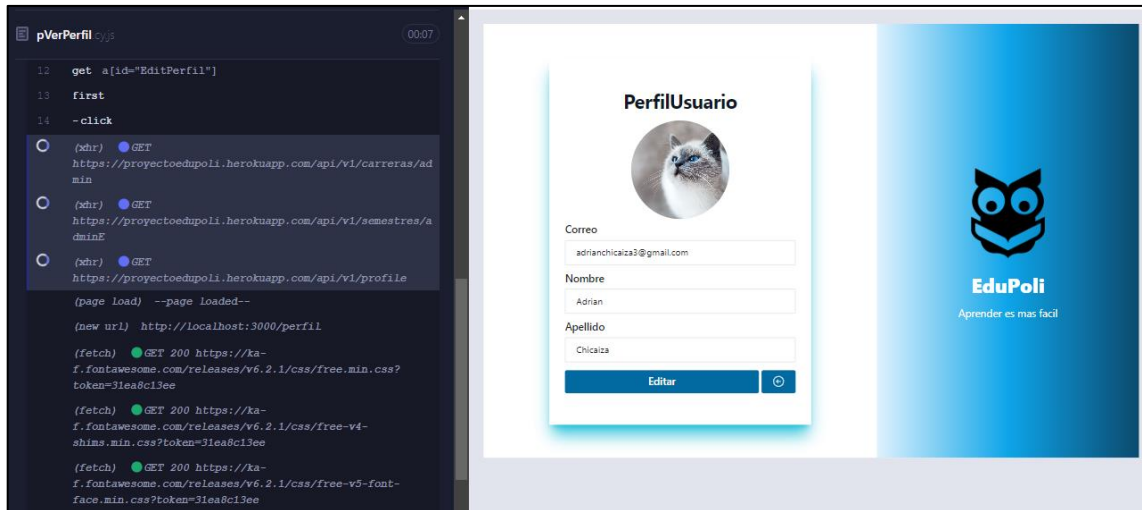


Fig. 16 Tanteo de peticiones

Sprint 3. Gestionar carreras y visualizar información

Tareas asignadas para el *Sprint 3* son:

- Diseño de prototipos e interfaz para la vista del usuario.
- Campos y modales para la gestión de carreras.
- Validación de campos para gestión de carreras.
- Prueba unitaria *Sprint 3*.

Diseño de prototipos e interfaz para la vista del usuario

El diseño del prototipo de la vista de semestre y carreras se realizó en base a los requerimientos de los clientes como indica la **Fig. 17**, mientras que el diseño final para la vista de las carreras se cambió estilizando y agregando componentes como indica la **Fig. 18**. La información de cada semestre dependiendo de su carrera se muestra a través del uso de *endpoints* del *backend*.



Fig. 17 Prototipo de la interfaz de Carreras

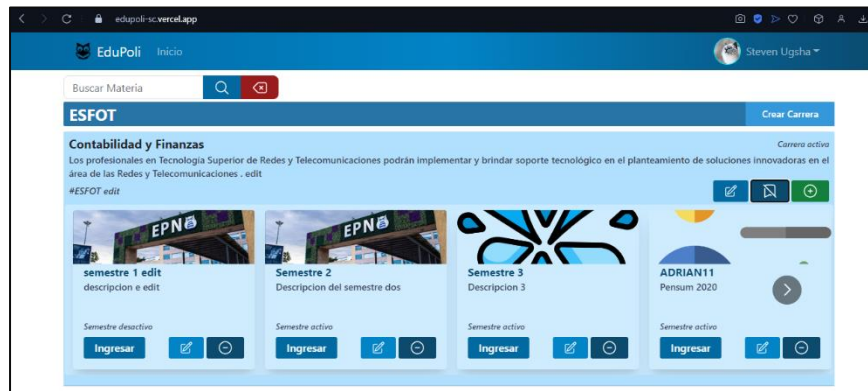


Fig. 18 Interfaz final para el usuario

Campos y modales para la gestión de carreras

Establecemos los campos necesarios dentro de un modal para la creación de una carrera, tanto como son el nombre, descripción y un encargado de la carrera como se muestra en la **Fig. 19**, para facilitar su creación. Para la edición de la carrera se presenta un modal similar en el cual no es necesario ingresar todos los datos ya que al actualizar se puede actualizar un solo campo indicado en la **Fig. 20**. Por último, presenta un aviso de confirmación cuando se requiere activar o desactivar la carrera como indica la **Fig. 21**.

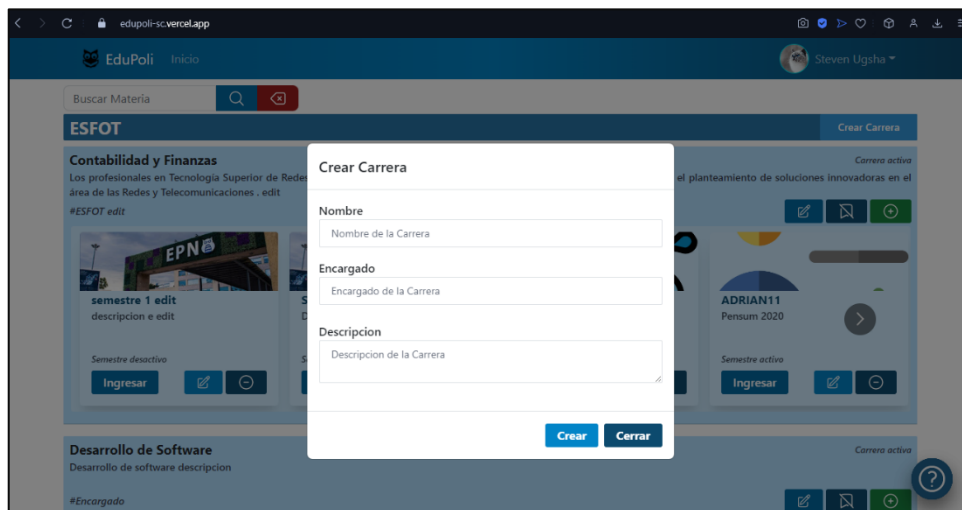


Fig. 19 Vista de crear la carrera

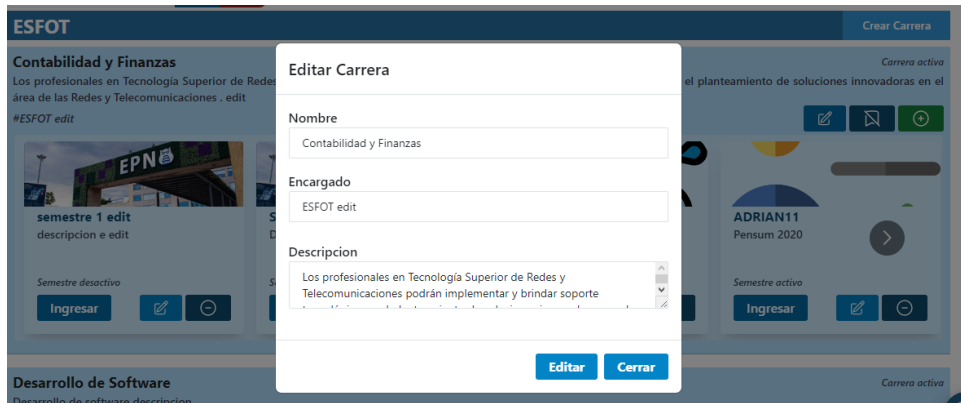


Fig. 20 Modal para editar la carrera



Fig. 21 Mensaje de confirmación

Validación de campos para gestión de carreras

Si el administrador no llena cada uno de los campos obligatorios del modal para crear una carrera, este le presentara un mensaje en rojo que le muestra que campo debe llenar, un ejemplo lo indica la Fig. 22.

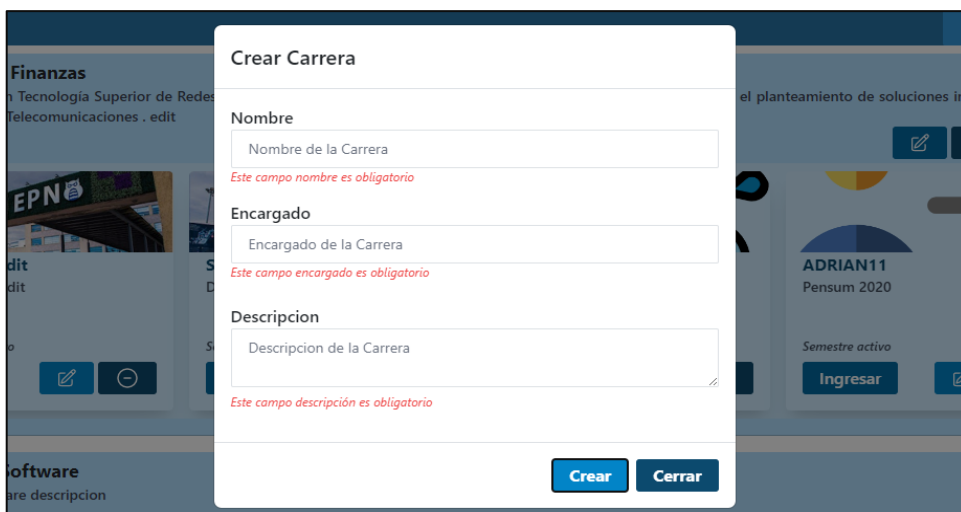


Fig. 22 Validaciones de campos en la creación de carreras

Prueba unitaria *Sprint 3*

Comprobamos la vista de la página principal y los componentes como son los modales, de manera que se realizan pruebas unitarias para verificar una creación de carrera en la **Fig. 23** y sus resultados se observan en la **Fig. 24**. Este apartado de pruebas unitarias se explica con más detalle en el **ANEXO II**.

```
cypress > e2e > pCrearCarrera.cyjs > ...
1 describe("Crear Carrera", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Crear una Carrera
9     cy.get("button").contains("Crear Carrera").click({ force: true }) //aquí abre el modal
10    cy.get('input[name="nombre"]').type("Agua y Saneamiento Ambiental")
11    cy.get('input[name="encargado"]').type("Encargado")
12    cy.get('textarea[name="descripcion"]').type("Descripcion para la carrera de Agua y Saneamiento Ambiental")
13    cy.contains("div", "Cerrar").contains("Crear").click();
14  });
15 });
16
```

Fig. 23 Prueba Unitaria de crear carrera

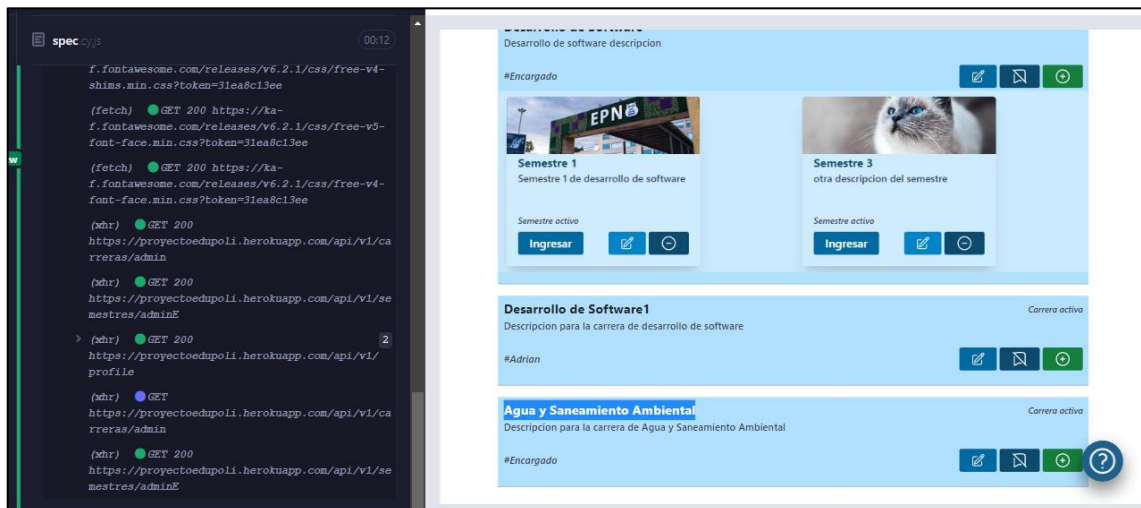


Fig. 24 Tanteo de peticiones

Sprint 4. Gestión y visualización de materiales educativos

Tareas asignadas para el *Sprint 4* son:

- Diseño de la interfaz de semestres y uso de métodos para el *CRUD* de las materias y documentos.
- Campos y modales para la gestión de materias y sus documentos.
- Validación de campos para gestión de materias y documentos.
- Prueba unitaria *Sprint 4*.

Diseño de la interfaz de semestres y uso de métodos para el CRUD de las materias y documentos.

El diseño de la interfaz del usuario para la vista de las materias con sus documentos se realizó de acuerdo a los requerimientos de los usuarios los cuales requerían ver los documentos disponibles de cada carrera como se muestra en la **Fig. 25**. Además de usar los *endpoints* del *backend* para la gestión de cada materia y documento.

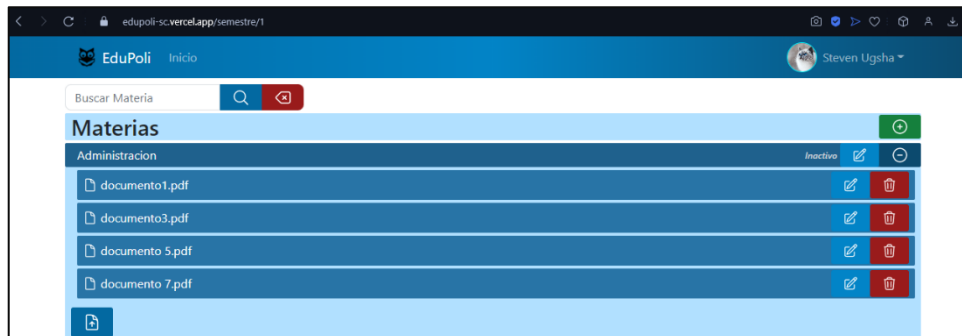


Fig. 25 Vista de materias y documentos

Campos y modales para la gestión de materias y sus documentos

En la gestión de las materias y sus documentos se establecen distintos modales para facilitar su creación, como indicada la **Fig. 26** para la creación de una nueva materia, para su edición se muestra la **Fig. 27** y para cambiar el estado de esta se presenta un mensaje de confirmación en la **Fig. 28**. Cada uno de estos modales utiliza un *endpoint* del *backend* para realizar la correcta gestión de las materias en la base de datos.

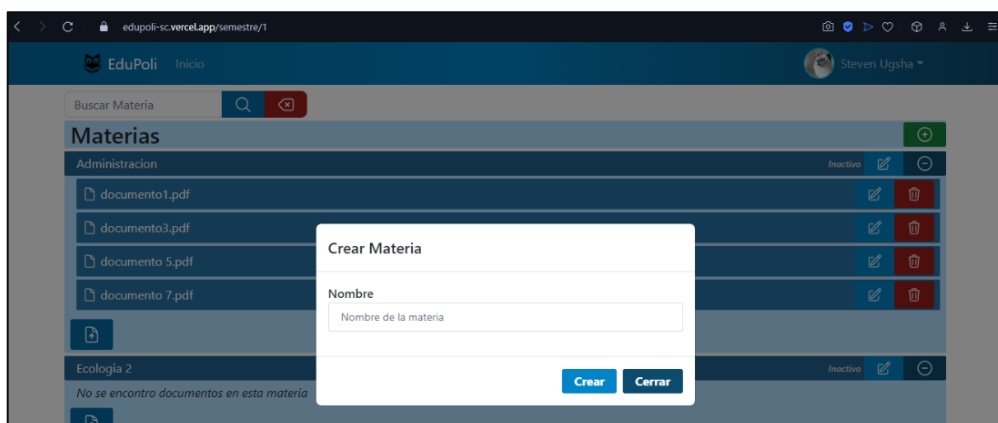


Fig. 26 Creación de una materia

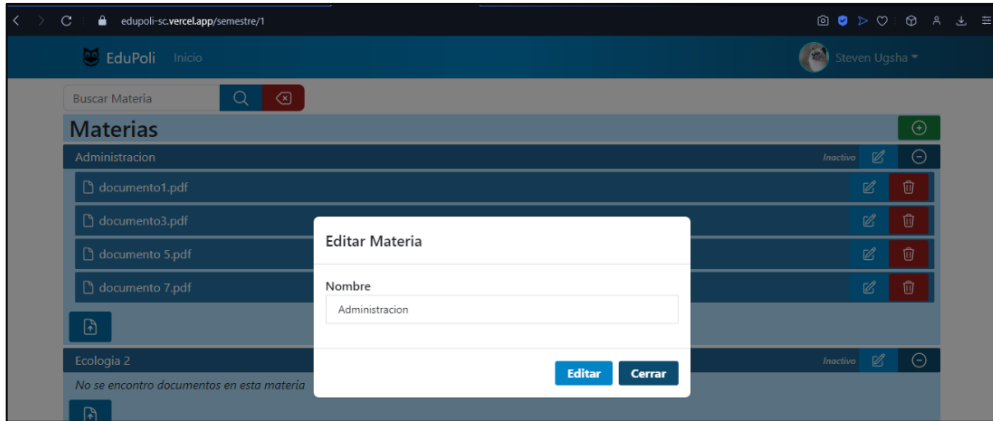


Fig. 27 Edición de una materia

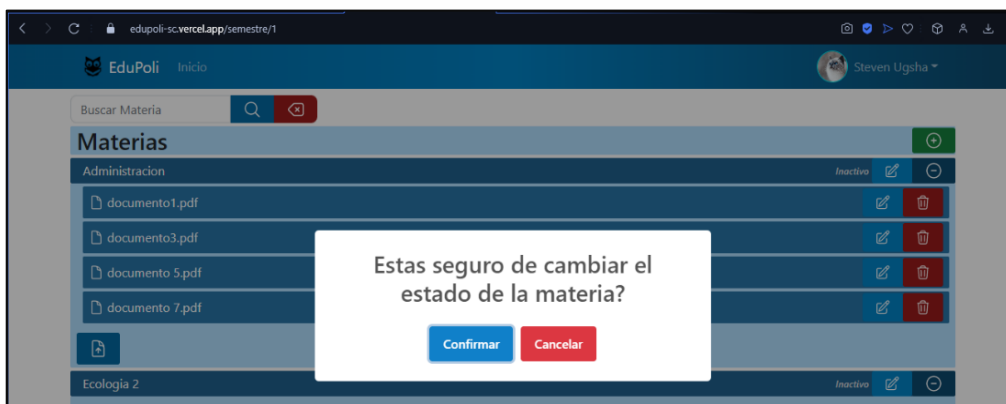


Fig. 28 Cambiar estado de una materia

Mientras que en la **Fig. 29**, se ingresa un documento en una materia que escogemos, esto con la ayuda de un modal y un *input file* el cual requiere de un documento de tipo *pdf* para la subida al sistema *web*, de la misma forma para su gestión se muestran modales como en la **Fig. 30** para la edición del documento o la **Fig. 31** para eliminar un documento de la materia a través de un mensaje de confirmación.

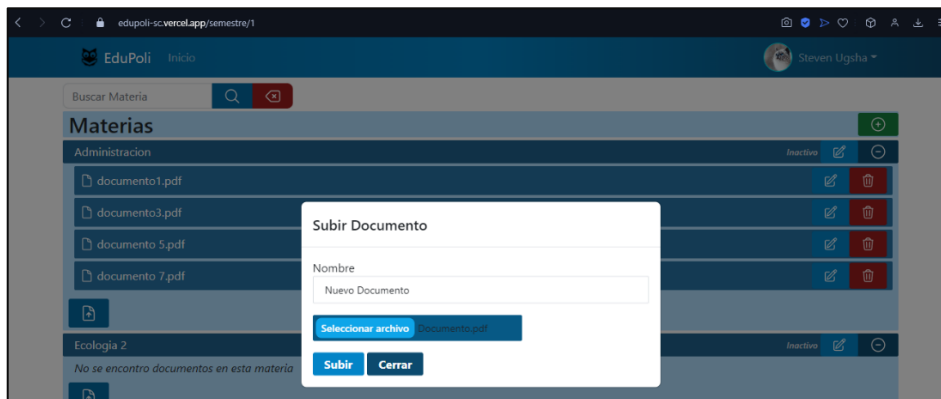


Fig. 29 Subida de un documento

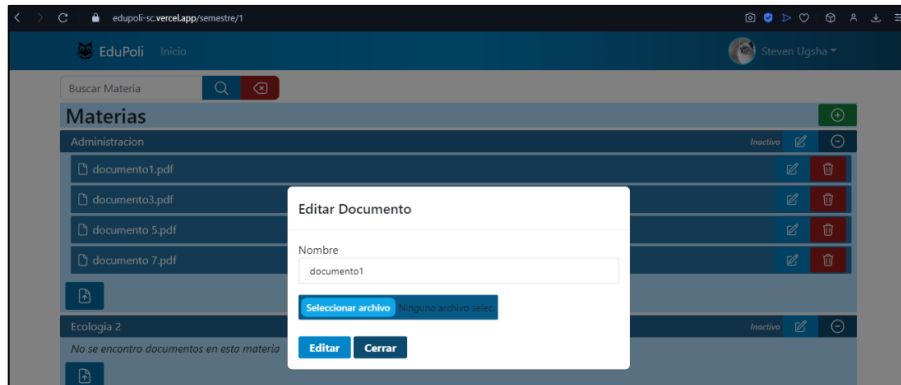


Fig. 30 Edición del documento

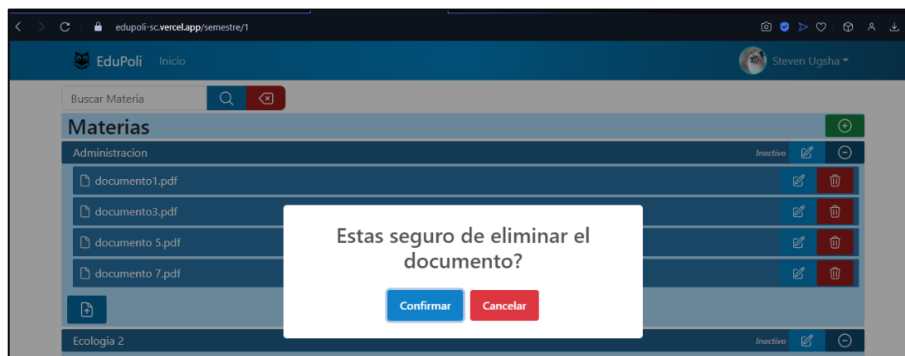


Fig. 31 Eliminación del documento

Validación de campos para gestión de materias y documentos

El administrador al momento de crear una materia debe llenar el campo del nombre para poder crearlo, caso contrario se presentará un aviso de que debe llenar el campo como se muestra en la **Fig. 32**, de la misma manera en la **Fig. 33** se presenta la validación al momento de subir un documento al cual se le debe especificar un nombre y el documento tipo pdf.

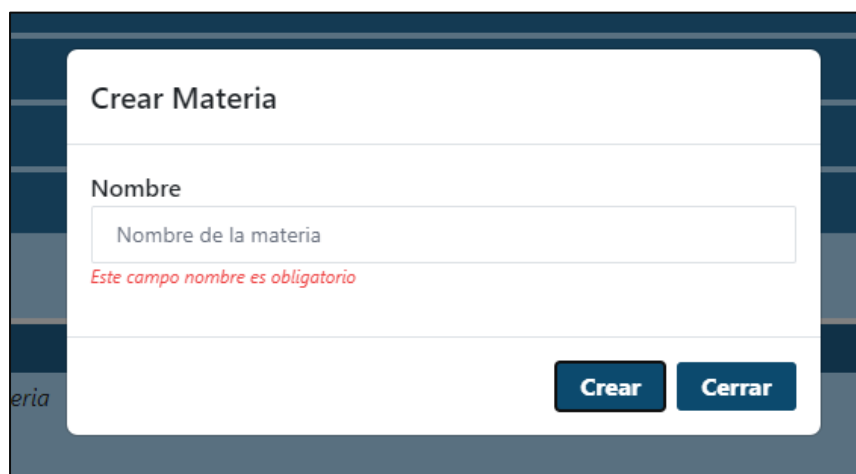


Fig. 32 Validación de materia

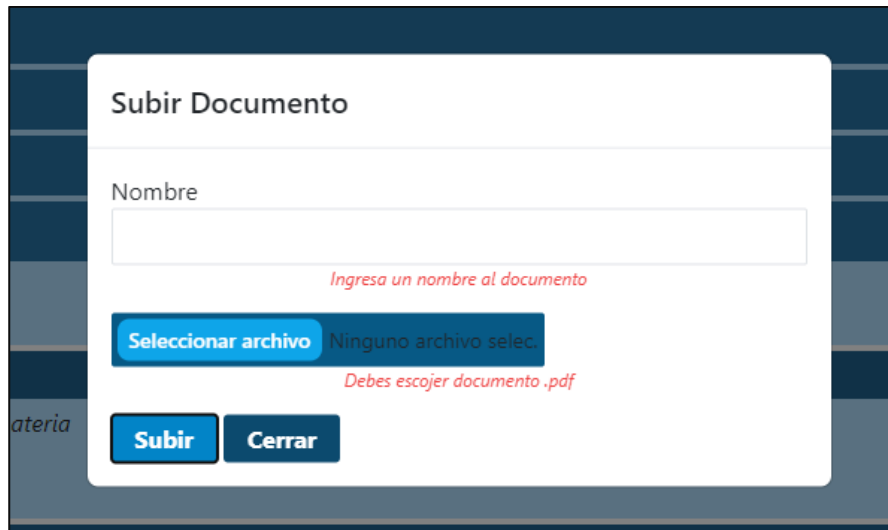


Fig. 33 Validación de documento

Prueba unitaria *Sprint 4*

Comprobamos mediante una prueba unitaria la creación de una materia como se muestra en la **Fig. 34** y podemos observar sus resultados en la **Fig. 35**, en donde se muestran todas las peticiones completadas por parte del usuario. Este apartado de pruebas unitarias se explica con más detalle en el **ANEXO II**.

```

cypress > e2e > pCrearMateria.cy.js > ...
1 describe("Crear Materia", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Ver y crear una materia
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="CrearMat"]').click()
11    cy.get('input[name="nombre"]').type("Calculo")
12    cy.get('button[id="CrearMatSelect"]').first().click()
13  });
14 });

```

Fig. 34 Prueba Unitaria crear materia

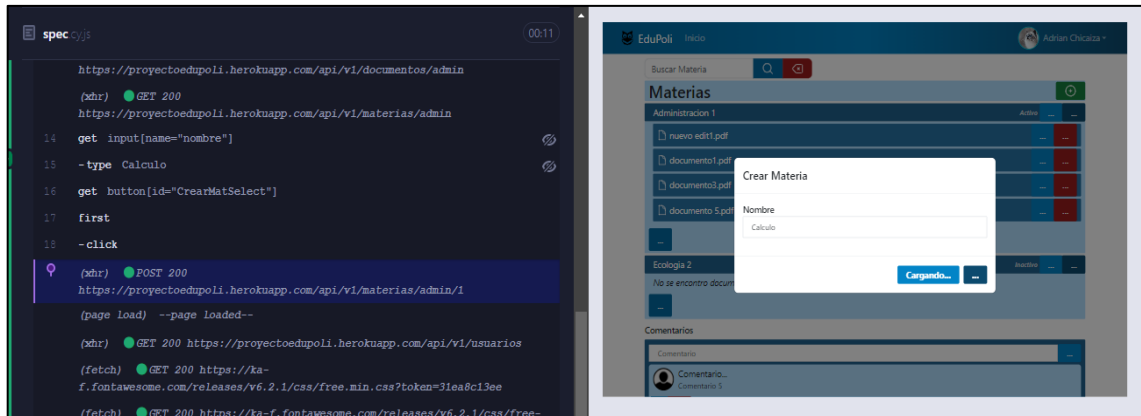


Fig. 35 Tanteo de peticiones

Sprint 5. Búsqueda de materias y gestión de inquietudes

Tareas asignadas para el *Sprint 5* son:

- Implementación de componente de búsqueda por materia.
- Implementación de la sección de inquietudes o comentarios.
- Sección de preguntas frecuentes.
- Prueba unitaria *Sprint 5*.

Implementación de componente de búsqueda por materia

La interfaz o componente realizado para la búsqueda de materias se realiza en base a los requerimientos de los usuarios, los cuales quieren encontrar más rápido la información y documentos que contienen una materia como se muestra en la **Fig. 36**, el cual mostrara solo la materia buscada.

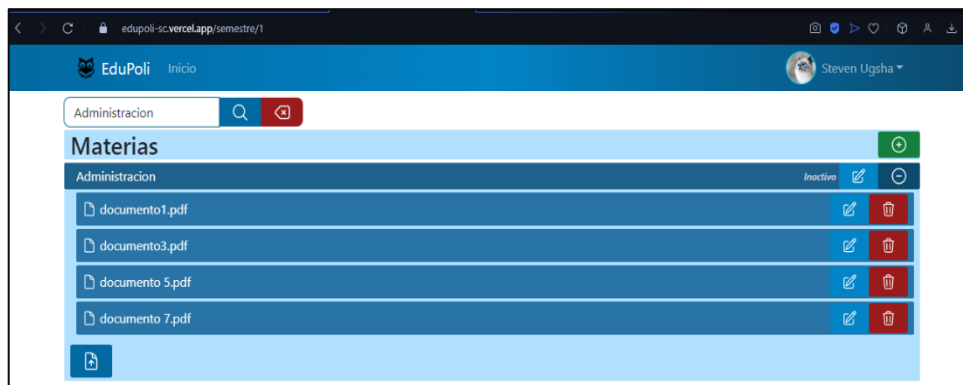


Fig. 36 Componente de búsqueda por materia

Implementación de la sección de inquietudes o comentarios

En la sección de inquietudes o comentarios se establece un campo en el cual el usuario *logueado* puede escribir una inquietud respecto al semestre en el que se encuentra o dejar un comentario del mismo, se muestra un ejemplo en la **Fig. 37**.

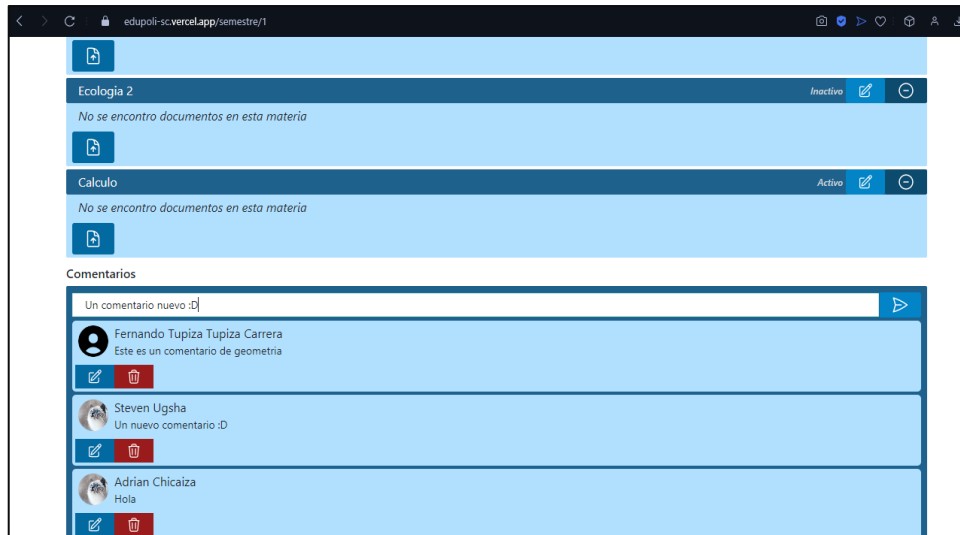


Fig. 37 Sección de inquietudes o comentarios

La gestión de las inquietudes se puede realizar dependiendo de los roles, el administrador puede editar y eliminar los comentarios, mientras que el rol estudiante puede editar o eliminar solo el comentario que realizó un ejemplo lo indica la **Fig. 38** y **Fig. 39**.

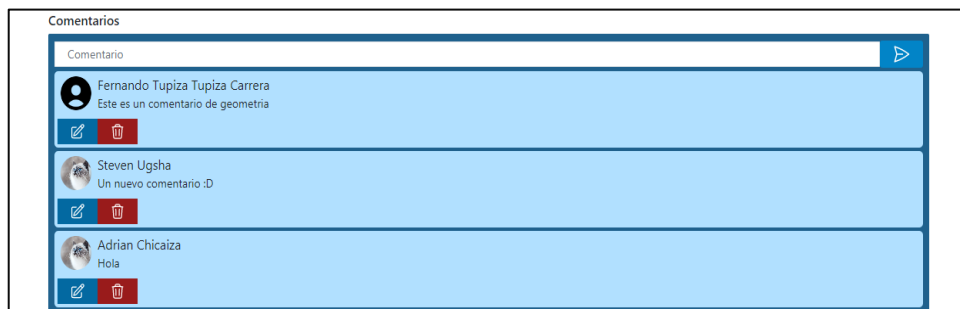


Fig. 38 Sección de inquietudes con rol administrador

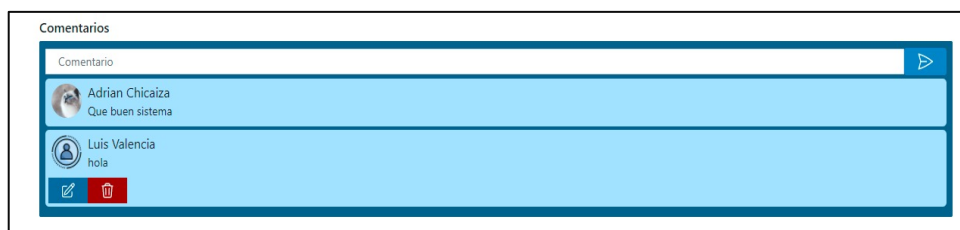


Fig. 39 Sección de inquietudes con rol estudiante

Dependiendo de su rol se puede editar el comentario, un ejemplo lo indica la **Fig. 40** o descartarlo como la **Fig. 41**.

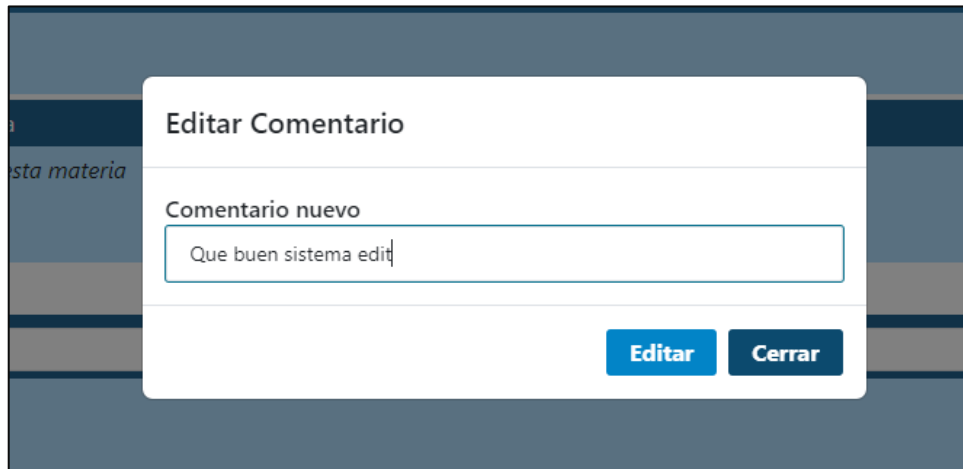


Fig. 40 Editar comentario

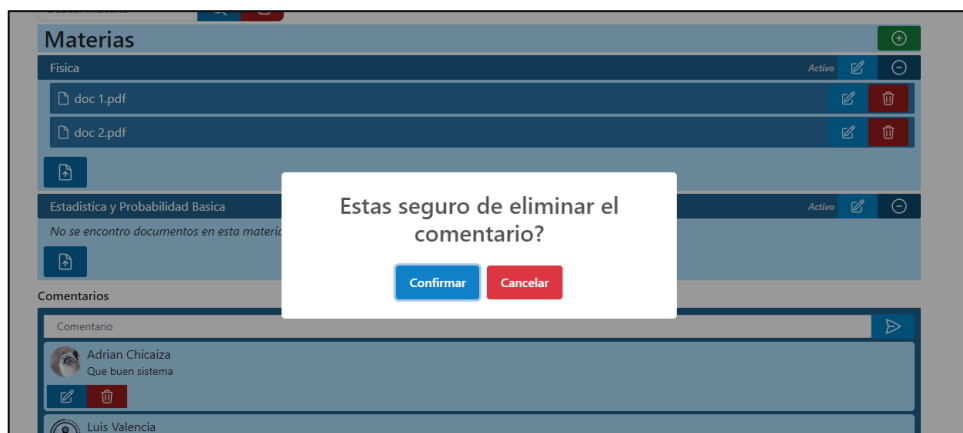


Fig. 41 Eliminar Comentario

Sección de preguntas frecuentes

Se desarrolla el componente de acuerdo a los requerimientos de usuario para una sección en donde puedan encontrar preguntas frecuentes de acuerdo al sistema web, que responde en tiempo real, además de realizar un diseño limpio y agradable para el usuario, un ejemplo lo indica la **Fig. 42**.

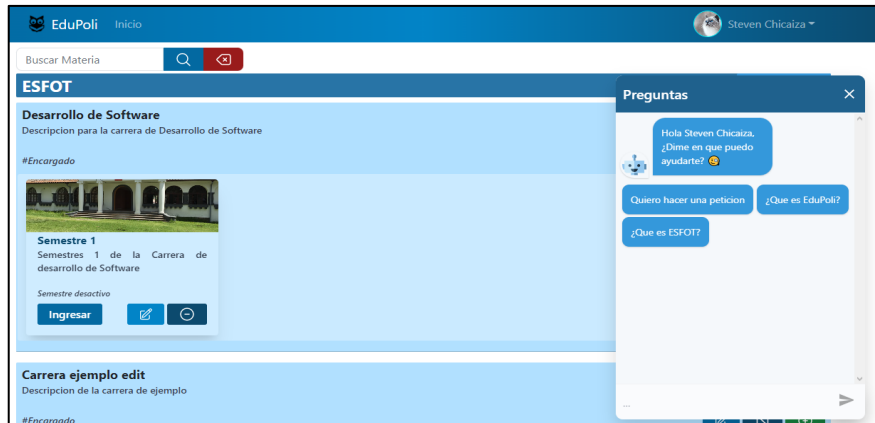


Fig. 42 Apartado preguntas frecuentes

Prueba unitaria *Sprint 5*

Comprobación de los métodos y peticiones al momento de crear un comentario mediante una prueba unitaria con *Cypress*, que se presenta en la **Fig. 43** y sus resultados en la **Fig. 44**. Este apartado de pruebas unitarias se explica con más detalle en el **ANEXO II**.

```

cypress > e2e > pComentar.cy.js > ...
1 describe("Crear Comentario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Crear Semestre
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('input[id="comentario"]').type("Un nuevo comentario :D")
11    cy.get('button[id="ComentarSend"]').first().click()
12  });
13 });

```

Fig. 43 Prueba Unitaria para comentar

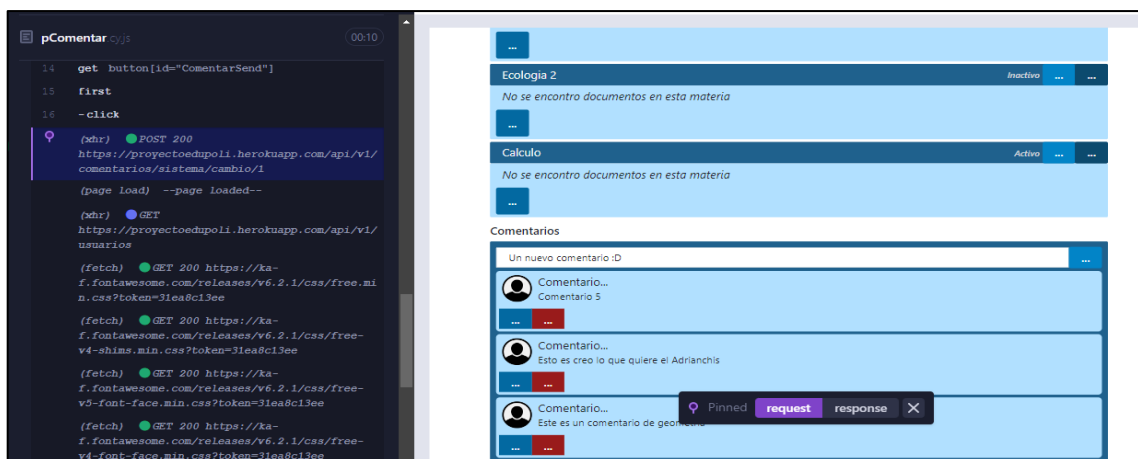


Fig. 44 Tanteo de peticiones

Sprint 6. Pruebas del sistema y despliegue

Una vez culminado los *Sprint*, se procede a realizar pruebas y nos enfocamos en las:

- Pruebas unitarias para el componente *frontend*.
- Pruebas de carga del componente *frontend*.
- Pruebas de compatibilidad para el componente *frontend*.
- Despliegue del *frontend*.

Pruebas unitarias para el componente *frontend*

Las pruebas consisten en la comprobación de un fragmento de código que funcionen bien y de acuerdo a los requerimientos del usuario, es por ello que existen distintas herramientas y métodos para realizarlas, para el presente documento se realiza las pruebas unitarias para el componente *Frontend* con la herramienta de *Cypress* el cual permite comprobar el código *JavaScript* de manera que los métodos y funciones que se realizan sean efectuados de la mejor manera, esto con la ayuda de una página de *testing* en donde se muestra en tiempo real la ejecución de cada acción en el *frontend*, además de las peticiones que se realizan [37].

Posteriormente la **Fig. 45** indica el código utilizado para la prueba unitaria del *login*, además de presentar los resultados en la **Fig. 46** en donde *Cypress* presenta las peticiones que se han realizado al momento de realizar la acción de iniciar sesión en el componente *Frontend*. Las pruebas unitarias de cada vista con su código y resultado se explican a detalle en el apartado del **ANEXO II**.

```
cyress > e2e > pLogin.cy.js > ...
1 describe("Inicio de sesion", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7   });
8 });
```

Fig. 45 Prueba unitaria *Login*

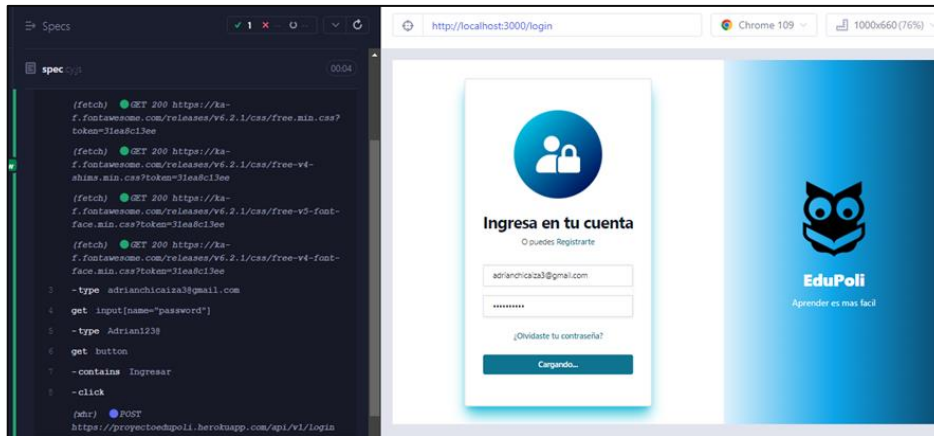


Fig. 46 Tanteo de peticiones

Pruebas de carga del componente frontend

Con el fin de realizar la prueba de carga utilizamos el sitio web de *PageSpeed Insights* de Google, el cual realizar la simulación de interacción de varios usuarios simultáneamente y así obtener en tiempo real los resultados de carga que presenta la página del lado del *Frontend*. En la **Fig. 47** podemos ver los resultados en donde se determina una calificación por los módulos utilizados.

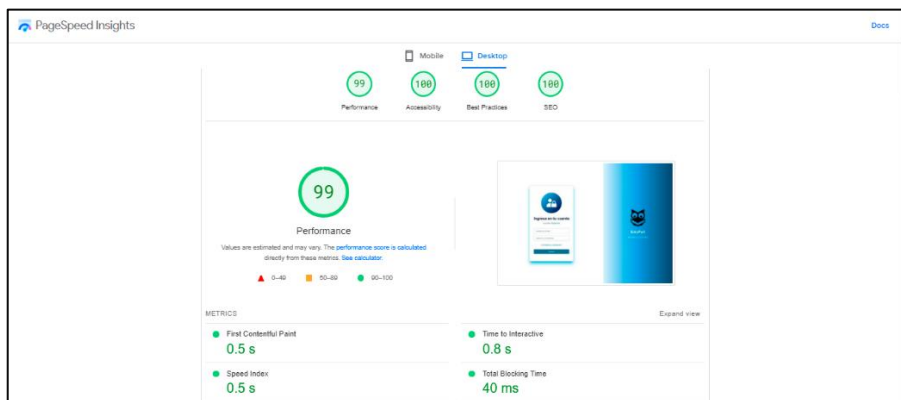


Fig. 47 Frontend bajo una prueba de carga

Pruebas de compatibilidad para el componente frontend

Este tipo de pruebas muestran cómo funciona en diferentes plataformas la interfaz a la que los usuarios del sistema tendrán acceso. Se realizó un análisis de los resultados de cada navegador como lo indica la **Tabla IX**, la cual presenta con más detalle este tipo de pruebas en los tres navegadores: *Google Chrome*, *Microsoft Edge* y *Opera*, cabe mencionar que no se presentaron inconvenientes y se presentan de manera correcta en cada navegador.

TABLA IX: Navegadores utilizados para pruebas

NOMBRE	VERSIÓN	OBSERVACIÓN
Google Chrome	99.0.4844.84	Completamente funcional
Microsoft Edge	109.0.1518.78	Completamente funcional
Opera	89.0.4447.71	Completamente funcional

En *Google Chrome* no se presentó ningún problema al iniciar, como vemos se nos presenta la página de *login* en la **Fig. 48** y el *dashboard* en la **Fig. 49**. Para verlo de forma más didáctica los resultados ejecutados en los distintos navegadores nos dirigimos al **ANEXO II**, que se presenta en este documento.

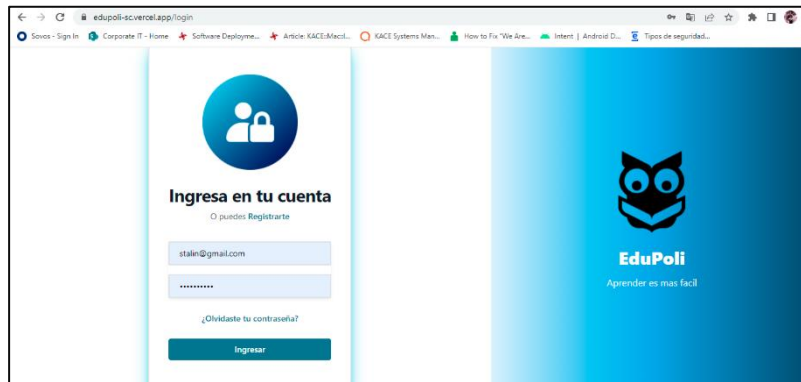


Fig.48 Pagina de iniciar sesión (Google Chrome)

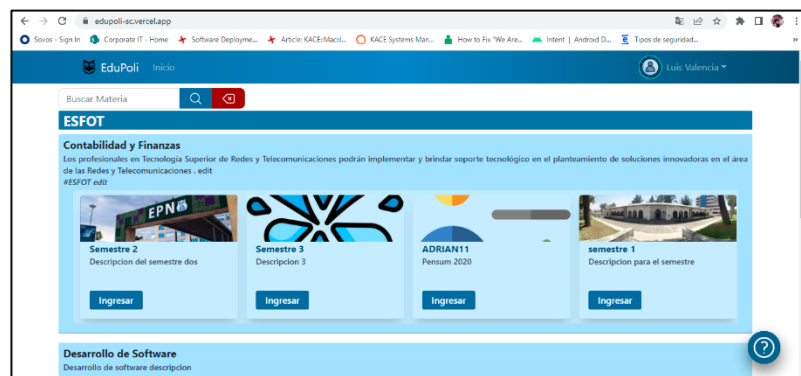


Fig.49 Página de inicio (Google Chrome)

Despliegue del *frontend*

Una vez culminado las pruebas en nuestro componente *frontend* pasamos a realizar el despliegue en alguna herramienta de alojamiento de *frontend*, como son *Netlify* o *Vercel*, el cual nos proporciona un hosting gratuito para poder compartir nuestro componente y así

sea consumido a través de internet. La herramienta utilizada para el despliegue del *frontend* de EDUPOLI es *Vercel* la cual es una herramienta intuitiva que se conecta directamente con la rama principal del proyecto y los despliega en sus servidores, dando como resultado actualizaciones automáticas cuando se actualiza el repositorio de *GitHub*, un ejemplo del despliegue de la aplicación se muestra en la **Fig. 50** y la *url* para consumir el sistema EDUPOLI.

<https://edupoli-sc.vercel.app/login>

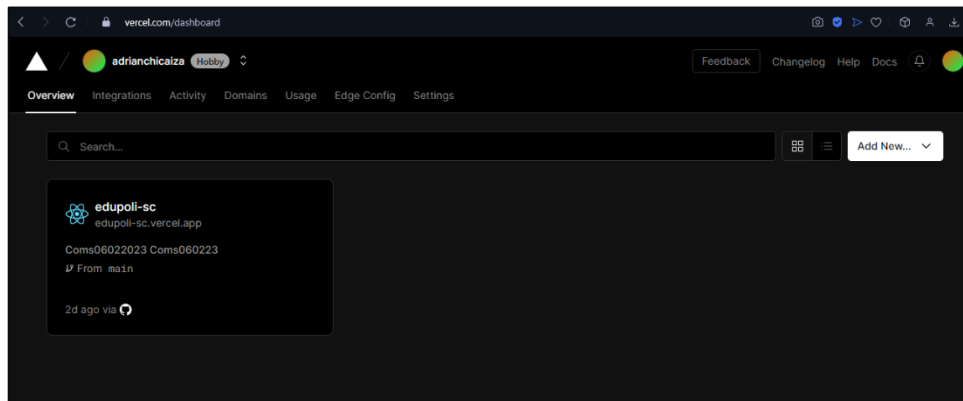


Fig.50 Despliegue en Vercel

CONCLUSIONES

A continuación, se presentan los resultados obtenidos en el desarrollo del componente *frontend* del actual sistema:

- Se obtiene como resultado que “EDUPOLI” abarca en su totalidad los objetivos y alcance de los requerimientos de los usuarios documentados en las historias de usuario y artefactos de la metodología ágil *SCRUM*.
- Utilizar la metodología *Scrum* ha hecho posible realizar el proyecto en el tiempo establecido y llevándolo de la mejor manera en cada uno de los *Sprints* realizados.
- Definir de manera correcta los requerimientos de los usuarios al inicio del proyecto ha permitido escoger las mejores herramientas y metodologías para llevar a cabo una correcta realización del proyecto en el componente *frontend*.
- Establecer prototipos de cada una de las vistas para el componente *frontend* ha hecho que agilizará el trabajo en la creación de componentes con la ayuda del lenguaje de *ReactJS*.

RECOMENDACIONES

En este apartado se muestran las recomendaciones que se llevaron a cabo para el desarrollo del componente *frontend* en el actual proyecto de integración curricular:

- Se recomienda a los administradores y estudiantes que utilicen el sistema *web* ver el manual para utilizarlo de la mejor manera.
- Se recomienda implementar más medidas de seguridad al momento de ingresar al sistema, pues cada día existen nuevas vulnerabilidades en las que se puede robar la información de los usuarios que ingresan.
- Es recomendable generar la carpeta *build* al culminar nuestro proyecto para poder desplegar lo de manera más fácil en cualquiera de las herramientas de despliegue cómo son *heroku*, *netlify* o *vercel*.
- El filtrado de cada materia por parte del *backend* no estaba estructurado, para ello se realiza un filtrado por parte del componente *frontend* a través del llamado de toda la información y estructurando lo que se necesita a través de *maps*, como resultado se obtiene los semestres de cada carrera, las materias de cada semestre, los documentos de cada materia y los comentarios de cada usuario.
- El acoplamiento de ambos componentes tanto *frontend* como *backend* permite comunicar los datos de usuario tales como las vistas, la información de los usuarios que provienen de los servidores.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Molina, S. Zamora, M. Diaz y E. Lagunes, «Aprendizaje electrónico: Puertas abiertas a la educación,» ecorfan, 2016. [En línea]. Available: https://www.ecorfan.org/booklets/Revista%20de%20Tecnologías%20de%20la%20Información_12%20PDF/Antonio%20Molina-Navarro,%20Sergio%20Zamora-Castro.pdf. [Último acceso: 02 Noviembre 2022].
- [2] N. Bohoquez, «Impacto de las tic en instituciones educativas,» uees, Septiembre 2017. [En línea]. Available: <http://repositorio.uees.edu.ec/bitstream/123456789/1978/1/Tesis%20-%20Bohórquez%20Cruz%20Nathaly.pdf>. [Último acceso: 02 Noviembre 2022].
- [3] E. Aguirre y B. Yupa, «Percepción de la educación virtual durante la covid-19,» ula, 03 Noviembre 2020. [En línea]. Available: <http://bdigital2.ula.ve:8080/xmlui/bitstream/handle/654321/7473/GICOS%204.pdf?sequence=1&isAllowed=y>. [Último acceso: 02 Noviembre 2022].
- [4] Z. Nieves, «La formación profesional en la universidad de hoy,» intercom, 2007. [En línea]. Available: http://web.intercom.es/jorgemas/teoria_mia_3.pdf. [Último acceso: 02 Noviembre 2022].
- [5] epn, «Proyectos de alto impacto que fueron premiados en la final del concurso INNOVA 2019,» epn, 2019. [En línea]. Available: <https://www.epn.edu.ec/proyectos-del-alto-impacto-fueron-premiados-en-la-final-del-concurso-innova-2019/>. [Último acceso: 02 Noviembre 2022].
- [6] EPN, «Boletín Microcosmos,» epn, [En línea]. Available: <https://www.epn.edu.ec/sinergia-ii-2022/>. [Último acceso: 02 Noviembre 2022].
- [7] coneduca, «El aprendizaje autónomo en la actualidad,» revistaedurama, 21 Febrero 2021. [En línea]. Available: <https://revistaedurama.com/el-aprendizaje-autonomo-en-la-actualidad/>. [Último acceso: 02 Noviembre 2022].
- [8] RedHat, «¿Qué es la metodología ágil?,» redhat, 19 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology#:~:text=En%20concreto%2C%20las%20metodologías%20ágiles,equipo%20para%20ofrecer%20mejoras%20constantes..> [Último acceso: 02 Noviembre 2022].
- [9] A. Requena, «10 diferencias entre Scrum y Kanban,» Openwebinars, 25 Diciembre 2018. [En línea]. Available: <https://openwebinars.net/blog/diferencias-scrum-kanban/>. [Último acceso: 11 Diciembre 2022].
- [10] Ilimit, «Metodología Scrum,» Ilimit, 18 Marzo 2022. [En línea]. Available: <https://www.ilimit.com/blog/metodologia-scrum/>. [Último acceso: 02 Noviembre 2022].
- [11] J. Warnimont, «Backend vs Frontend: ¿En Qué Se Diferencian?,» kinsta, 15 Junio 2022. [En línea]. Available: <https://kinsta.com/es/blog/backend-vs-frontend/>. [Último acceso: 02 Noviembre 2022].

- [12] I. de Souza, «Entiende las diferencias entre Front-End y Back-end,» rockcontent, 02 Marzo 2020. [En línea]. Available: <https://rockcontent.com/es/blog/front-end-y-back-end/>. [Último acceso: 02 Noviembre 2022].
- [13] MMA, «Todo lo que necesitás saber sobre backend,» mmaglobal, 06 Enero 2016. [En línea]. Available: <https://www.mmaglobal.com/news/todo-lo-que-necesitas-saber-sobre-backend-all-you-need-know-regarding-backend#:~:text=El%20Backend%2C%20tambi%C3%A9n%20conocido%20como,y%20devolverla%20al%20usuario%20final..> [Último acceso: 02 Noviembre 2022].
- [14] Tribalbyte, «React | Qué es, para qué sirve y cómo funciona |,» tribalite, 23 Junio 2021. [En línea]. Available: <https://tech.tribalyte.eu/blog-que-es-react>. [Último acceso: 02 Noviembre 2022].
- [15] K. Bracey, «¿Qué es Figma?,» 13 Mayo 2022. [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>. [Último acceso: 02 Noviembre 2022].
- [16] IBM, «Tipos de pruebas de software,» IBM, 16 Octubre 2020. [En línea]. Available: <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>. [Último acceso: 02 Noviembre 2022].
- [17] IBM, «ibm,» Despliegue de la aplicación web en un servidor web, 03 Marzo 2021. [En línea]. Available: <https://www.ibm.com/docs/es/spm/7.0.9?topic=system-deploying-your-web-application-web-server>. [Último acceso: 02 Noviembre 2022].
- [18] QuestionPro, «¿Qué es un estudio de caso y cómo realizarlo?,» questionpro, [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>. [Último acceso: 24 Noviembre 2022].
- [19] Santander Universidades, «Metodologías de desarrollo de software,» becas-santander, 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 24 Noviembre 2022].
- [20] proyectosagiles, «Qué es SCRUM,» proyectosagiles, [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 24 Noviembre 2022].
- [21] D. Molina, «Qué es un Product Owner,» iebsschool, 15 Diciembre 2021. [En línea]. Available: <https://www.iebschool.com/blog/diferencias-product-owner-product-manager-marketing-marketing-estrategico/#:~:text=Un%20Product%20Owner%20es%20el,producto%20desarrollado%20por%20una%20empresa..> [Último acceso: 24 Noviembre 2022].
- [22] P. Canal, «Definición y características del Scrum Master,» iebsschool, 12 Mayo 2022. [En línea]. Available: <https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/>. [Último acceso: 24 Noviembre 2022].

- [23] M. De Dios, «Scrum: qué es y cómo funciona este marco de trabajo,» wearemarketing, 09 Mayo 2022. [En línea]. Available: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#:~:text=Scrum%20es%20un%20framework%20que,valor%20posible%20productiva%20y%20creativamente..> [Último acceso: 24 Noviembre 2022].
- [24] O. Garcia, «Recopilación de requisitos,» proyectum, 01 Mayo 2013. [En línea]. Available: <https://www.proyectum.com/sistema/blog/recopilacion-de-requisitos/#:~:text=Basándonos%20en%20el%20PMBOK%2C%20la,con%20los%20objetivos%20del%20proyecto..> [Último acceso: 24 Noviembre 2022].
- [25] ScrumMexico, «Escribiendo Historias de Usuario,» scrum, 02 Agosto 2018. [En línea]. Available: <https://scrum.mx/informate/historias-de-usuario#queesunahdu=>. [Último acceso: 24 Noviembre 2022].
- [26] PYM, «Scrum: ¿Qué es el Product Backlog?,» programacionymas, [En línea]. Available: <https://programacionymas.com/blog/scrum-product-backlog>. [Último acceso: 24 Noviembre 2022].
- [27] M. Garcia , «¿Qué es el Sprint Backlog?,» ittude, 17 Julio 2020. [En línea]. Available: <https://ittude.com.ar/b/scrum/que-es-el-sprint-backlog/>. [Último acceso: 24 Noviembre 2022].
- [28] UOC, «¿Qué es una interfaz?,» uoc, [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>. [Último acceso: 24 Noviembre 2022].
- [29] P. Huet, «Arquitectura de software,» OpenWebinars, 24 Agosto 2022. [En línea]. Available: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>. [Último acceso: 26 Noviembre 2022].
- [30] desarrollowe6, «Qué es MVC,» desarrollowe6, 28 Julio 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html>. [Último acceso: 26 Noviembre 2022].
- [31] Next U, «¿Qué es React JS? ¿Cómo funciona?,» next_u, [En línea]. Available: <https://www.nextu.com/blog/que-es-react-js-como-funciona-rc22/>. [Último acceso: 26 Noviembre 2022].
- [32] B. Ramirez, «¿Qué es Vitejs? ¿Qué bueno tiene para ofrecer?,» Dev, 15 Julio 2021. [En línea]. Available: <https://dev.to/billiramirez/que-es-vitejs-que-bueno-tiene-para-ofrecer-nap#:~:text=Es%20una%20herramienta%20para%20Frontend,especifico%20o%20ya%20sea%20Vanillajs..> [Último acceso: 26 Noviembre 2022].
- [33] Guest Author, «Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo,» rockcontent, 12 Abril 2020. [En línea]. Available: <https://rockcontent.com/es/blog/bootstrap/>. [Último acceso: 27 Julio 2022].

- [34] J. Guaña, «La hoja de trucos de React Router,» freeCodeCamp, 02 Agosto 2021. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/la-hoja-de-trucos-de-react-router-todo-lo-que-necesitas-saber/>. [Último acceso: 26 Noviembre 2022].
- [35] axios, «¿Qué es Axios?,» axios-http, [En línea]. Available: <https://axios-http.com/es/docs/intro#:~:text=Axios%20es%20un%20Cliente%20HTTP,module%20nativo%20http%20de%20node..> [Último acceso: 26 Noviembre 2022].
- [36] A. Capparelli, «Qué son los hooks y cómo utilizarlos en tu proyecto con React,» Paradigma, 24 Febrero 2020. [En línea]. Available: <https://www.paradigmadigital.com/dev/hooks-como-utilizarlos-react/>. [Último acceso: 26 Noviembre 2022].
- [37] M. Rondón, «medium,» Cypress: ¿Qué es?, 10 Diciembre 2019. [En línea]. Available: <https://medium.com/get-on-board-dev/cypress-qué-es-por-qué-y-cómo-lo-usamos-en-get-on-board-17688b453fdc>. [Último acceso: 15 Enero 2023].

ANEXOS

Aquí se indica cada uno de los Anexos utilizados para el desarrollo del sistema EDUPOLI, los cuales esta conformados por:

- **ANEXO I.** Resultado de originalidad del proyecto.
- **ANEXO II.** Manual Técnico.
- **ANEXO III.** Manual de Usuario.
- **ANEXO IV.** Manual de Instalación.

ANEXO I

A continuación, se presenta el certificado que el director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta anti-plagio Turnitin.



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 24 de febrero de 2023

De mi consideración:

Yo, MAYRA ISABEL ALVAREZ JIMÉNEZ, en calidad de directora del Trabajo de Integración Curricular componente FRONTEND, titulado "DESARROLLO DE UN SISTEMA WEB QUE SIRVA COMO GUÍA DE ESTUDIO PARA ESTUDIANTES DE LA ESFOT", elaborado por el estudiante STEVEN ADRIAN CHICAIZA UGSHA de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



MAYRA ISABEL
ALVAREZ JIMÉNEZ

Ing. Mayra Alvarez MSc.
Técnico docente EPN - ESFOT

ANEXO II

A continuación, se presentan los distintos resultados obtenidos al desarrollar el componente *frontend*, además de los requisitos que se necesitaron para realizarlo, como son historias de usuario, recopilación de requerimientos, el *mockup*, pruebas del componente, entre otros.

Recopilación de requerimientos

La **Tabla XI** indica los requerimientos a cumplir al inicio del proyecto en base a lo solicitado por el cliente.

Tabla XI: Recopilación de requerimientos.

RECOPIACION DE REQUERIMIENTOS	
ID – RR	ENUNCIADO DEL ITEM
RR001	El usuario con perfil administrador y estudiante necesitan consumir varios métodos para: <ul style="list-style-type: none">• Iniciar sesión• Cerrar sesión• Recuperar contraseña
RR002	El usuario con perfil estudiante necesita consumir un método para: <ul style="list-style-type: none">• Registrarse a través de un formulario.
RR003	El usuario con perfil administrador y estudiante necesita consumir varios métodos para: <ul style="list-style-type: none">• Modificar perfil de usuario.
RR004	El usuario con perfil administrador necesita consumir varios métodos para: <ul style="list-style-type: none">• Gestionar carreras y semestres
RR005	El usuario con perfil estudiante necesita consumir un método para: <ul style="list-style-type: none">• Visualizar las carreras y semestres
RR006	El usuario con perfil administrador y estudiante necesita consumir un método para: <ul style="list-style-type: none">• Búsqueda por materias de la carrera
RR007	El usuario con perfil estudiante y administrador necesita consumir un método para: <ul style="list-style-type: none">• Visualizar su perfil
RR008	El usuario con perfil estudiante utiliza un método para: <ul style="list-style-type: none">• Visualizar el contenido educativo de cada semestre por carrera.
RR009	El usuario con perfil administrador necesita consumir varios métodos para: <ul style="list-style-type: none">• Gestionar el contenido educativo
RR010	El usuario con perfil estudiante y administrador utilizaran un método para: <ul style="list-style-type: none">• Gestionar las sugerencias e inquietudes (comentarios/foro).
RR011	El usuario con perfil estudiante requiere de métodos para: <ul style="list-style-type: none">• Sección de preguntas frecuentes en la vista del <i>Dashboard</i>

RR012	El usuario administrador requiere validar y verificar los componentes y métodos del sistema web, además del despliegue del componente <i>frontend</i> .
--------------	---

Historias de Usuario

Culminada la recolección de requerimientos, continuamos con las Historias de usuarios. Por tanto, se presentan las 12 Historias de usuario basadas en los requerimientos del cliente que dan inicio de la **TABLA XI** a la **TABLA XXI**.

Tabla XI: Registrarse en un formulario

HISTORIA DE USUARIO	
Identificador: HU002	Usuario: Estudiante
Nombre historia: Registrar usuarios a través de un formulario.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente estudiante puede registrarse al llenar los siguientes campos: <ul style="list-style-type: none"> • Nombre • Apellido • Correo electrónico • Contraseña • Confirmar Contraseña 	
Observación: El cliente Estudiante, debe ingresar sus datos correctamente para registrarse, y así poder consumir la información del sistema web.	

Tabla XII: Modificar perfil de usuario

HISTORIA DE USUARIO	
Identificador: HU003	Usuario: Administrador y estudiante
Nombre historia: Modificar perfil de usuario	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante hacen uso de métodos para modificar y visualizar su perfil que consta de: <ul style="list-style-type: none"> • Perfil • Nombre • Imagen de perfil • Entre otros. 	
Observación: El cliente administrador y estudiante pueden editar su perfil solo cuando estén logueados.	

Tabla XIII: Gestión de carreras y semestres

HISTORIA DE USUARIO	
Identificador: HU004	Usuario: Administrador
Nombre historia: Gestionar carreras y semestres	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador hace uso de métodos que gestionan las carreras y semestres con acciones como: <ul style="list-style-type: none"> • Crear • Eliminar • Editar • Leer 	
Observación: El cliente administrador puede realizar acciones de CRUD dentro del sistema.	

Tabla XIV: Visualizar las carreras y semestres

HISTORIA DE USUARIO	
Identificador: HU005	Usuario: Administrador y Estudiante
Nombre historia: Visualizar las carreras y semestres	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 1	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante pueden observar las carreras y semestres en la ventana principal.	
Observación: Los distintos usuarios tienen acceso a visualizar los contenidos de cada carrera con sus respectivos semestres.	

Tabla XV: Búsqueda

HISTORIA DE USUARIO	
Identificador: HU006	Usuario: Administrador y estudiante
Nombre historia: Buscar contenido por materias	
Prioridad en Negocio: Baja	Riesgo en Negocio: Baja
Iteración asignada: 3	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante pueden realizar búsquedas de asignaturas de una carrera.	
Observación: Los dos tipos de perfiles tienen la posibilidad de utilizar el buscador para encontrar más fácil los temas deseados.	

Tabla XVI: Visualizar perfil

HISTORIA DE USUARIO	
Identificador: HU007	Usuario: Administrador y Estudiante
Nombre historia: Visualizar perfil	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante pueden visualizar las características de su perfil.	
Observación: Los dos tipos de usuarios tienen la posibilidad de observar las características de su perfil en una pequeña sección.	

Tabla XVII: Visualizar el contenido

HISTORIA DE USUARIO	
Identificador: HU008	Usuario: Estudiante
Nombre historia: Visualizar el contenido educativo de cada semestre.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante pueden visualizar los contenidos de cada materia en una sección.	
Observación: Los dos tipos de usuarios tienen la posibilidad de observar los recursos académicos dentro del sistema.	

Tabla XVIII: Gestión del contenido

HISTORIA DE USUARIO	
Identificador: HU009	Usuario: Administrador
Nombre historia: Gestionar el contenido educativo	
Prioridad en Negocio: Medio	Riesgo en Negocio: Medio
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador hace uso de métodos para gestionar el contenido educativo con acciones como: <ul style="list-style-type: none"> • Crear • Eliminar • Editar • Leer 	
Observación: El cliente administrador puede realizar acciones de CRUD dentro del sistema.	

Tabla XIX: Gestión de sugerencia e inquietudes

HISTORIA DE USUARIO	
Identificador: HU010	Usuario: Administrador y estudiante
Nombre historia: Gestionar las sugerencias e inquietudes.	
Prioridad en Negocio: Bajo	Riesgo en Negocio: Bajo
Iteración asignada: 1	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante hacen uso de métodos para gestionar las sugerencias e inquietudes con acciones de: <ul style="list-style-type: none"> - Crear - Eliminar - Editar - Leer 	
Observación: El usuario administrador y estudiante pueden realizar acciones de CRUD para las sugerencias e inquietudes.	

Tabla XX: Sección de preguntas frecuentes en la vista del *Dashboard*

HISTORIA DE USUARIO	
Identificador: HU011	Usuario: Estudiante y Administrador
Nombre historia: Mostrar sección de preguntas frecuentes en la vista del <i>Dashboard</i>	
Prioridad en Negocio: Medio	Riesgo en Negocio: Medio
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: El cliente administrador y estudiante hacen uso de métodos para revisar la sección de preguntas frecuentes con un botón en la vista del <i>Dashboard</i> .	
Observación: El cliente administrador puede realizar acciones de CRUD dentro del sistema.	

Tabla XXI: Pruebas y despliegue del componente *frontend*

HISTORIA DE USUARIO	
Identificador: HU012	Usuario: Estudiante y Administrador
Nombre historia: Probar y desplegar el componente <i>frontend</i>	
Prioridad en Negocio: Medio	Riesgo en Negocio: Medio
Iteración asignada: 2	
Responsable: Adrian Chicaiza	
Descripción: Los usuarios administrador y estudiante, necesitan verificar y validar los métodos y componentes de manera que se realizan pruebas de compatibilidad, de carga y unitarias. Además, consumir el sistema en cualquier dispositivo o navegador, es por ello que se requiere el despliegue del <i>frontend</i> .	
Observación: El código a probar no debe ser modificado en la etapa de pruebas y despliegue.	

Product Backlog

La **TABLA XXII** indica en orden la importancia de cada módulo a realizar en el componente *frontend*.

Tabla XXII: Elaboración del Product Backlog

ELABORACION DEL PRODUCT BACKLOG				
ID-HU	HISTORIA DE USUARIO	ITERACION	ESTADO	PRIORIDAD
HU001	Iniciar sesión, cerrar sesión y recuperar contraseña.	1	Finalizado	Alta
HU002	Registrar usuarios través de un formulario.	2	Finalizado	Media
HU003	Modificar perfil de usuario	2	Finalizado	Media
HU004	Gestionar carreras y semestres	2	Finalizado	Alta
HU005	Visualizar las carreras y semestres	1	Finalizado	Alta
HU006	Buscar por materia	3	Finalizado	Baja
HU007	Visualizar su perfil	2	Finalizado	Alta
HU008	Visualizar el contenido educativo de cada semestre	2	Finalizado	Alta
HU009	Gestionar el contenido educativo	2	Finalizado	Medio
HU010	Gestionar las sugerencias e inquietudes en el foro.	1	Finalizado	Bajo
HU011	Mostrar sección de preguntas frecuentes en la vista del Dashboard	2	Finalizado	Media
HU012	Probar y desplegar el <i>frontend</i>	2	Finalizado	Media

Sprint Backlog

La **TABLA XXIII** muestra todos los *Sprints* a realizar según lo planificado en los requerimientos.

Tabla XXIII: Elaboración del Sprint Backlog

ELABORACIÓN DEL SPRINT BACKLOG						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo	----	----	-----	<ul style="list-style-type: none"> - Definir los requerimientos funcionales y no funcionales para el componente <i>frontend</i> - Configuración de herramientas - Configuración del ambiente de desarrollo 	20 h
SB001	Iniciar sesión y recuperar contraseña del usuario	Módulo – Inicio de Sesión rol Estudiante	HU001	<p>Iniciar sesión, cerrar sesión y recuperar contraseña</p>	<ul style="list-style-type: none"> - Definir los campos obligatorios para iniciar sesión y recuperar contraseña - Diseño de interfaz para el inicio de sesión y recuperar contraseña. - Crear las funciones para el inicio de sesión y recuperar contraseña del usuario estudiante en el <i>frontend</i>. - Validación de los campos en iniciar sesión y recuperar contraseña. - Consumir varios métodos para iniciar sesión como estudiante. 	35h
		Módulo – Inicio de Sesión rol administrador		<p>Iniciar sesión, cerrar sesión y recuperar contraseña</p>	<ul style="list-style-type: none"> - Diseño de prototipos. - Definir los campos obligatorios para iniciar sesión y recuperar contraseña - Diseño de interfaz para el inicio de sesión y recuperar contraseña. - Crear las funciones para el inicio de sesión y recuperar contraseña del usuario administrador en el <i>frontend</i>. - Validación de los campos en iniciar sesión y recuperar contraseña. 	

					- Consumir varios métodos para iniciar sesión como administrador.	
		Prueba unitaria				
SB0 02	Registrar usuario y editar perfil de usuario	Módulo – Registro de usuario estudiante	HU0 02	Registrar se a través de un formulario.	-Definir campos obligatorios para el inicio de sesión. -Diseño de la interfaz para el registro. -Creación de funcionalidades para registrar al estudiante - Validación de campos de registro. -Consumo de métodos para registrar estudiante.	35h
		Módulo – Modificar perfil	HU0 03	Modificar perfil de usuario	- Definir los campos que se mostraran en el módulo - Definir los campos que serán editables. - Consumir varios métodos para modificar el perfil.	
		Módulo – Visualizar usuario	HU0 07	Visualizar su perfil	- Diseño de prototipos. - Consumir varios métodos para visualizar el perfil. - Diseño de interfaz para el perfil del usuario.	
		Prueba unitaria				
SB0 03	Gestionar carreras y visualizar información Visualizar información	Módulo – Gestionar carreras	HU0 04	Gestionar carreras y semestres	- Definir los campos que se mostraran en el módulo modal. - Definir los campos que serán editables. -Establecer validaciones.	45h
		Módulo – Visualización de carreras	HU0 05	Visualizar las carreras y semestres	- Diseño de prototipos. - Diseño de interfaz para el perfil del usuario. - Consumir varios métodos para visualizar la información.	
	Prueba unitaria					
SB0 04	Gestión y visualización de documentos educativos Visualización de materiales educativos	Módulo – Gestión de material educativo y materias	HU0 09	Gestionar el contenido educativo	- Definir los campos que se mostraran en el módulo - Definir los campos que serán editables - Consumir varios métodos para la gestión de los documentos -Establecer validaciones	40h
		Módulo – Visualización de documentos y materias	HU0 08	Visualizar el contenido educativo	- Diseño de <i>interfaz</i> . - Consumir varios métodos para visualizar la información.	

				de cada semestre		
		Prueba unitaria				
SB005	Búsqueda de materias y gestión de inquietudes Gestión de inquietudes	Módulo – Búsqueda de materias	HU006	Búsqueda	- Utilización de función de búsqueda del <i>frontend</i> . - Diseño de interfaz.	35h
		Módulo – Gestión de sección de inquietudes	HU010	Gestionar las sugerencias e inquietudes.	-Definir el campo -Diseño de la interfaz para las inquietudes -Creación de funcionalidades -Consumo de métodos para inquietudes	
		Módulo – Sección de preguntas frecuentes	HU011	Preguntas frecuentes	- Diseño de la Interfaz - Utilización de métodos para las preguntas frecuentes	
	Prueba unitaria					
SB06	Pruebas y despliegue <i>frontend</i>	Pruebas y despliegue del <i>frontend</i>	HU012	- Pruebas Unitarias - Pruebas de carga - Pruebas de compatibilidad - Despliegue en <i>Vercel</i>	30h	
Total					240	

Diseño de Interfaces

Enseguida se muestra el *mockup* de las pantallas que se realizaron para el componente *frontend*. Las cuales abarcan de la **Fig. 51** a la **Fig. 58**

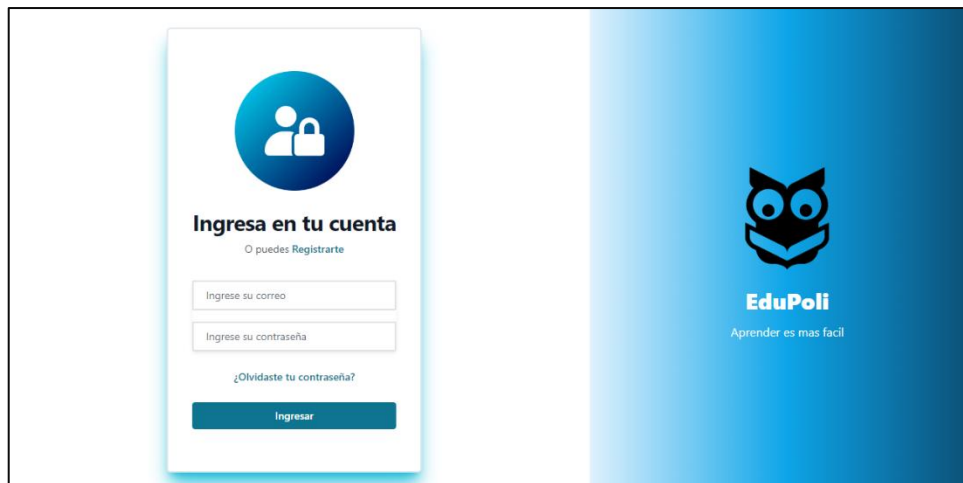


Fig. 51 Vista de inicio sesión del sistema *web*

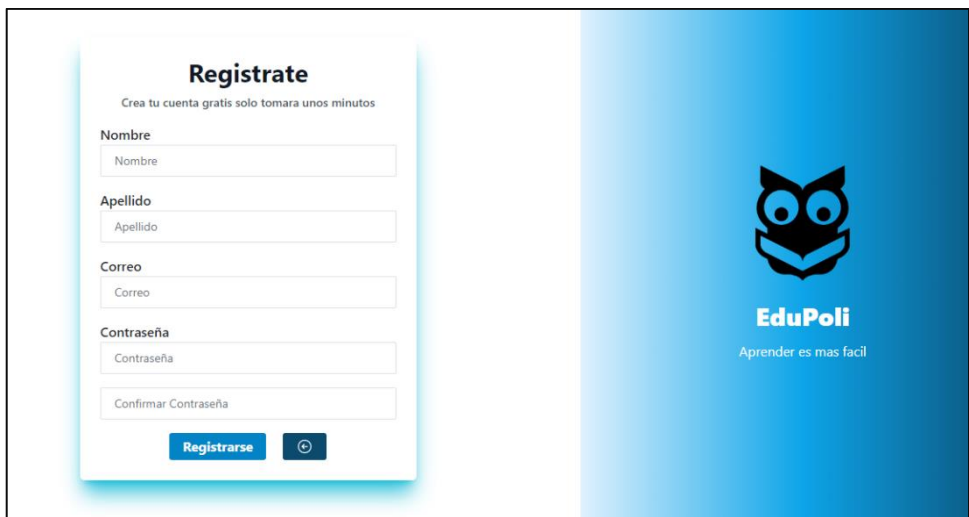


Fig. 52 Vista de registro

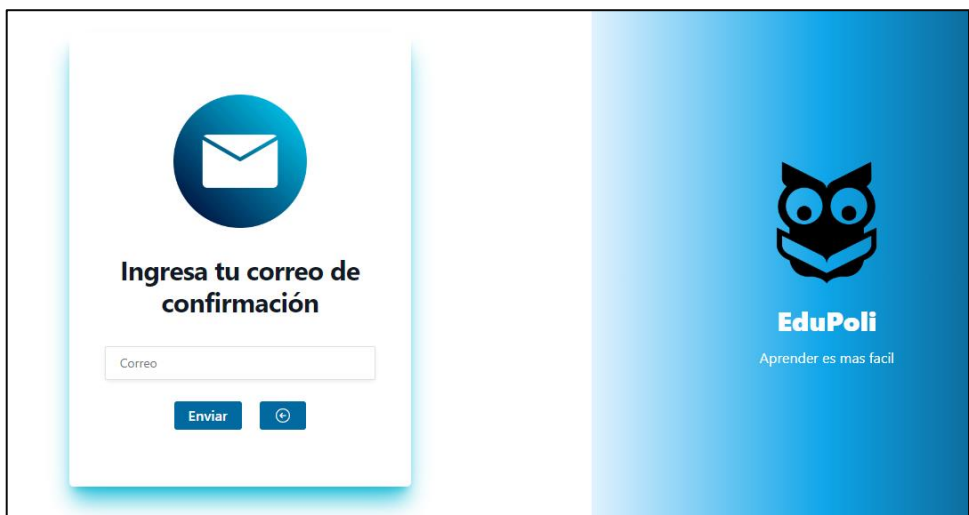


Fig. 53 Prototipo de página Recuperar contraseña

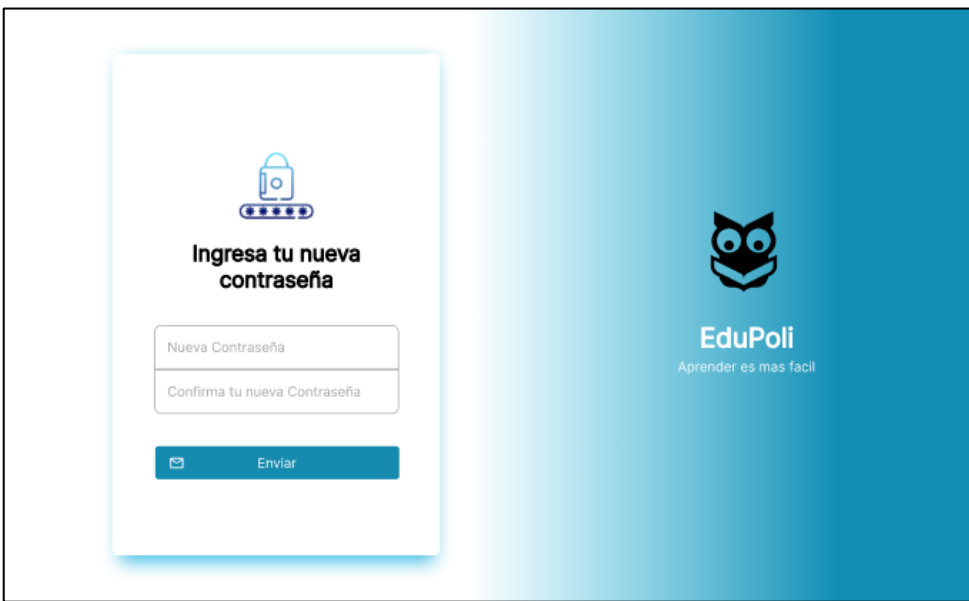


Fig. 54 Vista Cambio de Contraseña

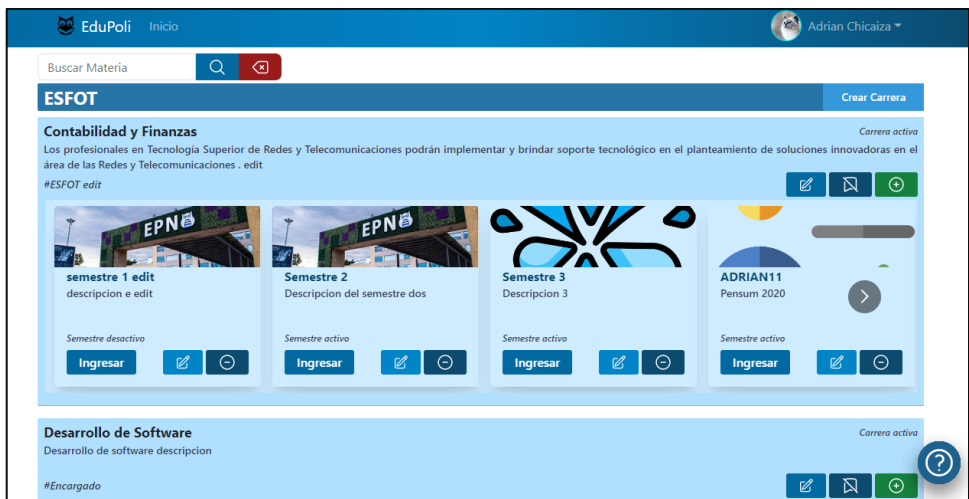


Fig. 55 Vista de página *Dashboard*

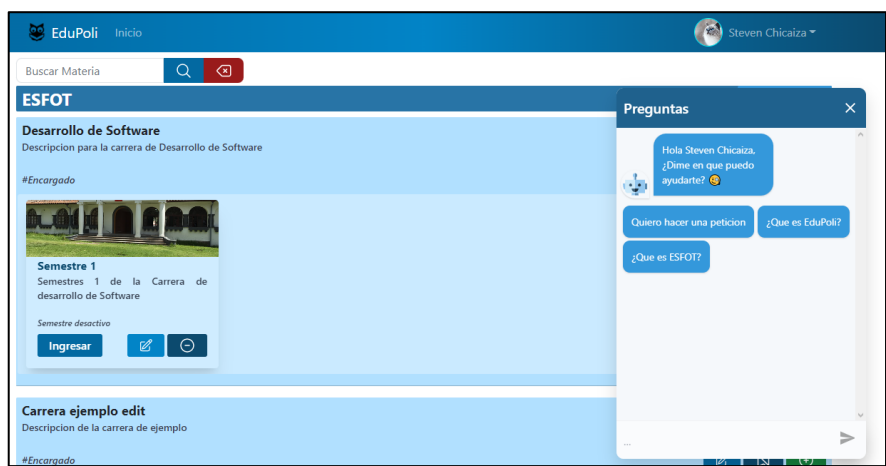


Fig. 56 Vista de preguntas frecuentes

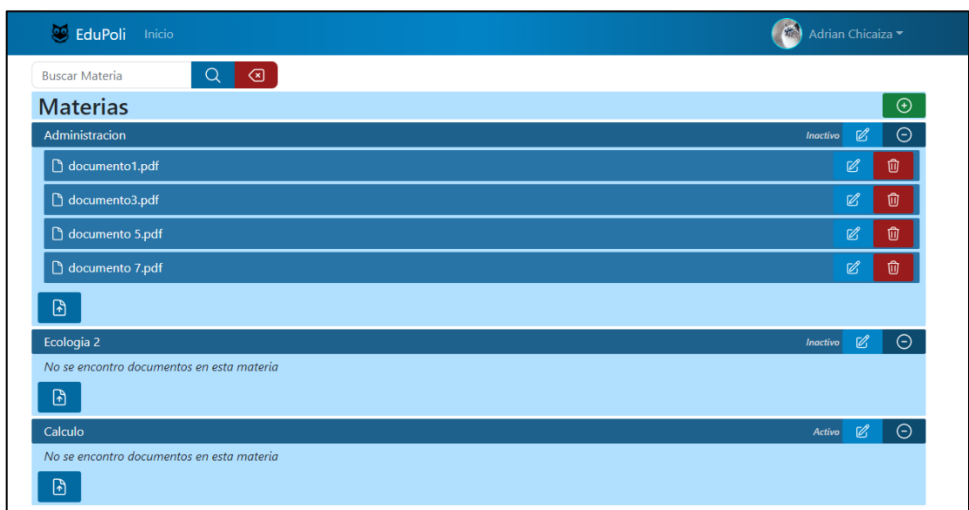


Fig. 57 Vista de página *Semestre*

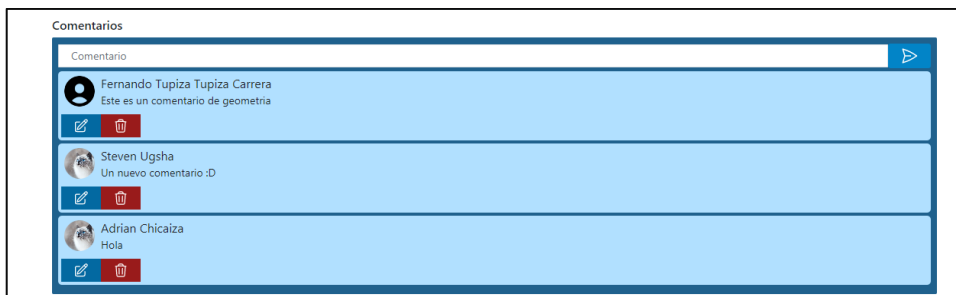


Fig. 58 Vista de página comentarios

Pruebas del sistema y despliegue

Una terminado nuestro componente *frontend* procedemos a la realización de los test de las vistas y componentes para la comprobación de su correcto funcionamiento.

Pruebas unitarias

La realización de las pruebas unitarias se facilitó con la herramienta *Cypress* la cual proporciona una página de *test* en donde podemos ver en tiempo real la ejecución del *test* y las peticiones que se han utilizado para poder realizar las acciones en cada uno de los módulos y vistas de nuestro componente *frontend*, a continuación, se presentan las pruebas unitarias y sus resultados de cada una de las vistas:

En la **Fig. 59** se muestra el comando para la instalación de nuestra herramienta de *Cypress*, que instala dependencias dentro de nuestro proyecto, el cual genera una carpeta que se encarga de reconocer la lógica de cada una de las vistas y componentes que realizamos para facilitar el trabajo y la realización de los *test*, luego ejecutamos el comando `npx cypress open` y nos mostrara una ventana como en la **Fig. 60** y configuramos nuestro entorno de pruebas, si la configuración ha sido exitosa se presentará un mensaje como indica la **Fig. 61**.

```
PS C:\Users\Adrian\Documents\GitHub\EDUPOLI-AC> npm install cypress --save-dev
[#####] / reify:nan: timing metavuln:calculate:security-advisory:glob-parent:1089029 Completed in 190ms
```

Fig. 59 Instalación Cypress

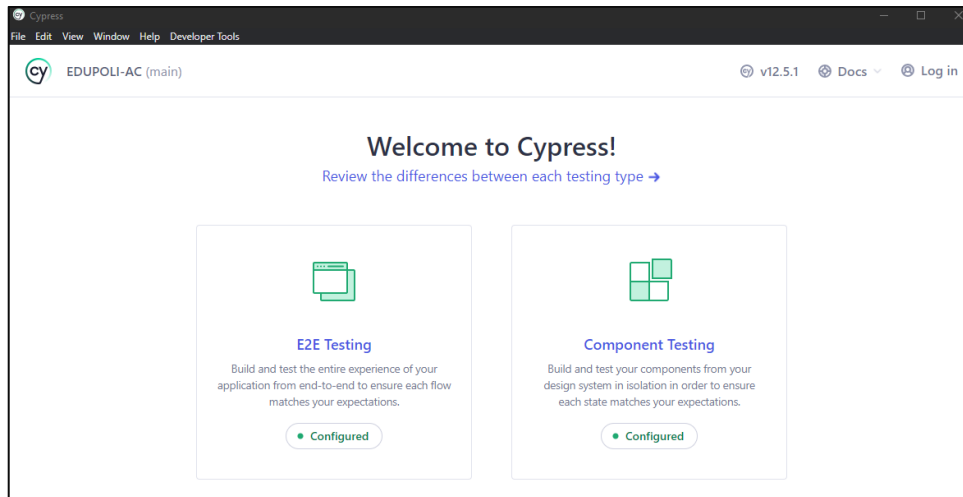


Fig. 60 Ventana configuración

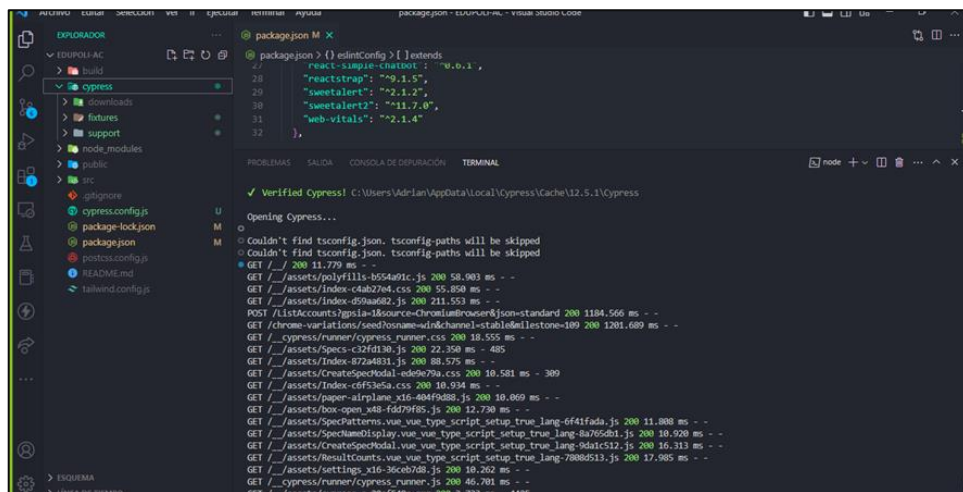


Fig. 61 Resultado de la configuración exitosa

Una vez instalado y configurado nuestro entorno de pruebas, realizamos las pruebas de cada una de las vistas con sus respectivos resultados.

A continuación, se presenta la función para el *login* mediante Cypress el cual proporciona una función de ejemplo el cual se basa en el flujo de la vista, de manera que se establece en donde se quiere realizar la prueba en la línea 3, y posteriormente establecemos que campos llenamos y con que valores como en la línea 4 y 5, por último se realiza la acción de dar clic en el botón que contenga el nombre Ingresar como en la línea 6, todo esto en resumen se muestra en la Fig. 62 y los resultados de las peticiones para la vista como indica la Fig. 63.

```

cypress > e2e > plogin.cy.js > ...
1 describe("Inicio de sesion", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7   });
8 });

```

Fig. 62 Prueba unitaria del Dashboard

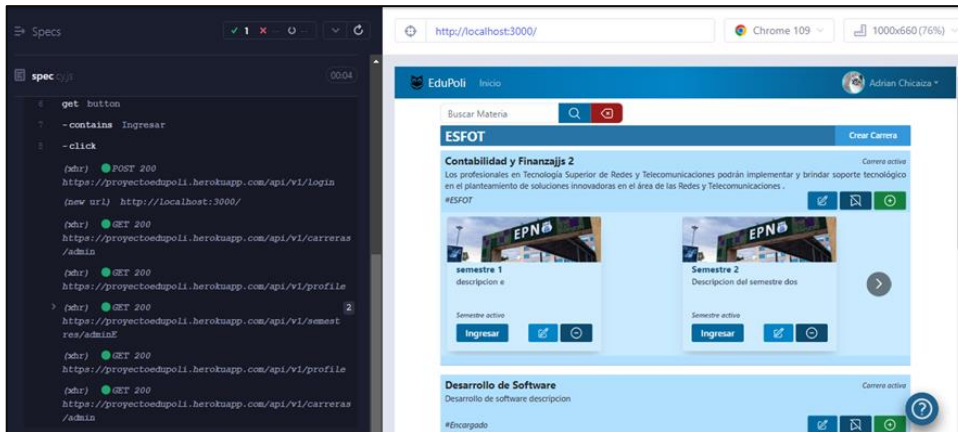


Fig. 63 Resultado de las peticiones para vista Dashboard

A continuación, se muestra el código de prueba unitaria para la creación de una carrera, en donde se establece los campos y valores para ingresar y crear la carrera de ejemplo como indica la Fig. 64 y sus resultados en la Fig. 65.

```

cypress > e2e > pCrearCarrera.cy.js > ...
1 describe("Crear Carrera", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Crear una Carrera
9     cy.get("button").contains("Crear Carrera").click({ force: true }) //aquí abre el modal
10    cy.get('input[name="nombre"]').type("Agua y Saneamiento Ambiental")
11    cy.get('input[name="encargado"]').type("Encargado")
12    cy.get('textarea[name="descripcion"]').type("Descripcion para la carrera de Agua y Saneamiento Ambiental")
13    cy.contains("div","Cerrar").contains("Crear").click();
14  });
15 });

```

Fig. 64 Prueba unitaria para creación de carrera

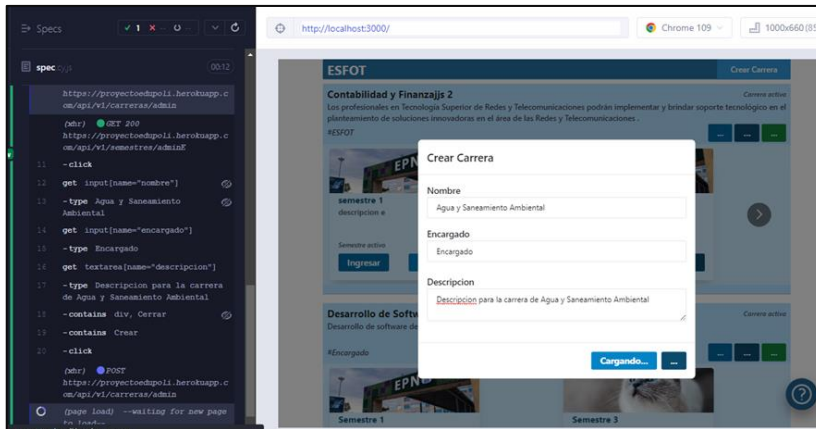


Fig. 65 Resultado de las peticiones para creación de carrera

Prueba unitaria para la edición de una carrera con valores de cambio como se muestra en la **Fig. 66** y los resultados de las peticiones en la **Fig. 67**.

```

cypress > e2e > pEditarCarrera.cy.js > ...
1 describe("Editar Carrera", () => {
2   it("pases", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Editar Carrera
9     cy.get('button[id="EditCarr"]').first().click()
10    cy.get('input[name="nombre"]').type(" edit")
11    cy.get('input[name="encargado"]').type(" edit")
12    cy.get('textarea[name="descripcion"]').type(" edit")
13    cy.get('button[id="ActualizarCarrera"]').click()
14    cy.get("button").contains("Confirmar").click()
15
16  });
17 });

```

Fig. 66 Prueba unitaria para actualización de carreras

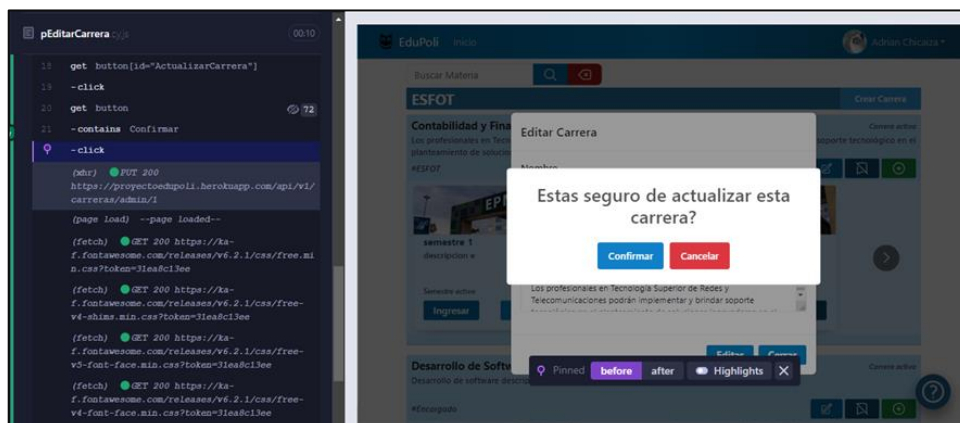


Fig. 67 Resultado de las peticiones para actualización de carrera

Prueba unitaria para cambiar el estado de una carrera mediante el código de la **Fig. 68** y también los resultados de las peticiones para realizar dicha acción en la **Fig. 69**.

```

cypress > e2e > pDesactivarCarrera.cy.js > ...
1 describe("Editar Carrera", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Desactivar Carrera
9     cy.get('button[id="DesactivarCarrera"]').first().click()
10    cy.get("button").contains("Confirmar").click()
11
12    });
13  });

```

Fig. 68 Prueba unitaria para cambiar estado de la carrera

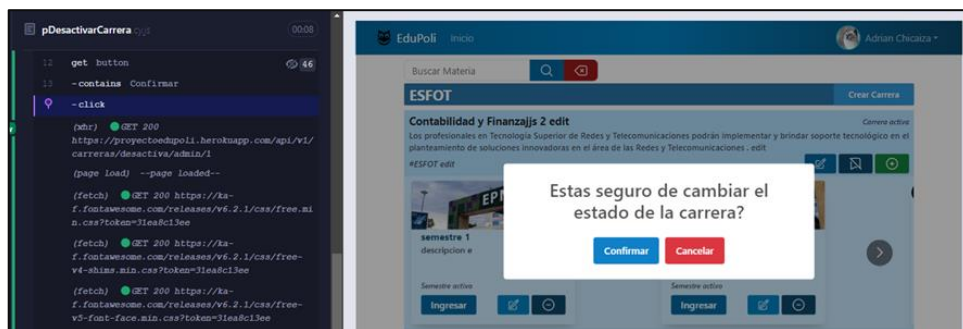


Fig. 69 Resultado de peticiones al cambiar estado de la carrera

Prueba unitaria para la creación de un nuevo semestre de ejemplo en base a sus inputs y con valores enviados desde el código de la prueba unitaria como se muestra en la **Fig. 70** y sus resultados de peticiones en la **Fig. 71**.

```

cypress > e2e > pCrearSemestre.cy.js > ...
1 describe("Crear Semestre", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Crear Semestre
9     cy.get('button[id="CrearSem"]').first().click()
10    cy.get('input[type="file"]').selectFile('cypress/e2e/esfot.png')
11    cy.get('input[name="nombre"]').type("semestre 1")
12    cy.get('textarea[name="descripcion"]').type("Descripcion para el semestre")
13    cy.get('button[id="CrearSemes"]').click()
14
15    });
16  });

```

Fig. 70 Prueba unitaria para la creación de semestre

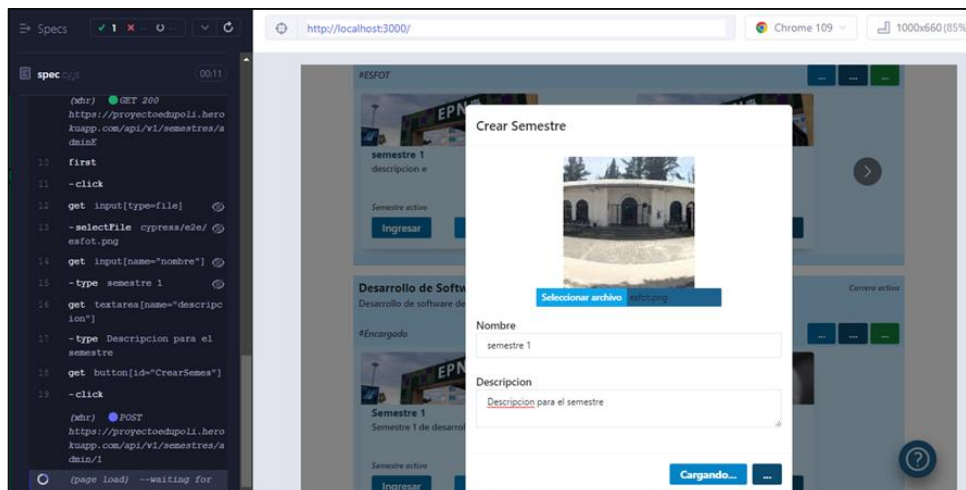


Fig. 71 Resultado de peticiones al cambiar estado de la carrera

Prueba unitaria para la creación de un semestre con valores de prueba como se indica en la Fig. 72 y los resultados de las peticiones en la Fig. 73.

```

cypress > e2e > pEditarSemestre.cy.js > ...
1 describe("Editar Semestre", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Editar Semestre
9     cy.get('button[id="EditarSemestre"]').first().click()
10    cy.get('input[name="nombre"]').type(" edit")
11    cy.get('textarea[name="descripcion"]').type(" edit")
12    cy.get('button[id="EditarSemSelect"]').click()
13    cy.get("button").contains("Confirmar").click()
14  });
15 });

```

Fig. 72 Prueba unitaria para creación de semestre

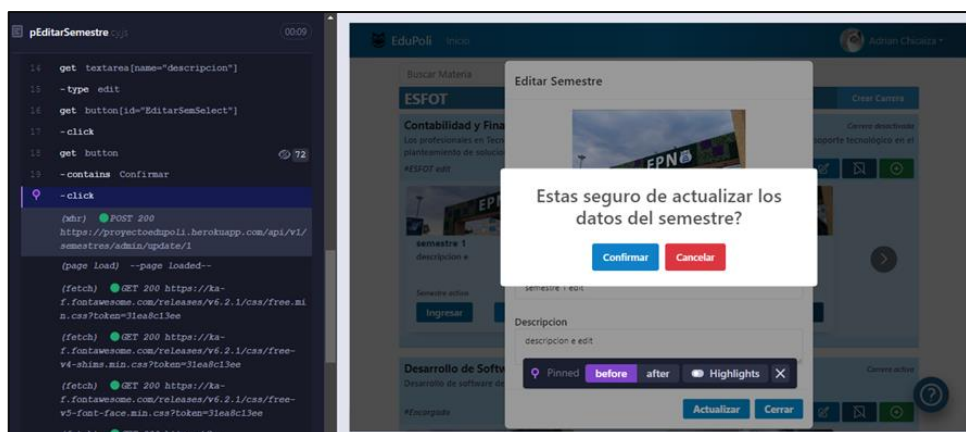


Fig. 73 Resultado de peticiones al crear semestre

Prueba unitaria para cambiar el estado de un semestre mediante las acciones del código de prueba unitaria de la Fig. 74 y su resultado de las peticiones en la Fig. 75.

```

cypress > e2e > pDesactivarSemestre.cy.js > ...
1 describe("Desactivar Semestre", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Desactivar Semestre
9     cy.get('button[id="EstadoSemestre"]').first().click()
10    cy.get("button").contains("Confirmar").click()
11  });
12 });
13

```

Fig. 74 Prueba unitaria para cambiar estado semestre

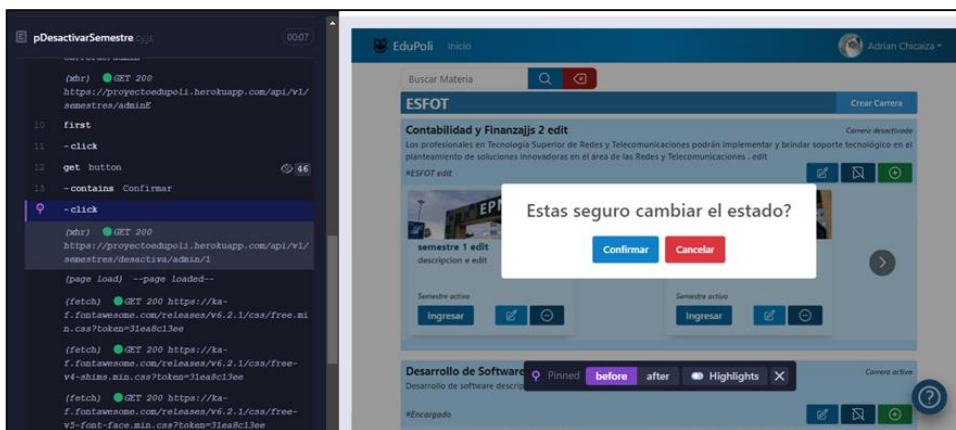


Fig. 75 Resultado de peticiones al cambiar estado del semestre

Prueba unitaria para la vista de las materias en la vista de semestres con el código de prueba en la Fig. 76, además de los resultados de las peticiones en la Fig. 77.

```

4 describe("Inicio de sesion", () => {
5   it("passes", () => {
6     cy.visit("http://localhost:3000")
7     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
8     cy.get('input[name="password"]').type("Adrian123@")
9     cy.get("button").contains("Ingresar").click()
10
11     //VerMaterias
12     cy.get('button[id="VerMaterias"]').first().click()
13

```

Fig. 76 Prueba unitaria ver las materias

```

(new url) http://localhost:3000/semestre/1

(xhr) GET 200 https://proyectoedupoli.herokuapp.com/api/v1/materias/admin

(xhr) GET 200
https://proyectoedupoli.herokuapp.com/api/v1/documentos/admin

(xhr) GET 200 https://proyectoedupoli.herokuapp.com/api/v1/usuarios

(xhr) GET 200
https://proyectoedupoli.herokuapp.com/api/v1/comentarios/sistema

(xhr) GET 200
https://proyectoedupoli.herokuapp.com/api/v1/documentos/admin

(xhr) GET 200 https://proyectoedupoli.herokuapp.com/api/v1/materias/admin

(xhr) GET 200
https://proyectoedupoli.herokuapp.com/api/v1/comentarios/sistema

(xhr) GET 200 https://proyectoedupoli.herokuapp.com/api/v1/usuarios

```

Fig. 77 Resultado de peticiones al ver materias

Prueba unitaria para la creación de una materia mediante el código de la Fig. 78 y también los resultados de las peticiones en la Fig. 79.

```

cypress > e2e > pCrearMateria.cy.js > ...
1 describe("Crear Materia", () => {
2   it("pases", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Ver y crear una materia
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="CrearMat"]').click()
11    cy.get('input[name="nombre"]').type("Calculo")
12    cy.get('button[id="CrearMatSelect"]').first().click()
13  });
14 });

```

Fig. 78 Prueba unitaria para crear una materia

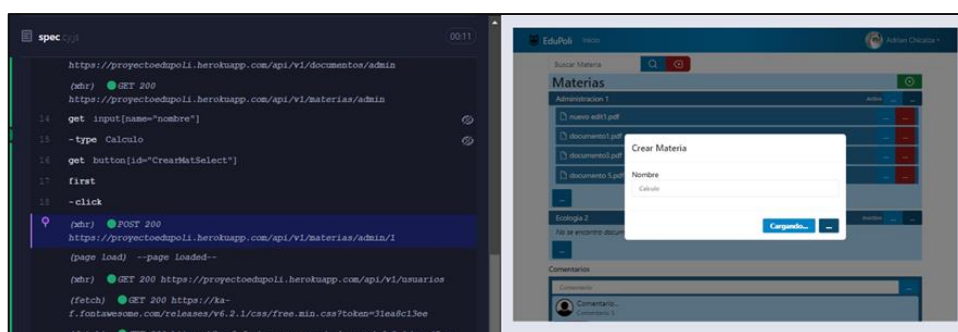


Fig. 79 Resultado de peticiones al crear una materia

Prueba unitaria para la edición de una materia con valores de predeterminados como indica la Fig. 80 y los resultados de las peticiones en la Fig. 81.

```

cypress > e2e > pEditarMateria.cy.js > ...
1 describe("Editar Materia", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Editar una materia
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="EditMateria"]').first().click()
11    cy.get('input[name="nombre"]').type(" edit")
12    cy.get('button[id="EditMateriaSelect"]').first().click()
13    cy.get("button").contains("Confirmar").click()
14
15  });
16 });
17

```

Fig. 80 Prueba unitaria para actualizar una materia

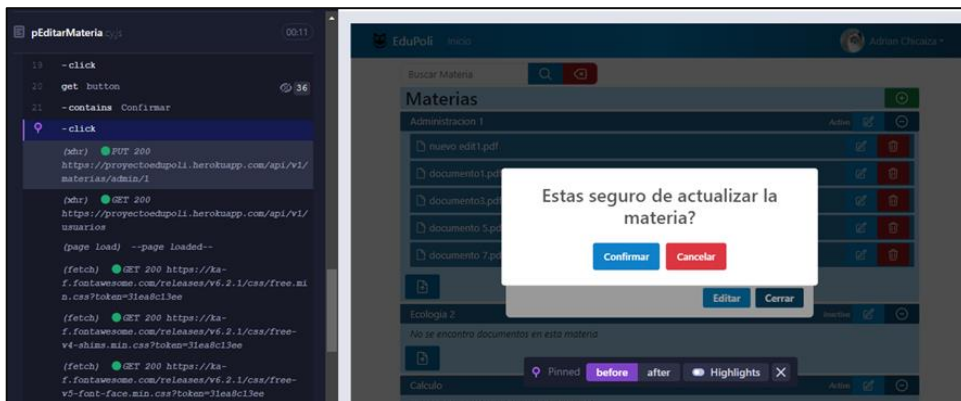


Fig. 81 Resultado de peticiones al actualizar materia

Prueba unitaria para desactivar una materia dependiendo del código de prueba unitaria de la Fig. 82 y sus resultados de las peticiones en la Fig. 83.

```

cypress > e2e > pDesactivarMateria.cy.js > ...
1 describe("Desactivar/Activar Materia", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Desactivar Materia
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="DesactivarMat"]').first().click()
11    cy.get("button").contains("Confirmar").click()
12
13  });
14 });
15

```

Fig. 82 Prueba unitaria para cambiar estado de materia

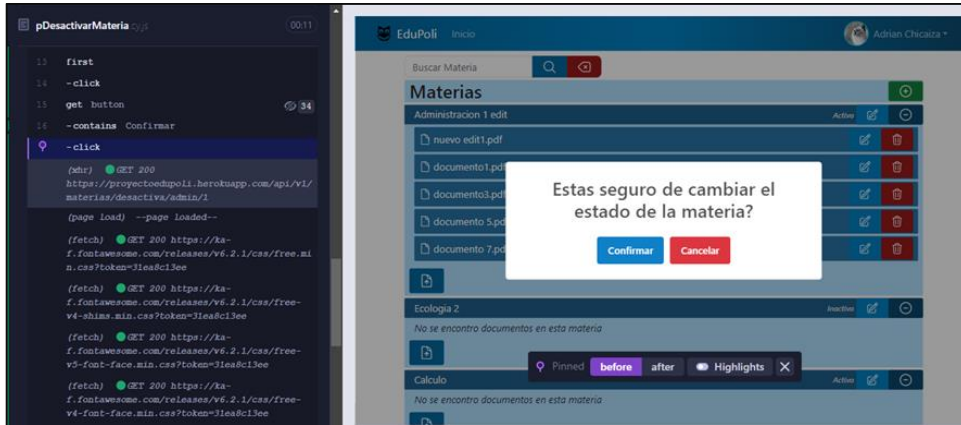


Fig. 83 Resultado de peticiones para cambiar estado de la materia

Prueba unitaria para subir un documento de ejemplo como indica la **Fig. 84**, con datos predeterminados en los inputs, además de los resultados de las peticiones en la **Fig. 85**.



Fig. 84 Prueba unitaria para subir documento

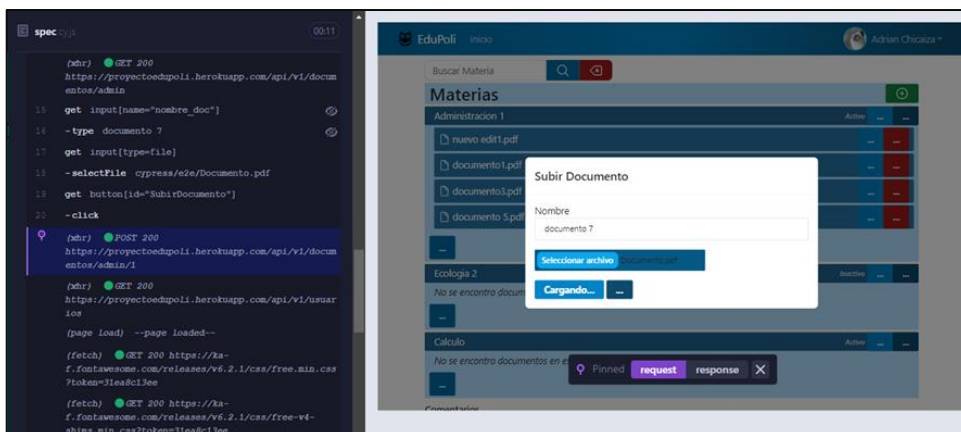


Fig. 85 Resultado de peticiones al subir documento

Prueba unitaria para editar el documento de una carrera como se indica en la **Fig. 86**, se establecen campos de prueba y valores para cambiar el nombre del documento y en la **Fig. 87** se indica el tanteo de las peticiones para realizar dicha acción.

```

cypress > e2e > pEditarDocumento.cy.js > ...
1 describe("Editar Documento", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // Editar documento
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="EditDocSelec"]').first().click()
11    cy.get('input[name="nombre_doc"]').type(" edit")
12    cy.get('button[id="EditarDocconfirm"]').click()
13    cy.get("button").contains("Confirmar").click()
14
15  });
16 });
17

```

Fig. 86 Prueba unitaria al editar un documento

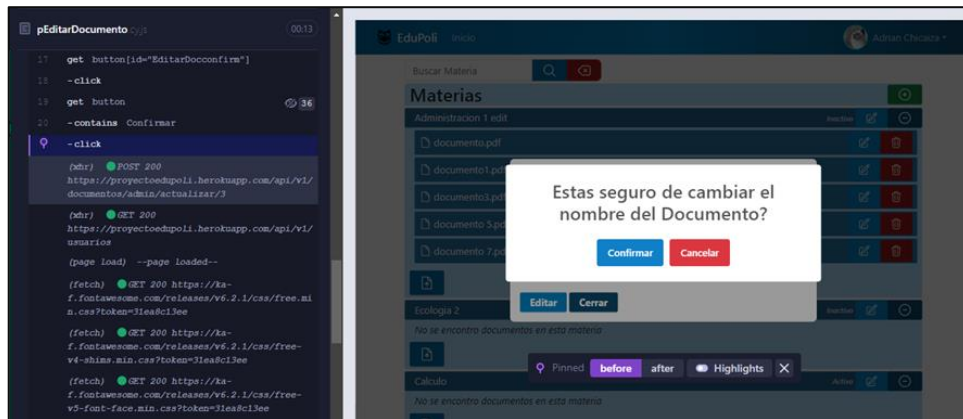


Fig. 87 Resultado de peticiones al editar un documento

Prueba unitaria para la eliminación de un documento mediante acciones como se muestra en la Fig. 88 y los resultados de las peticiones en la Fig.89.

```

cypress > e2e > pEliminarDocumento.cy.js > ...
1 describe("Eliminar Documento", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // Eliminar documento
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="BorrarDocSelec"]').first().click()
11    cy.get("button").contains("Confirmar").click()
12
13  });
14 });

```

Fig. 88 Prueba unitaria para eliminar un documento

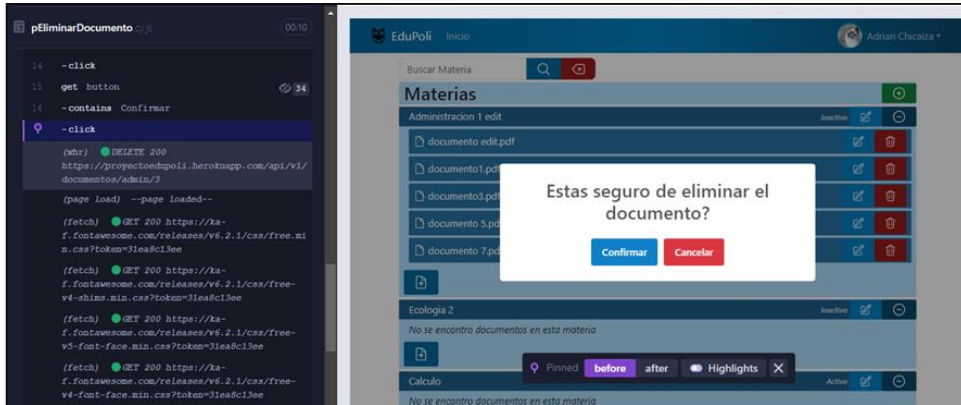


Fig. 89 Resultado de peticiones al eliminar un documento

Prueba unitaria para crear un comentario de ejemplo con la ayuda del código de la **Fig. 90** y los resultados de las peticiones en la **Fig. 91**.

```

cypress > e2e > pComentar.cy.js > ...
1 describe("Crear Comentario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     //Crear Semestre
9     cy.get("button[id="VerMaterias"]').first().click()
10    cy.get('input[id="comentario"]').type("Un nuevo comentario :D")
11    cy.get("button[id="ComentarSend"]').first().click()
12  });
13 });

```

Fig. 90 Prueba unitaria crear un comentario

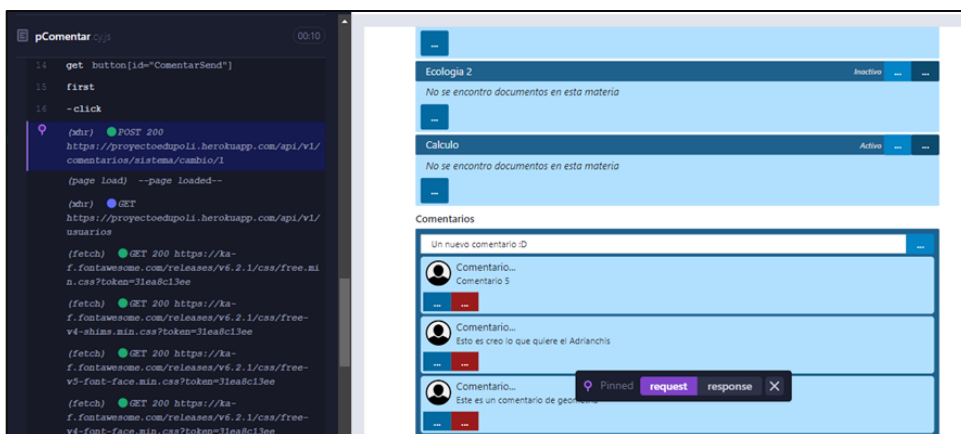


Fig. 91 Resultado de peticiones al crear un comentario

Prueba unitaria para editar el comentario de un usuario, con la ayuda del código de la **Fig. 92** y los resultados de peticiones en la **Fig. 93**.

```

cypress > e2e > pEditarComentario.cy.js > ...
1 describe("Editar Comentario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // Editar Comentario
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="EditComentarioSelect"]').first().click()
11    cy.get('input[id="comentarioEdit"]').type(" edit")
12    cy.get('button[id="EditComent"]').click()
13    cy.get("button").contains("Confirmar").click()
14
15  });
16 });

```

Fig. 92 Prueba unitaria para editar un comentario

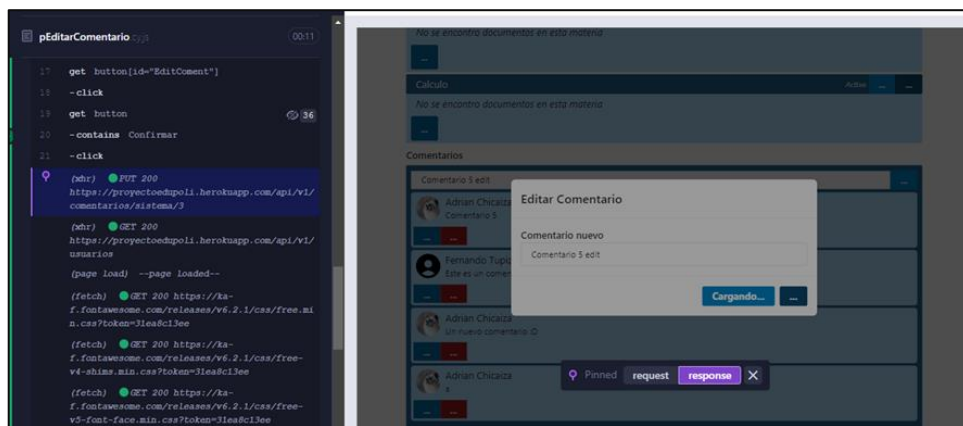


Fig. 93 Resultado de peticiones al editar comentario

Prueba unitaria para la eliminación de un comentario como se muestra en la Fig. 94 al realizar acciones button como dar clic en los botones de eliminar y confirmar, los resultados del tanteo de peticiones lo indica la Fig. 95.

```

cypress > e2e > pEliminarComentario.cy.js > ...
1 describe("Eliminar Comentario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // Eliminar Comentario
9     cy.get('button[id="VerMaterias"]').first().click()
10    cy.get('button[id="EliminarDocSelec"]').first().click()
11    cy.get("button").contains("Confirmar").click()
12
13  });
14 });

```

Fig. 94 Prueba unitaria eliminar comentario

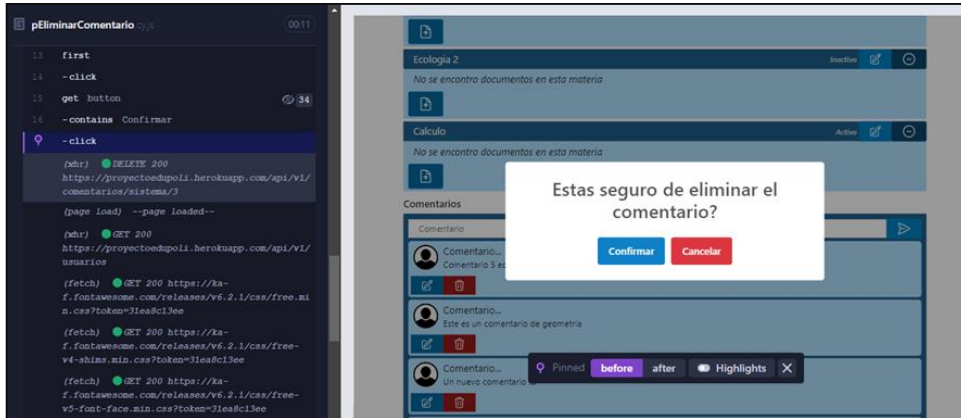


Fig. 95 Resultado de peticiones al eliminar un comentario

Prueba unitaria para ver el perfil de usuario mediante datos de prueba como se indica en la Fig. 96 y los resultados de las peticiones en la Fig. 97.

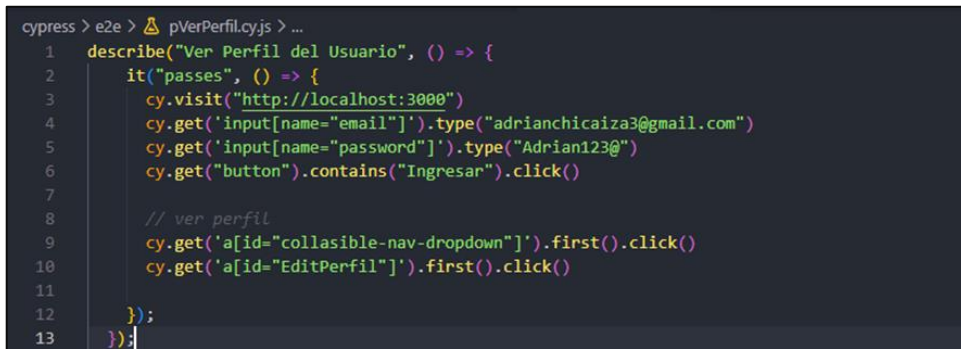


Fig. 96 Prueba unitaria visualizar perfil de usuario

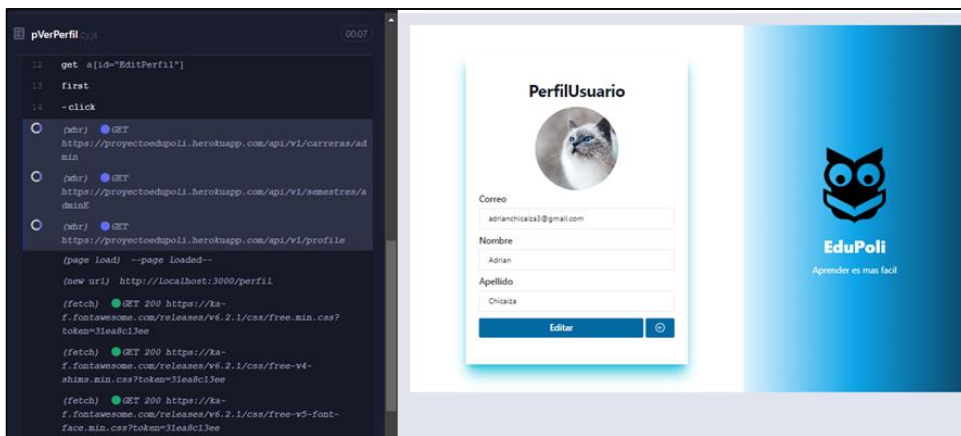


Fig. 97 Tanto de peticiones en perfil

Prueba unitaria para la edición de datos del usuario mediante el cambio de valores en los inputs como se indica en el código de la Fig. 98 y sus resultados de las peticiones en la Fig. 99.

```

cypress > e2e > pEditarPerfil.cy.js > ...
1 describe("Editar Perfil del Usuario", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('input[name="email"]').type("adrianchicaiza3@gmail.com")
5     cy.get('input[name="password"]').type("Adrian123@")
6     cy.get("button").contains("Ingresar").click()
7
8     // Editar perfil
9     cy.get('a[id="collapsible-nav-dropdown"]').first().click()
10    cy.get('a[id="EditPerfil"]').first().click()
11    cy.get('button[id="EditarSelect"]').click()
12    cy.get('input[type=file]').selectFile('cypress/e2e/gato2.jpg')
13    cy.get('input[id="email"]').clear().type("stevenchicaiza@gmail.com")
14    cy.get('input[id="first_name"]').clear().type("Steven")
15    cy.get('input[id="last_name"]').clear().type("Ugsha")
16    cy.get('button[id="ActualizarDatos"]').click()
17    cy.get("button").contains("Guardar").click()
18  });
19 });

```

Fig. 98 Prueba unitaria para editar perfil de usuario

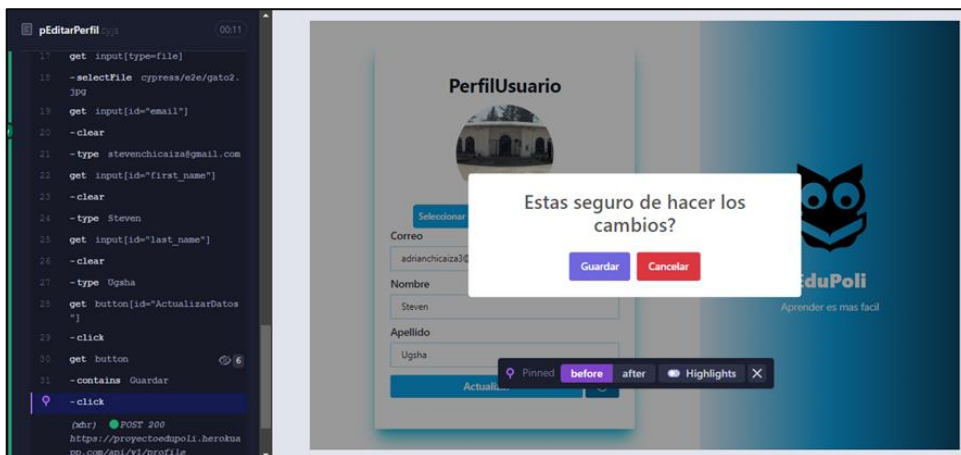


Fig. 99 Resultado de peticiones al editar usuario

Prueba unitaria para el registro de un usuario en la vista de registro, con la ayuda de valores establecidos de la Fig. 100 y los resultados del mismo en la Fig. 101.

```

cypress > e2e > pRegistrar.cy.js > ...
1 describe("Registrar", () => {
2   it("passes", () => {
3     cy.visit("http://localhost:3000")
4     cy.get('a[id="Regist"]').click()
5     cy.get('input[id="first_name"]').type("Juan")
6     cy.get('input[id="last_name"]').type("Salazar")
7     cy.get('input[id="email"]').type("juansalaz45@gmail.com")
8     cy.get('input[id="password"]').type("JuanSalaz123@")
9     cy.get('input[id="password_confirmation"]').type("JuanSalaz123@")
10    cy.get('button[id="RegitrarNuevo"]').click()
11  });
12 });

```

Fig. 100 Prueba unitaria para Registro

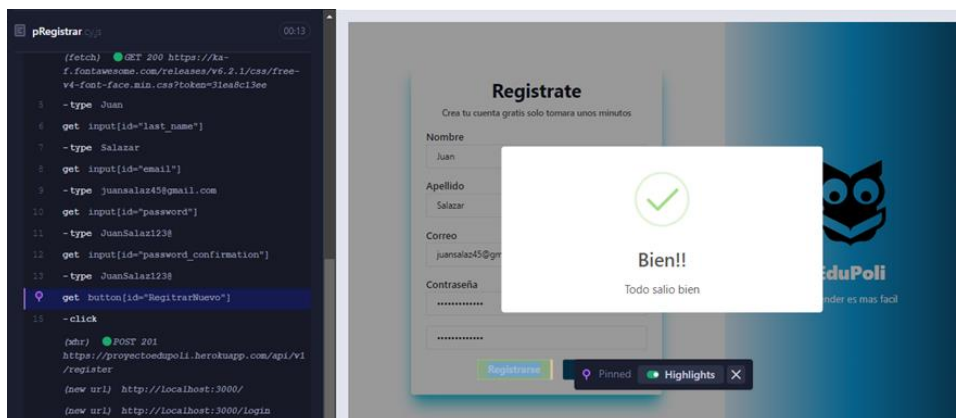


Fig. 101 Resultado de peticiones al registrarse

Pruebas de carga

Este tipo de pruebas consisten en la simulación de casos reales de uso, en donde se toman en cuenta el rendimiento, la accesibilidad, prácticas recomendadas y el SEO de nuestro sitio web, garantizando así el desempeño del presente proyecto. Seguidamente, se indican las pruebas realizadas del proyecto EDUPOLI de la **Fig. 102** a la **Fig. 106**.

Se realiza la prueba de carga general del componente *frontend* en el cual muestra que se obtiene un 98% de Rendimiento, y el 100% en las demás categorías de carga como son la accesibilidad, practicas recomendadas y el SEO, en la **Fig. 102**.

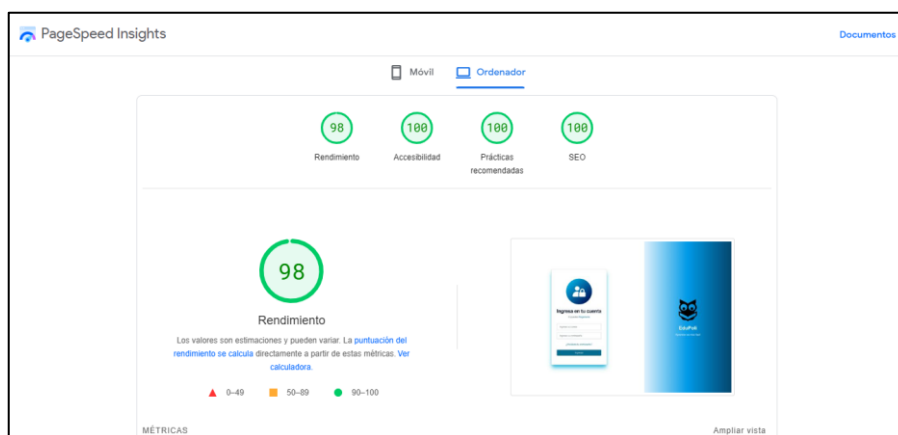


Fig. 102 Prueba de carga general de EDUPOLI

En la **Fig. 103** se muestra la prueba de accesibilidad el cual asegura que todos los usuarios puedan utilizar el sistema web, principalmente enfocado para las personas que tienen discapacidades.

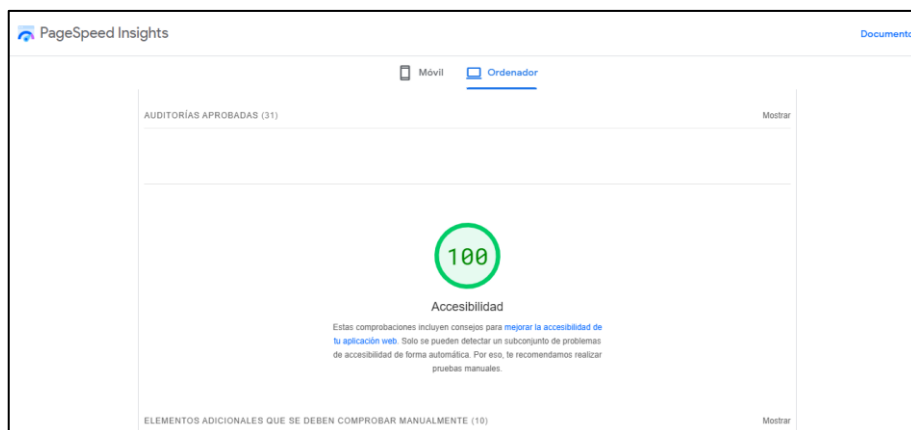


Fig. 103 Prueba de accesibilidad

La prueba de practicas recomendadas se basa en el uso correcto de métodos, código o estructuras para la creación del componente frontend, como resultado obtenemos el 100% en la **Fig. 104** el cual confirma que se aplican practicas recomendadas.

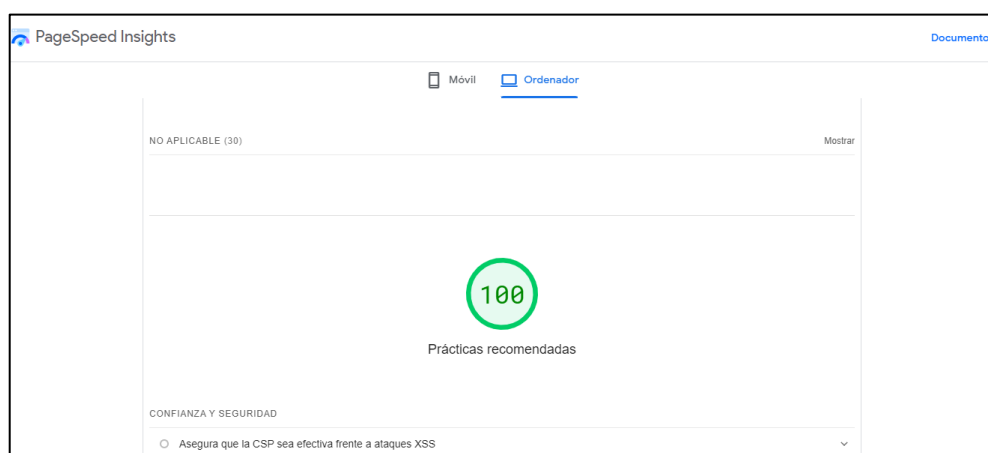


Fig. 104 Prueba de prácticas recomendadas

En la **Fig. 105** se obtuvo el 100% en SEO el cual asegura que el sistema web toma en cuenta las mejores prácticas para la optimización de buscadores, el cual facilitaría el posicionamiento del sistema web en los buscadores cuando los usuarios quieran encontrar EDUPOLI.

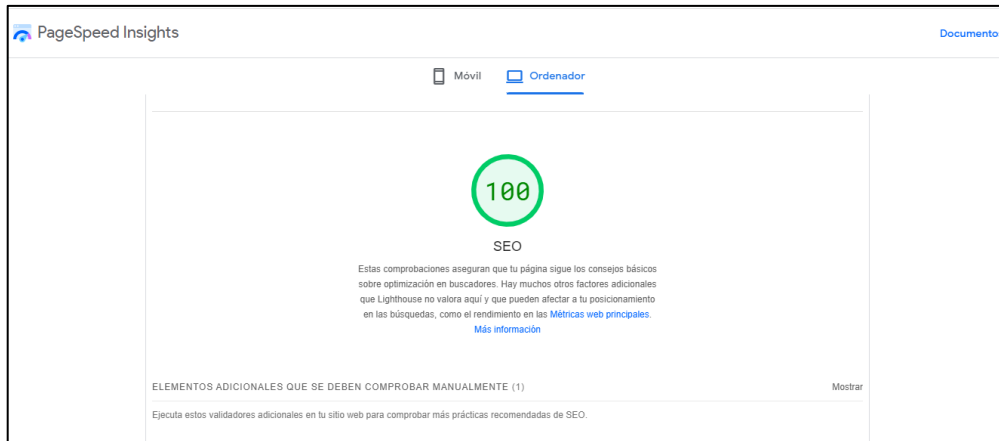


Fig. 105 Prueba de SEO

La **Fig. 106** indica las pruebas de carga generales en dispositivos móviles, el cual presenta un rendimiento del 82%, mientras que en las demás categorías presentan un 100%.

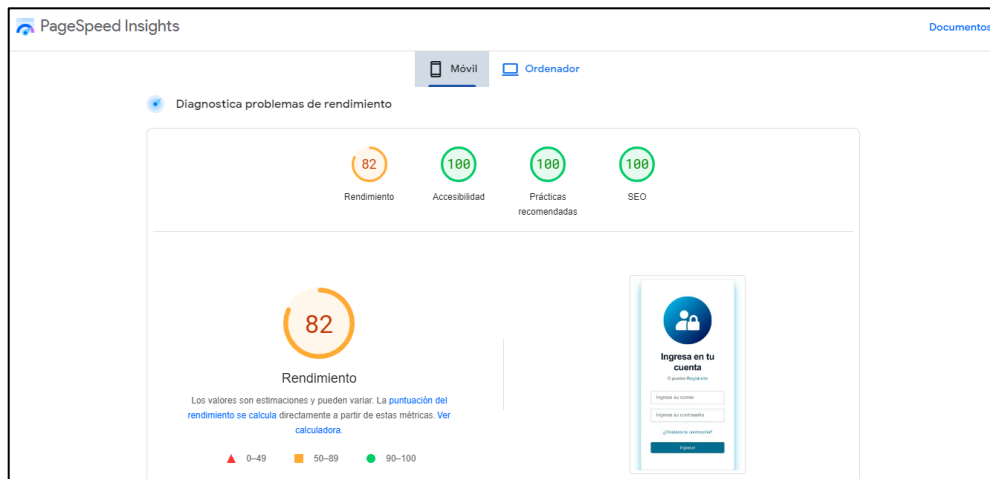


Fig. 106 Prueba de carga general en móvil EDUPOLI

Pruebas de compatibilidad

Estas pruebas consisten en la verificación de nuestro sistema web en distintos navegadores o dispositivos de manera que se presenten los componentes de manera correcta. A continuación, se indica lo obtenido de la **Fig. 107** a la **Fig. 115**.

En la **Fig. 107** se indica la pantalla del *login* el cual carga correctamente en el navegador de Google Chrome.

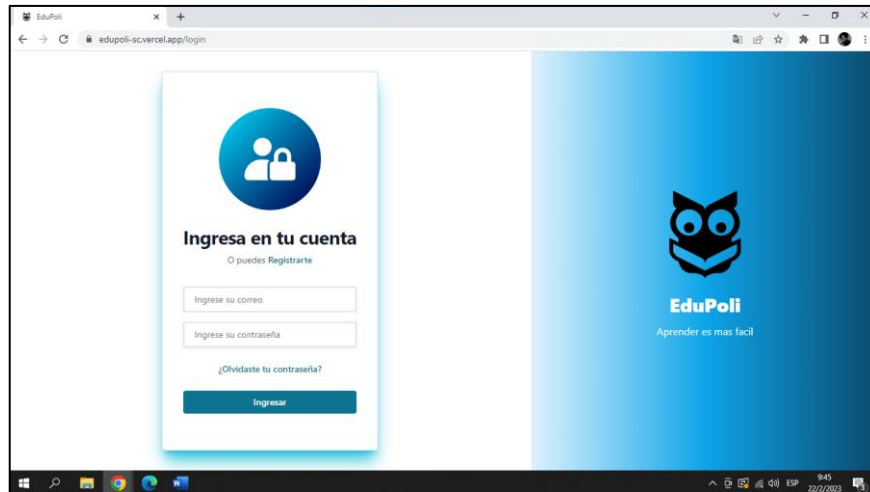


Fig. 107 Compatibilidad con Google Chrome

La **Fig. 108** indica la vista del Dashboard en el navegador de Google Chrome el cual se presenta sin inconvenientes y cargan todos los componentes.

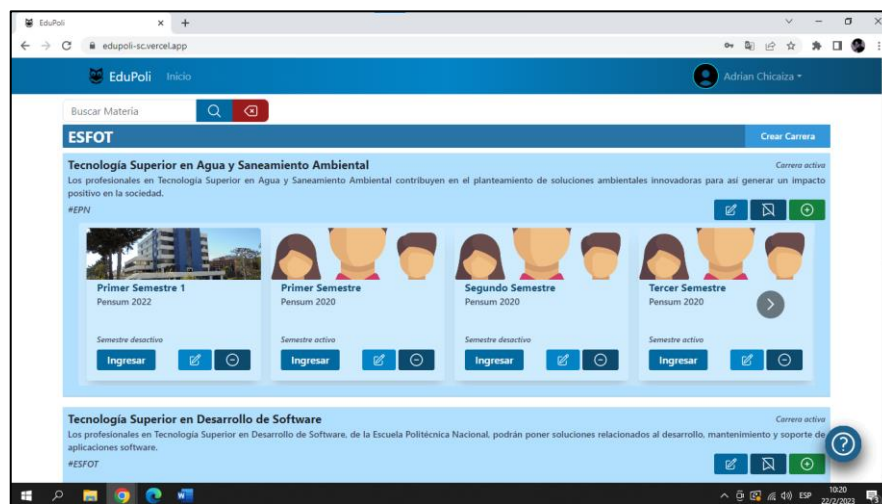


Fig. 108 Compatibilidad con Google Chrome *Dashboard*

En la **Fig. 109** se presenta de manera correcta la pantalla de semestre con sus componentes y materias.

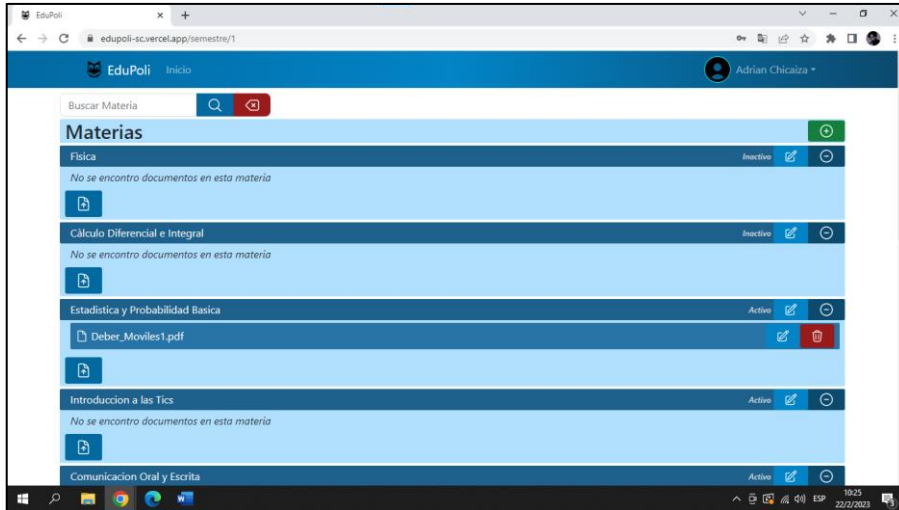


Fig. 109 Compatibilidad con Google Chrome Vista Semestres

La **Fig. 110** indica la pantalla del *login* de manera correcta en el navegador de Microsoft Edge, sin inconvenientes.

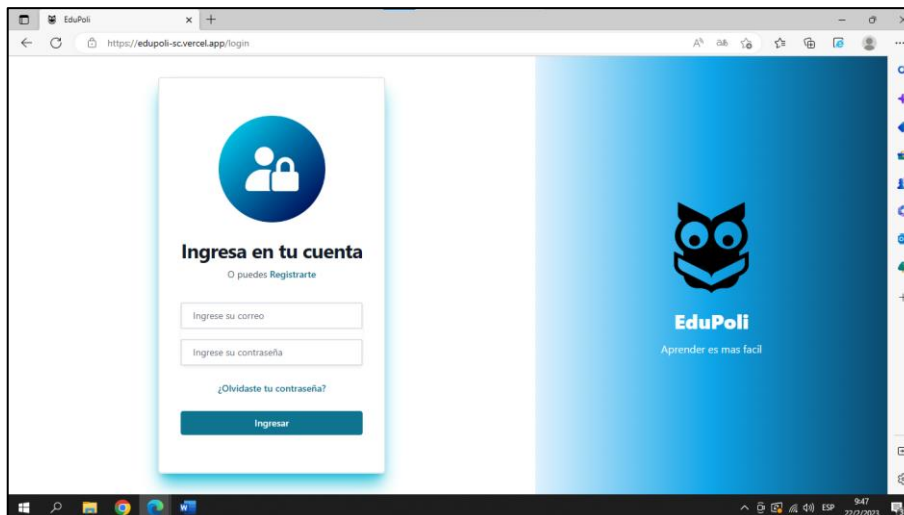


Fig. 110 Compatibilidad con Microsoft Edge

La **Fig. 111** indica la vista del *Dashboard* en el navegador de Microsoft Edge el cual se presenta sin inconvenientes y cargan todos los componentes.

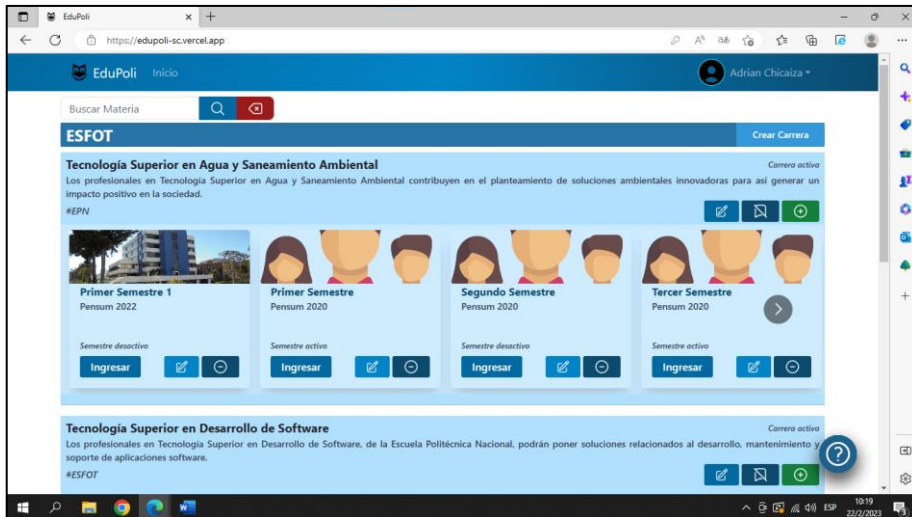


Fig. 111 Compatibilidad con Microsoft Edge Dashboard

En la **Fig. 112** se presenta de manera correcta la pantalla de semestre con sus componentes y materias en el navegador de Microsoft Edge.

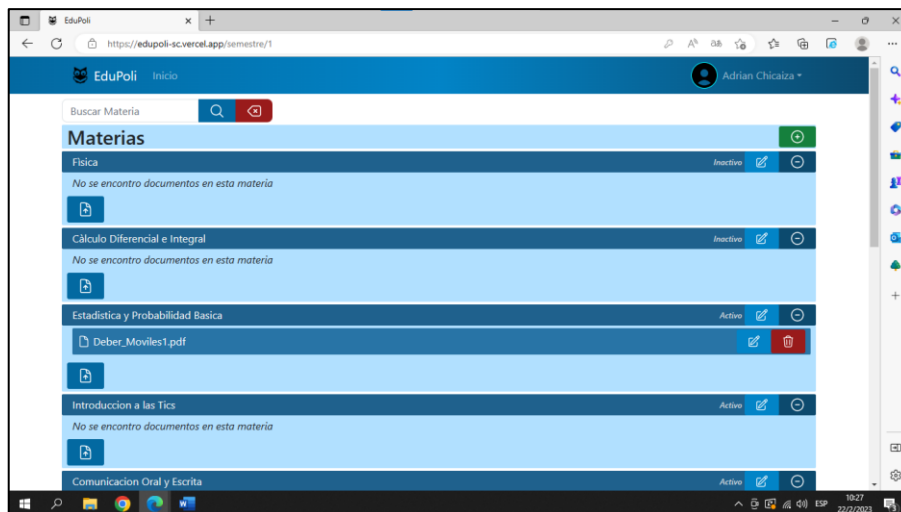


Fig. 112 Compatibilidad con Microsoft Edge Vista Semestres

La **Fig. 113** indica la pantalla del *login* el cual carga correctamente en el navegador de Opera.

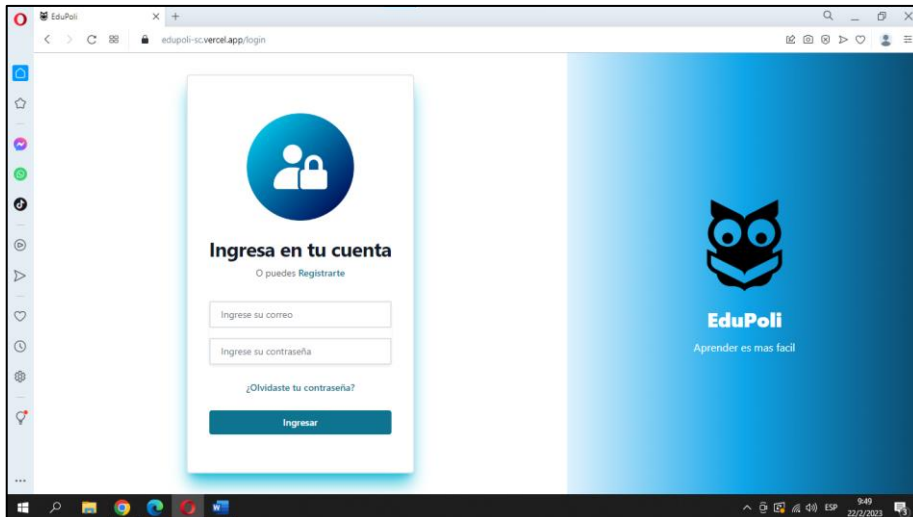


Fig. 113 Compatibilidad con Opera

La **Fig. 114** indica la vista del *Dashboard* en el navegador de Opera el cual se presenta sin inconvenientes y cargan todos los componentes.

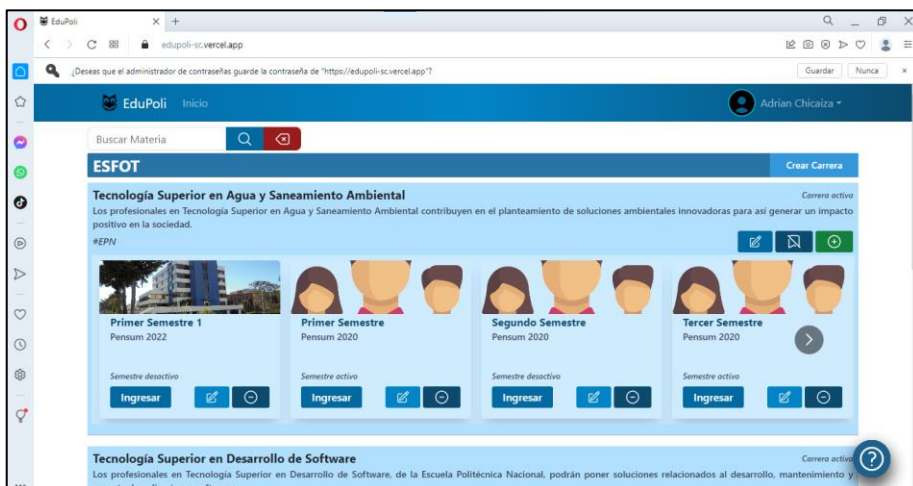


Fig. 114 Compatibilidad con Opera *Dashboard*

En la **Fig. 115** se presenta de manera correcta la pantalla de semestre con sus componentes y materias en el navegador de Opera.

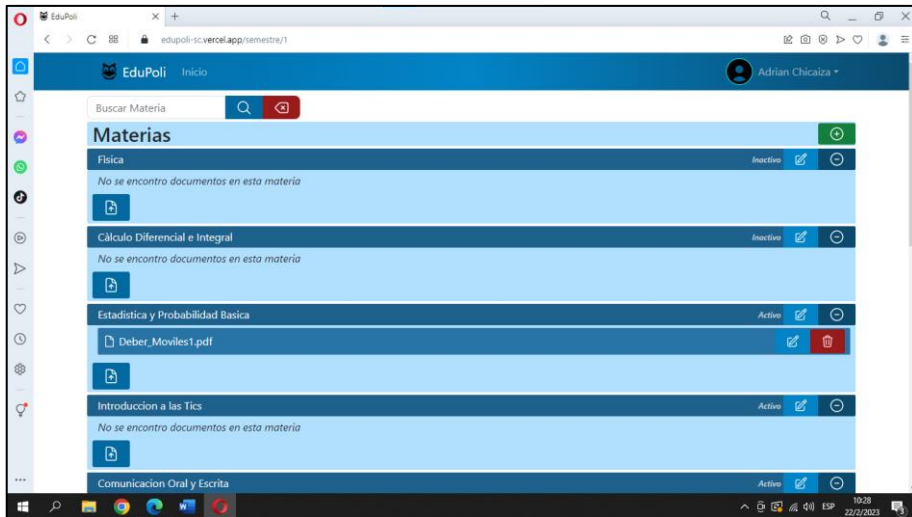


Fig. 115 Compatibilidad con Opera Vista Semestres

Despliegue del *frontend*

El despliegue de nuestro sistema es de suma importancia para compartirlo a través de internet, de manera que se pueda utilizar en cualquier dispositivo con conexión a Internet. Seguidamente, se presenta la manera en cómo se despliega el componente *frontend*, que se indican de la Fig. 116 a la Fig. 120.

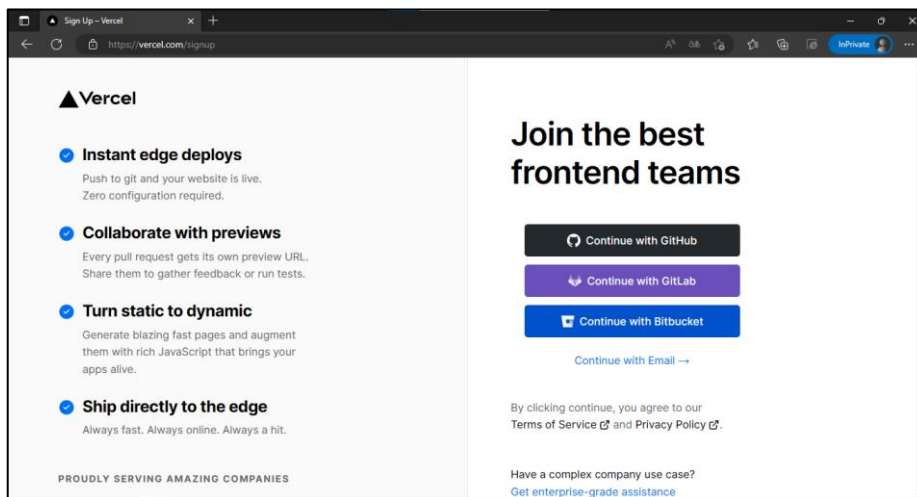


Fig. 116 Creación de cuenta

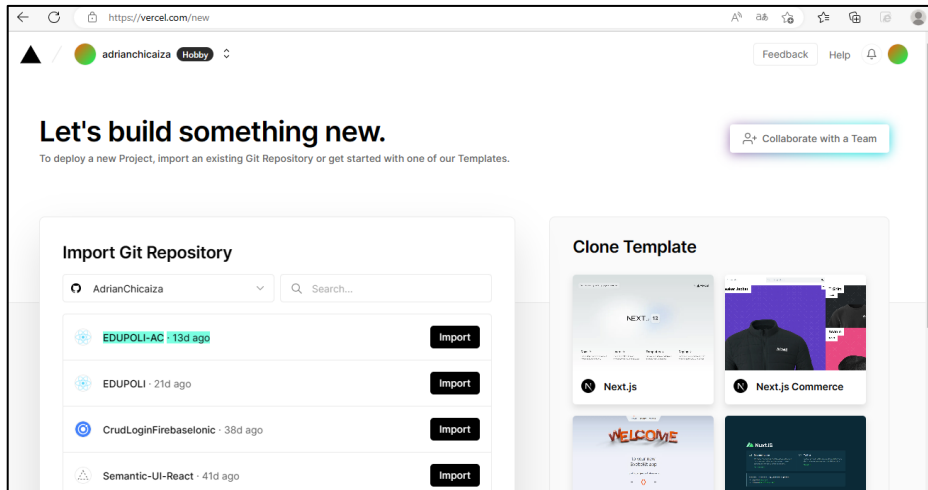


Fig. 117 Seleccionar el repositorio del proyecto

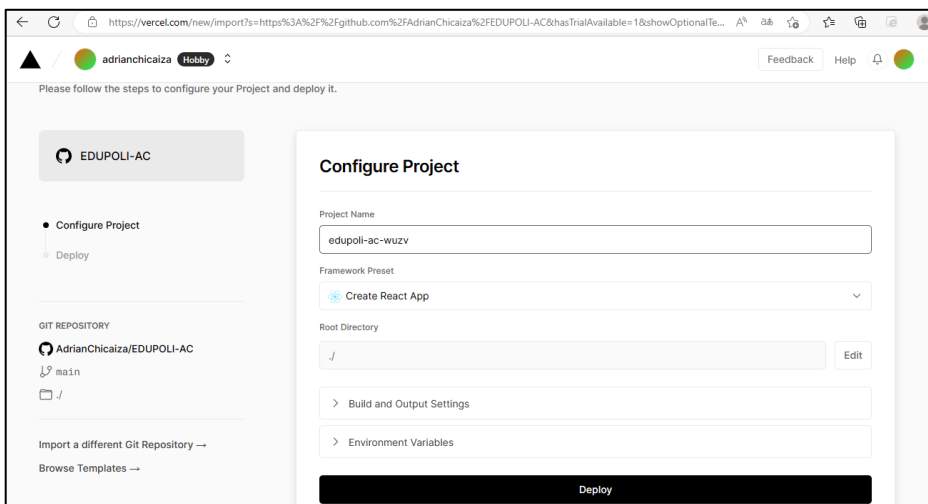


Fig. 118 Escoger el nombre del sitio web

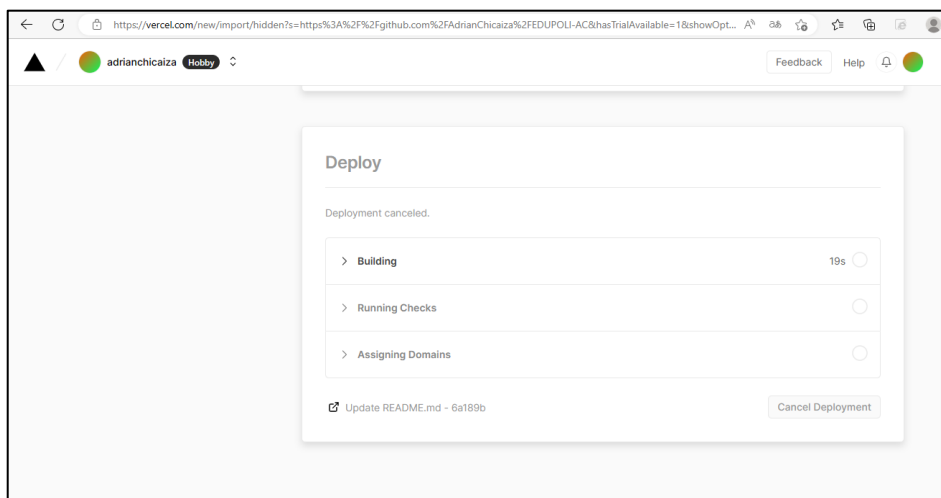


Fig. 119 Desplegar el sistema

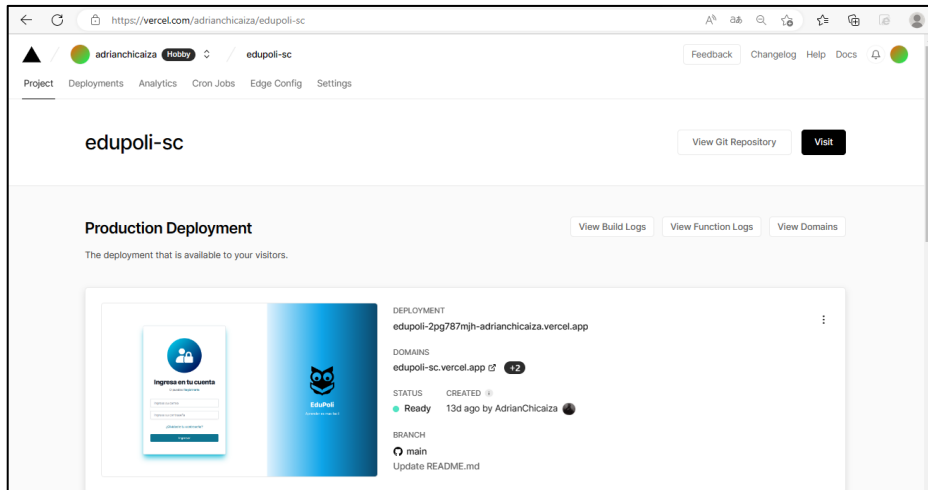


Fig. 120 Resultado desplegado

ANEXO III

Manual de usuario

En este apartado se presenta un video instructivo para utilizar de mejor manera el sistema de EDUPOLI:

https://www.youtube.com/watch?v=tD4j_m7bGb8

ANEXO IV

Manual de instalación

A continuación, se presenta el usuario y contraseña de acceso a EDUPOLI, tanto para el usuario administrador como estudiante, seguidamente del *GitHub* donde consta el código de EDUPOLI.

Credenciales de usuarios.

El ingreso a EDUPOLI se lo hace a través de la herramienta de Vercel la cual proporciono un dominio para el sistema, puedes visitar el siguiente link:

<https://edupoli-sc.vercel.app/login>

Credenciales del perfil administrador:

- Correo: adrianchicaiza3@gmail.com
- Contraseña: Secret123*

Credenciales del perfil estudiante:

- Correo: estudiante@gmail.com
- Contraseña: Estudiante123@

GitHub de EDUPOLI

El código del proyecto se encuentra en el siguiente link de *GitHub*:

<https://github.com/AdrianChicaiza/EDUPOLI-AC>