

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIÓN WEB DE PROMOCIÓN DE EMPREDIMIENTOS DE LA COMUNIDAD DE LA ESFOT.

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

EDUARDO PAÚL FARINANGO MALES

DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO

DMQ, MARZO 2023

CERTIFICACIONES

Yo, EDUARDO PAÚL FARINANGO MALES 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.



EDUARDO PAÚL FARINANGO MALES

eduardo.farinango@epn.edu.ec

epfarinango@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por EDUARDO PAÚL FARINANGO MALES, bajo mi supervisión.



ING. JUAN PABLO ZALDUMBIDE PROAÑO
DIRECTOR

Juan.zaldumbide@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante 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.


EDUARDO PAUL FARINANGO MALES

DEDICATORIA

Dedico este proyecto en primer lugar a Dios, quien me ha permitido continuar mis estudios y alcanzar este momento significativo en mi desarrollo profesional. A mi esposa y mis hijos, que son que son el pilar más importante, por demostrarme su amor incondicional y apoyo, sin importar las diferencias y opiniones. A mis padres, hermanas, hermano, y cuñados que a pesar de las adversidades siempre han estado dispuestos a escucharme y han puesto su hombro en momentos de angustia. A mis suegros, tíos políticos, que me brindaron su ayuda para superar los obstáculos en este tiempo. Sin cada uno de ellos, no habría logrado alcanzar mi objetivo.

AGRADECIMIENTO

Ante todo, quiero expresar mi profundo agradecimiento a Dios por otorgarme la valentía y sabiduría necesarias para superar los obstáculos y lograr mi objetivo en mi formación profesional. También quiero reconocer el papel fundamental que ha tenido mi familia en este logro, especialmente su apoyo incondicional, tanto moral como económico que me brindaron desde el inicio de mis estudios. Sin ellos, no habría sido posible llegar hasta aquí y finalizar con éxito este camino. Les estoy eternamente agradecido por su amor y dedicación.

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
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
2 METODOLOGÍA.....	5
2.1 Metodología de Desarrollo	5
Roles.....	6
Artefactos.....	7
2.2 Diseño de interfaces	9
Herramienta utilizada para el diseño	9
Sistema Web	9
2.3 Diseño de la arquitectura	10
Arquitectura de Datos	10
2.4 Herramientas de desarrollo.....	11
3 RESULTADOS	12
3.1 Iteración 0.....	12

3.2 Iteración 1.....	14
3.3 Iteración 2.....	27
3.4 Iteración 3.....	32
3.5 Iteración 4.....	36
3.6 Iteración 5.....	38
3.7 Despliegue.....	40
4 Conclusiones	42
5 Recomendaciones	43
6 REFERENCIAS BIBLIOGRÁFICAS.....	45
7 ANEXOS.....	1
ANEXO II	3
ANEXO III	33
ANEXO IV.....	34

RESUMEN

Este trabajo tiene como objetivo principal la implementación del frontend de una herramienta diseñada para promover y compartir el espíritu emprendedor dentro de la comunidad de la ESFOT y la Comisión Emprende. Para el cumplimiento de este objetivo se aplicarán diferentes técnicas y estrategias que permitan generar interés y motivación en los visitantes para que desarrollen nuevas ideas.

Para asegurar el éxito del proyecto, se definirá un alcance adecuado siguiendo la metodología de desarrollo de software XP, la cual incluirá una revisión bibliográfica, entrevistas y análisis de datos recabados de la comisión con el fin de crear un diseño visual atractivo y funcional de la herramienta en cada iteración. Así también, se considerará la identidad visual de la Comisión Emprende para crear una experiencia de usuario coherente y familiar, utilizando la herramienta de diseño Figma para prototipar y validar la interfaz de su implementación.

Dentro de este trabajo se optó por utilizar una arquitectura basada en MVC y React, lo que proporcionará una estructura ordenada y mantenible en el tiempo, esta combinación permitirá además una mayor flexibilidad en la implementación de funcionalidad y una mejor interacción con el usuario, todo este proceso se encuentra documentado en las secciones I y II.

Es importante destacar que el enfoque de XP son iteraciones cortas y pruebas en cada implementación que aumentarán la satisfacción del usuario y se lograrán así resultados óptimos en la entrega. Esto se puede observar en la sección III del documento, donde se presentarán los resultados obtenidos y las conclusiones basadas en el proceso descrito. Además, se ofrecen recomendaciones para ayudar al lector.

PALABRAS CLAVE: comunidad ESFOT, técnicas y estrategias, identidad visual, React.

ABSTRACT

The main objective of this work is the implementation of the frontend of a tool designed to promote and share the entrepreneurial spirit within the ESFOT community and the Emprende Commission. In order to achieve this objective, different techniques and strategies will be applied to generate interest and motivation in visitors to develop new ideas.

To ensure the success of the project, an adequate scope will be defined following the XP software development methodology, which will include a literature review, interviews and analysis of data collected from the commission in order to create an attractive and functional visual design of the tool in each iteration. Likewise, the visual identity of the Emprende Commission will be considered to create a coherent and familiar user experience, using the Figma design tool to prototype and validate the interface of its implementation.

Within this work it was decided to use an architecture based on MVC and React, which will provide an orderly and maintainable structure over time, this combination will also allow greater flexibility in the implementation of functionality and better interaction with the user, this whole process is documented in sections I and II.

It is important to highlight that the XP approach consists of short iterations and tests in each implementation that will increase user satisfaction, and thus achieve optimal results in the delivery. This can be seen in section III of the paper, where the results obtained, and conclusions based on the process described will be presented. In addition, recommendations are provided to assist the reader.

KEYWORDS: Community, ESFOT, techniques and strategies, visual identity, React

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La interacción de la humanidad con la *web* se ha vuelto imprescindible puesto que esta abarca todos y cada uno de los campos en el que el hombre se puede desarrollar profesional y económicamente de manera autónoma. La llegada del COVID-19 ha generado un fuerte impacto económico en muchos emprendedores y empresas de todos los tamaños debido a la disminución de la demanda de sus productos y la interrupción de la cadena de suministro dejando como única opción el cierre temporal de sus puertas. La pandemia ha acelerado la demanda de productos de primera necesidad y de servicios digitales, haciendo que los emprendedores busquen una manera de pivotar sus modelos de negocios para poder generar presencia digital demostrando así su capacidad de adaptación en tiempos de crisis.

El emprendimiento promocionado de forma digital permite una mayor flexibilidad y libertad a comparación de uno tradicional generando así un ahorro en costos de arriendos de un local comercial y en consecuencia pueden llegar a un público más amplio y global, en otras palabras, un emprendimiento digital permite a las personas vender y consumir productos de un negocio desde cualquier lugar con una conexión a Internet.

En resumen, la pandemia ha creado nuevos desafíos para las personas, empresas y entidades educativas, así mismo se han generado nuevas oportunidades para quienes estén dispuestos a adaptarse y adoptar el espíritu digital. Con el ánimo de aportar a este crecimiento docentes altamente calificados de la ESFOT ha contactado varios emprendimientos de profesionales y emprendedores en todas las ramas, centralizando su información para poder promocionar sus productos como parte de sus objetivos institucionales a través de una un radar comercial.

Este proceso es llevado de forma no convencional tomando en cuenta el avance de la tecnología de esta década, mediante publicaciones generales en redes sociales o distribuyendo un catálogo en formato pdf dan a conocer varios emprendimientos con información básica del emprendimiento.

Es por lo que este componente es creado para que no importe el lugar ni el momento, el usuario podrá ver los distintos emprendimientos que estén avalados por la comunidad de una forma gráfica e interactiva en la *web*. Utilizando tecnologías que permitan la interacción de los administradores con los emprendedores creando una conexión directa con todo tipo de clientes potenciales.

1.1 Objetivo general

Desarrollar el frontend de una aplicación *web* que permita promocionar los emprendimientos propuestos por los miembros de la comunidad de la ESFOT.

1.2 Objetivos específicos

Levantar los requerimientos del frontend de la aplicación *web*.

Diseñar los prototipos y la arquitectura de la solución.

Codificar los diseños implementando las funcionalidades.

Comprobar el funcionamiento del frontend de la aplicación.

Desplegar el componente en una plataforma especializada para alojar el frontend de la aplicación.

1.3 Alcance

El presente componente es creado con el fin de dar funcionalidad visual y permitir que la comunidad de emprendedores de la ESFOT pueda interactuar con los diferentes emprendimientos publicados en la aplicación. Dentro de las principales actividades en este componente se tiene programado realizar entrevistas con los encargados de la comisión Emprende, esto nos permitirá conocer el trabajo que realizan y así levantar requerimientos necesarios. Se realizará prototipos que permitan crear una idea de cómo se verá el frontend llevando un contexto visual acorde a la marca Emprende.

Uno de los módulos que forman parte del frontend es la página de inicio en la que se realizará la presentación de la comisión Emprende junto a la sección de módulos en los que se podrá visualizar la información de los emprendimientos que incluirá datos de los fundadores, productos, dirección y contactos junto a la dirección exacta en el caso de que tenga local físico.

Se creará el módulo de administración que permitirá la creación de usuarios en este caso serán: super administrador y administrador, esto permitirá llevar un control activo de los procesos, una de las principales actividades que podrán realizar los administradores será aprobar y validar la información de los usuarios mediante el uso de formularios que servirán para la recepción de solicitudes de aprobación de sus emprendimientos y la publicación de videoconferencias organizadas por la comisión Emprende.

1.4 Marco Teórico

Una de las maneras que el humano ha logrado generar conocimiento científico en cualquier campo, es la aplicación de metodologías permitiendo un enfoque sistemático a un conjunto de principios para resolver un problema o lograr una meta y así poder encontrar las ventajas y limitaciones de un trabajo de investigación con resultados óptimos [1]. Una buena metodología está bien definida, cuando es sistemática y se basa en principios sólidos y mejores prácticas en el campo correspondiente [2]. Dentro del campo de desarrollo de software existe también la necesidad de aplicar metodologías de desarrollo de software enfocadas a la creación de sistemas eficientes y de calidad que cumplan con el ciclo de vida de software establecido.

Metodologías de desarrollo de software

Enfocadas en la colaboración en equipo cuyo objetivo es construir sistemas eficaces, robustos y escalables de una manera relativamente sencilla mediante la organización y distribución de tareas, con niveles y jerarquías establecidas para lograr un buen desempeño [3].

Dentro de la clasificación de metodologías para el desarrollo de software están las metodologías tradicionales y las metodologías ágiles siendo esta última la más utilizada debido a sus buenos resultados y alto rendimiento en los sistemas de información.

Metodologías Ágiles

Las metodologías ágiles son un conjunto de principios y prácticas para el desarrollo de software que enfatizan la colaboración, la flexibilidad y la iteración. El objetivo del desarrollo ágil de software es entregar software de alta calidad de manera rápida y eficiente, dividiendo los proyectos en partes más pequeñas, manejables y reevaluando regularmente el plan de desarrollo en función de los comentarios de las partes interesadas con un objetivo común de permitir que los equipos entreguen software de alta calidad de manera rápida y eficiente [4].

Aplicación Web

Dentro de la creación de aplicaciones web podemos encontrar muchos servicios relacionados al entorno laboral o educativo, estas aplicaciones pueden ser utilizadas directamente en un navegador web sin necesidad de instalar ningún programa en nuestro equipo [5] siendo lo primero que vea el usuario el frontend.

FrontEnd

El desarrollo frontend, también conocido como desarrollo del lado del cliente, se refiere a la práctica de crear la interfaz junto a la experiencia del usuario, se complementa con el proceso del diseño y la implementación de los elementos visuales e interactivos de un sitio *web*, como el diseño, los colores y las fuentes [6], así como el comportamiento de esos elementos cuando los usuarios interactúan con ellos, formando así la base del frontend que no es más que el uso de herramientas y técnicas enfocadas al usuario que luego de seguir una metodología específica y la codificación necesaria se podrán ver y estar disponibles para interactuar directamente con el usuario [6].

Basados en este concepto este componente que forma parte de una aplicación que permitirá la promoción de emprendimientos de la comunidad de la ESFOT, se fundamentará en una metodología ágil que, básicamente se compone de métodos y técnicas específicas que permiten enfocar el esfuerzo en la creación del proyecto de software modelándolo de manera que brinde flexibilidad, eficacia y autonomía generando costos bajos en comparación al producto generado.

2 METODOLOGÍA

Una de las principales herramientas que tiene el humano es su inteligencia, esta le permite observar a su alrededor y preguntarse el porqué de las cosas, planteándose así teorías que servirán para iniciar una investigación en su propio entorno tomando en cuenta que es un actor social quien puede o no mediante su interpretación encontrar una teoría que le permita encontrar su respuesta a su problema [1].

En este caso, se ha identificado una problemática real y se ha planteado una hipótesis que deberá ser confirmada o rechazada mediante la utilización de la metodología cualitativa, que permitirá llegar a un cierto nivel de conocimiento junto con los participantes que estén involucrados en esta investigación y que actúen de manera interactiva [7].

Para ello, se utilizará el diseño de Investigación – Acción debido a que el resultado nos permitirá reformar de forma estructural ciertos procesos actuales con ayuda de los participantes que estén inmersos en esta problemática, y así recopilaremos datos mediante una serie de reuniones que permitirá cumplir con las fases de este diseño: observar, pensar y actuar [7].

2.1 Metodología de Desarrollo

Dentro del desarrollo de software, existen varias metodologías ágiles que permiten a los desarrolladores contar con pautas para poder diseñar software de calidad. Para este proyecto, se estará aplicando la metodología XP, que básicamente busca simplificar el diseño de manera sutil, llevar el desarrollo del proyecto y, después de realizar las pruebas necesarias, entregar el software de manera incremental. XP se basa en reuniones recurrentes en las cuales se plantearán iteraciones que deberán ser entregadas en el menor tiempo posible [4].

Fases de la metodología XP

Para la correcta implementación de esta metodología ágil en la creación de la aplicación web se deberá seguir un orden establecido como se observa en la **TABLA I**.

TABLA I: Fases XP

Planificación	Determinación y levantamiento de requerimientos.
Diseño	Diseño y codificación de la aplicación.

Prueba	Codificar y ejecutar pruebas que garanticen que el código realice sus funciones.
Implementación	Entrega de una aplicación en forma de producto al cliente.

Roles

Dentro de la metodología XP tenemos distintos roles enfocándonos en el frontend como lo son:

Programador

Es la persona encargada de recopilar datos en forma de historias de usuario, esto permitirá realizar las iteraciones de forma prioritaria de manera coordinada con el cliente y así entregar nuevas funcionalidades en cada interacción.

Cliente

Esta metodología gira alrededor directamente con el cliente quien es el que delimita las características que tiene el software de manera comprensible y responsable. El cliente también tiene la última palabra sobre qué características se implementan y es el único que puede considerar que el proyecto ha terminado, así también se espera que el cliente esté disponible durante todo el proyecto para brindar comentarios y responder preguntas [8].

Consultor

Persona externa al equipo de desarrollo que posee el conocimiento necesario para brindar ayuda en problemas específicos, y brinda capacitación al equipo de programadores. El consultor también puede brindar tutoría a los miembros del equipo para ayudarlos a mejorar sus habilidades y conocimientos, es importante tener en cuenta que el consultor no reemplaza al cliente, y el cliente sigue siendo quien toma las decisiones finales en el proyecto [9], esta distribución se observa resumida en la **TABLA II** .

TABLA II: Asignación de roles

ROLES	NOMBRES
Cliente	Comisión Emprede EPN – ESFOT
Consultor	Ing. Juan Pablo Zaldumbide
Programador	Eduardo Farinango

Artefactos

Recopilación de requerimientos

Mediante entrevistas a los representantes de la comisión Emprende se ha comprobado las necesidades que la comisión presenta al momento de promocionar los emprendimientos, esto permitirá después de un análisis de los alcances, crear un diseño con una estructura adecuada para la creación del frontend.

Historias de Usuario

Una historia de usuario es una descripción breve y simple de una característica o parte de la funcionalidad desde la perspectiva del usuario. Las historias de usuario se utilizan para capturar los requisitos y objetivos de un proyecto y sirven como base para la discusión en el proceso de desarrollo de software ágil, y la planificación entre el equipo de programadores y las partes interesadas como muestra de este proceso se tiene la **TABLA III**, el complemento de las historias de usuarios se las podrá encontrar en el **ANEXO II**. Manual técnico, apartado Historias de Usuarios.

TABLA III: Historias de usuarios 1 – Página de inicio

HISTORIA DE USUARIO	
Identificador (ID): HU000	Usuario: Administradores
Nombre de la Historia: Página de inicio	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Iteración Asignada: 0	
Responsable: Farinango Eduardo	
Descripción: Como usuario desea ver una página de inicio con la presentación de la comisión Emprende, en su contenido debe tener el logo de la marca Emprende, sus colores y un botón para el envío de un formulario.	
Observaciones: El formulario deberá ser enviado con todos los campos llenos añadiéndole de manera obligatoria una foto referencial al emprendimiento.	

Iteraciones

Una iteración es un período de tiempo limitado durante el cual se desarrolla y entrega al cliente un conjunto específico de funciones. Las iteraciones suelen durar entre una y cuatro semanas, según el proyecto. Cada iteración comienza con una reunión de planificación en la que el equipo y el cliente deciden qué funciones se implementarán. Luego, el equipo trabaja en la implementación de estas funciones y realiza pruebas para garantizar que cumplan con los requisitos del cliente. La iteración finaliza con una demostración de las características completadas al cliente [10], la lista de iteraciones para este componente se la puede observar en la **TABLA IV**.

TABLA IV: Iteraciones para la aplicación *web*

ITERACION	DESCRIPCION
0	Configuración del ambiente de desarrollo Diseño y prototipado Codificación de la página de inicio. Página de información de la comisión EMPRENDE
1	Barra de navegación. Formularios de inicio de sesión. Control de inicio de sesión consumiendo datos de una <i>endpoint</i> . Creación del módulo y CRUD usuarios Super-Admin
2	Creación del módulo y CRUD de usuarios administradores generales. Creación del módulo y CRUD de emprendimientos. Validación de campos para formularios.
3	Creación de formularios petición de ingreso Alertas de nuevas peticiones.
4	Filtros de búsqueda de emprendimientos. Creación del módulo de videoconferencias de la comisión.
5	Pruebas. Despliegue.

2.2 Diseño de interfaces

El diseño de la interfaz es un aspecto esencial del desarrollo *web*, ya que juega un papel crucial en la determinación de la experiencia del usuario de una aplicación *web*. Un buen diseño de interfaz se basa en una comprensión profunda de las necesidades, los objetivos de los usuarios, basándose en varios factores como la usabilidad, la accesibilidad y la estética. El diseño de interfaz implica una combinación de investigación, planificación, pruebas y refinamientos iterativos, ya que juega un papel clave para determinar el éxito de un sitio *web* o una aplicación. Una interfaz bien diseñada puede mejorar la satisfacción del usuario y ayudar a garantizar el éxito del proyecto [11].

Herramienta utilizada para el diseño

Una de las herramientas principales que se dedican especialmente en el prototipado y creación de mockups es FIGMA, muy útil al momento de crear prototipos para *web* y editar gráficos vectoriales debido a su usabilidad y de su intuitivo manejo [12].

Sistema Web

Para poder realizar el prototipado de la aplicación *web* de la comisión Emprende de la ESFOT se deben utilizar los colores que lo identifican, Figma nos da la posibilidad de configurar tanto los colores y la tipografía de la marca Emprende así también nos permite maquetar como se verá el módulo de administración de usuarios como se observa en la **Figura 2.1**, los prototipos principales restantes están listados en el **ANEXO II** Manual Técnico, apartado Prototipado.

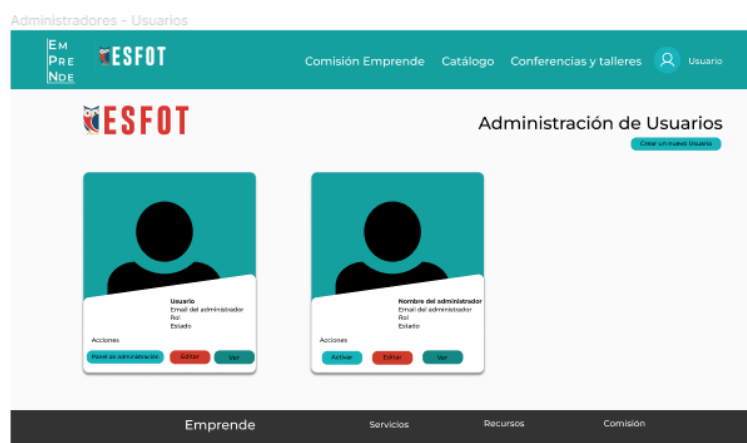


Figura 2.1 Prototipo – Panel administración de usuarios.

2.3 Diseño de la arquitectura

Cuando se trata de la implementación de un nuevo sistema de información, es muy importante considerar ciertos factores esenciales que harán que la estructura del proyecto sea consistente, reduciendo así el tiempo y el esfuerzo dedicado a la búsqueda de la calidad y confiabilidad del software durante las pruebas. Además, se debe tratar de garantizar la escalabilidad y flexibilidad del sistema, para que el sistema pueda evolucionar sin la carga de trabajo que lo haría pesado y difícil de mantener. Por lo que uno de los pilares importantes es el diseño de patrones arquitectónicos.

Arquitectura de Datos

En la implementación del presente proyecto se aplicará el patrón arquitectónico MVC (Modelo Vista Controlador), debido a su flexibilidad al momento de separar el manejo de datos, la presentación de información al usuario y la lógica del negocio en tres componentes diferentes [13]. El modelo especifica los datos que va a tener el sistema tomando en cuenta su estado para poder presentarlos en la vista [14].

La vista es responsable de mostrar los datos al usuario y permitir que el usuario interactúe con el sistema. La vista se implementará utilizando componentes *REACT* y estilos CSS proporcionado por la herramienta Bootstrap que en conjunto con los componentes de orden superior *HOC* se reutilizarán componentes de *React* necesarios [15], por otra parte, dentro de la vista esta Axios que mediante el uso de promesas enviaremos *XMLHttpRequests* desde el navegador para poder consumir datos desde una *API* [16].

El controlador es el intermediario entre el modelo y la vista que actualiza sus estados dependiendo de las peticiones del usuario [14], esto se conceptualiza de mejor manera en la **Figura 2.2**.

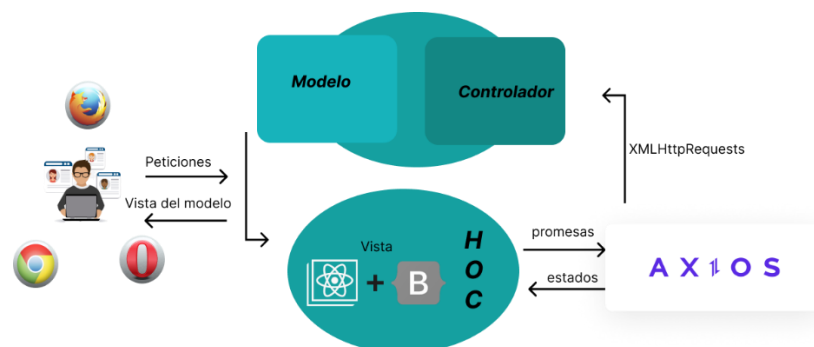


Figura 2.2 Patrón Arquitectónico FrontEnd

2.4 Herramientas de desarrollo

Las herramientas de desarrollo *web* son importantes porque ayudan a simplificar el proceso de creación de sitios *web*, aplicaciones y otros proyectos digitales. Pueden simplificar y agilizar el proceso de desarrollo al automatizar tareas y proporcionar una plataforma para la experimentación y la colaboración. Las herramientas de desarrollo *web* también pueden ahorrarle tiempo y recursos al reducir la cantidad de código que escribe, en la **TABLA V** se puede observar las herramientas que se utilizarán en todo el proceso.

TABLA V: Herramientas para el desarrollo del Sistema *Web*

Herramienta	Justificación
React	Permite crear componentes reutilizables que ayudan a acelerar el proceso de desarrollo facilitando el mantenimiento y la actualización de código.
Node.js	Permite la comunicación y las actualizaciones en tiempo real, lo que puede ser útil para las aplicaciones que requieren datos en tiempo real o funciones interactivas.
HTML	HTML es una herramienta importante para construir y publicar contenido en la <i>web</i> , y es una parte fundamental de la <i>Web</i> .
Bootstrap	Proporciona una forma consistente y fácil de usar para diseñar y crear sitios <i>web</i> receptivos y móviles.
Axios	Proporciona una forma eficiente de realizar solicitudes HTTP desde el navegador.

Adicional a estas herramientas se utilizarán librerías que fueron optimizadas para trabajar con React mismas que nos servirán para crear conponentes y dar estilos a nuestra aplicación estas serán listadas en el **ANEXO II** apartado Librerías.

3 RESULTADOS

3.1 Iteración 0

Esta iteración primera iteración contempla la base del proyecto como se observa en a la **TABLA VI** y se complementa con la historia de usuario 0 -1.

TABLA VI: Iteración 0

Iteración	Historia de usuario #	Descripción
0	HU000 – Página de inicio	Como usuarios se desea visualizar una página de inicio con el logo de la comisión EMPRENDE y los colores establecidos.
0	HU001 - Información Comisión Emprende	Administradores y usuarios general desean visualizar información sobre la comisión Emprende, sus miembros, contactos y redes sociales.

Configuración del ambiente de desarrollo

En este apartado se presentará el proceso de instalación y configuración de las herramientas que nos permitirán la implementación de nuestro componente, en este caso inicialmente para poder codificar necesitaremos un entorno de desarrollo (*IDE*).

Proceso de instalación de herramientas.

Visual Studio Code es un editor de código popular no solo sirve para crear aplicaciones con *React*. Tiene una interfaz limpia e intuitiva, así como una gama de funciones integradas que pueden hacer que la codificación sea más rápida y eficiente, como la finalización del código y el resaltado de sintaxis.

Node.js es un entorno de tiempo de ejecución que se utiliza para crear aplicaciones del lado del servidor con *JavaScript* que permite a los desarrolladores usarlo en el backend, lo que facilita la creación de aplicaciones completas con *React* estas herramientas pueden ser personalizadas para satisfacer las necesidades de este proyecto el proceso será litado en el **ANEXO II** apartado Herramientas.

Una vez instalados nuestras herramientas y nuestros *frameworks* se creará el directorio con las carpetas necesarias para alojar nuestro código esta organización se la puede observar en la **Figura 3.1** Directorio aplicación Emprende

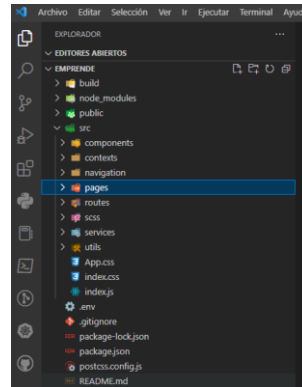


Figura 3.1 Directorio aplicación Emprende

Implementación de la página de inicio.

La página de inicio es un apartado creado específicamente con el propósito de permitir la visualización de toda la información de interés para el público. Con el uso de componentes de *React* sumado al gran desempeño de *Bootstrap* podremos implementar diseños basados en la marca Emprende anteriormente diseñada.

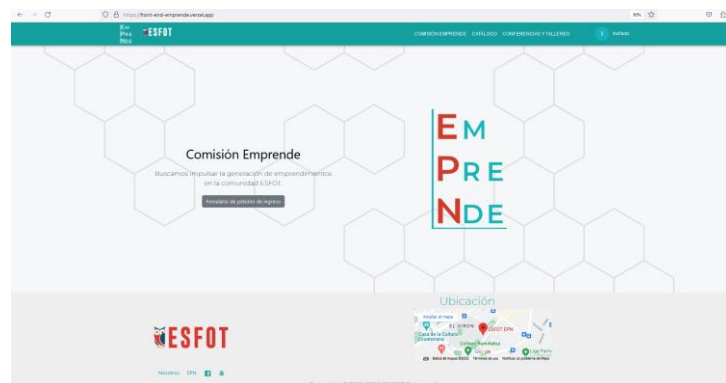


Figura 3.2 Página de inicio aplicación emprende

Para poder visualizar esta pantalla se hizo uso de dos componentes, fueron codificados en base a las necesidades de la estructura del proyecto, este código exporta un componente funcional llamado "Inicio" que contiene los componentes, "*Main*" y "*Footer*", dentro de él. Estos componentes de página se importan desde sus respectivas ubicaciones de archivo tomando siempre en cuenta los prototipos y colores de la marca Emprende configurados en las hojas de estilo o directamente con Bootstrap.

Página Información EmPreNde

En el caso de la información de la Comisión EmPrende tiene una propia sección en la que se desplegará la información como: historia, visión, miembros que la conforman, redes sociales y ubicación.



Figura 3.3 Comisión EmPrende

3.2 Iteración 1

En la **TABLA VII** se observa que esta iteración se complementa con las historias de usuario 2, 3 y 4.

TABLA VII: Iteración 1

Iteración	Historia de usuario #	Descripción
1	HU002 – Menú general (Barra de navegación)	Como administradores desean ver un menú en el que se tenga opciones: Comisión, catálogo de emprendimientos, repositorio de conferencias, panel de administración.
1	HU003 - Inicio de sesión	Como administradores desean consumir un <i>endpoint</i> que le permita iniciar sesión, recuperar y cambiar contraseña.
1	HU004 - CRUD usuarios super-administradores. HU005 - CRUD usuarios administradores.	Como Super-Admin se desea consumir un <i>endpoint</i> que mediante un formulario permita crear, actualizar o eliminar un usuario y super-administrador.

Barra de navegación

Uno de los componentes principales de la aplicación es la barra de navegación puesto que esta nos brinda funcionalidades que nos permiten interactuar con la aplicación, cuenta con accesos a los apartados de la Comisión Emprende, catálogo de emprendimientos, repositorio de videoconferencias y como parte principal de este componente está el botón en forma de avatar que nos permite ingresar al sistema.



Figura 3.4 Barra de navegación – Inicio de sesión

Esta barra de navegación está construida con la ayuda de la librería MUI que contiene componentes precargados con funcionalidades útiles al momento de integrarlas a React [17] que brindara una experiencia agradable a la vista del usuario. Junto a los operadores ternarios podremos identificar si un usuario inicio sesión para poder presentar el menú adecuado de sesión junto al nombre de usuario y la opción de cerrar sesión. En el caso de un visitante siempre se mostrará el nombre “Invitado”. Este componente esta por sobre todas rutas asignada en el sistema, por tal razón siempre será visible y será independiente del contenido de cada componente mostrado en pantalla.

Para un usuario registrado se presentará formulario del inicio de sesión que constará de los campos usuario y contraseña mismos que son obligatorios, de no ser completados se presentaran alertas con mensajes informativos del motivo que las activo.

Figura 3.5 Formulario de inicio de sesión

Una vez que el usuario reciba sus credenciales podrá iniciar sesión ingresando los campos correo y contraseña, este formulario será manejado por *Hooks* de React que nos permitirán crear y reutilizar componentes que tengan un estado y dependan de un ciclo de vida, en este caso utilizaremos:

useReducer

Para poder controlar el inicio de sesión utilizaremos *authReducer* función reductora que maneja acciones de tipo *login* y *logout* respectivamente. Cuando se envía una acción de tipo *login*, el reductor actualiza el campo *logged* en el estado *true* y establece el campo *user* en la carga útil de la acción. Cuando se envía una acción de tipo *logout*, el reductor actualiza el campo *logged* en el estado *false* y establece el campo *user* en *null*. Si el tipo de acción no coincide con ninguno de los casos, el reductor devuelve el estado actual sin realizar ningún cambio.

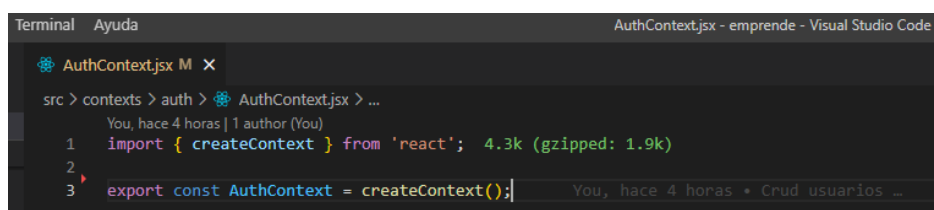


```
authReducer.js M X
src > contexts > auth > authReducer.js > ...
You, hace 4 horas | 1 author (You)
1 import { types } from './types';
2
3 export const authReducer = (state, action) =>
4 {
5
6   switch (action.type)
7   {
8     case types.login:
9       return {
10         ...state,
11         logged: true,
12         user: action.payload,
13       };
14
15     case types.logout:
16       return {
17         ...state,
18         logged: false,
19         user: null,
20       };
21
22     default:
23       return state;
24   }
25 }
```

Figura 3.6 Autenticación *Hook useReducer*

useContext:

Este código exporta un *AuthContext* como objeto que se crea utilizando la función *createContext* que es una función que crea un objeto de contexto que se puede usar para pasar datos a través de las rutas de los componentes sin necesidad las propiedades heredadas.



```
Terminal Ayuda AuthContext.jsx - emprende - Visual Studio Code
AuthContext.jsx M X
src > contexts > auth > AuthContext.jsx > ...
You, hace 4 horas | 1 author (You)
1 import { createContext } from 'react'; 4.3k (gzipped: 1.9k)
2
3 export const AuthContext = createContext();
```

Figura 3.7 Autenticación *Hook useContext*

useProvider:

En este caso se utilizó también un *AuthProvidercomponent* usa el enlace *useReducer* para administrar el estado de autenticación de la aplicación este proporciona las funciones *state* y *dispatch* como contexto para el resto de la aplicación, lo que permite que otros componentes accedan y actualicen el estado de autenticación.

```
AuthProvider.jsx X
src > contexts > auth > AuthProvider.jsx > @0 AuthProvider > @0 dispatch
You need to use <|> author (log)
1 import React, { useReducer } from 'react'; // 6.9k (gzipped: 2.7k)
2 import { AuthContext } from './AuthContext';
3 import { authReducer } from './authReducer';
4 import { types } from './types';
5 export const AuthProvider = ({ children }) =>
6 {
7   const initialization = () =>
8   {
9     const user = JSON.parse(localStorage.getItem('user'));
10    return {
11      logged: !!user,
12      user: user,
13    }
14  }
15  const [authState, dispatch] = useReducer(authReducer, [], initialization)
16  const login = async (user, token) =>
17  {
18    const action = { type: types.login, payload: user }
19    localStorage.setItem('user', JSON.stringify(user));
20    localStorage.setItem('token', token);
21    dispatch(action);
22  }
23  const logout = () =>
24  {
25    localStorage.removeItem('user');
26    localStorage.removeItem('token');
27    const action = { type: types.logout }
28    dispatch(action)
29  }
30  return (
31    <AuthContext.Provider value={{
32      ...authState,
33      login: login,
34      logout: logout,
35    }}>
36      {children}
37    </AuthContext.Provider>
38  );
39 }
40 }
```

Figura 3.8 Autenticación Hook *useProvider*

Una vez creada la lógica para el inicio de sesión ingresaremos nuestros datos en el formulario de inicio de sesión. Estos datos serán enviados como una petición al servidor establecido como backend con el uso de Axios para enviar una solicitud *POST* al *APIRest* con un objeto que contiene *email* y contraseña como cuerpo de la solicitud. El bloque *try* contiene el código que se ejecuta cuando la solicitud es exitosa, posteriormente desestructura los datos de respuesta para obtener las propiedades *access_token*, *token_type*, *user* y se guardan en una constante. Como último paso de este proceso se le envía como argumentos *user* y *`\${token_type} \${access_token}* a la función *login* que su funcionamiento fue mencionado en la **Figura 3.7** si la autenticación fue exitosa navega a la ruta *'/administracion'*. El bloque *catch* contiene el código que se ejecuta cuando falla la solicitud. Si el error no es un error de red, registra el error en la consola. Si el error es un

error de red, registra el mensaje de error de los datos de respuesta y borra las variables de estado *email* y *contraseña*.

```
const onLogin = async (e) => {
  e.preventDefault();

  try {
    const response = await axios.post('https://backend-emprende.herokuapp.com/api/v1/login', {
      email,
      password
    });
    const { access_token, token_type, user } = response.data.data
    console.warn(access_token, token_type, user);

    login(user, `${token_type} ${access_token}`);
    // eslint-disable-next-line no-lone-blocks
    {
      navigate('/administracion');
    }
  } catch (error) {
    if (!isNetworkWired) {
      console.log(error);
    } else {
      console.log(error.response.data.message, error);
      setEmail('');
      setPassword('');
    }
  }
}
```

Figura 3.9 Envío de datos de autenticación con AXIOS.

Con los datos de autenticación enviados se podrá controlar en función a los roles que es lo que se va a mostrar en pantalla, para el caso de super administrador quien tendrá acceso a todo el panel de administracion y para el administrador solamente tendrá acceso al contenido de administracion de emprendimientos. En el código de la **Figura 3.10** se puede observar la condición que nos permite separa las funciones.

```
return (
  <Row className='mt-2 d-flex justify-content-between'>
    {user && user.rol === 'superadmin' && (
      <Card className={classes.root}>
        <CardActionArea> ...
      </CardActionArea>
      <CardActions>
        <Button size="small" color="primary"> ...
      </Button>
      </CardActions>
    </Card>
    )}
    {user && user.rol === 'superadmin' && (
      <Card className={classes.root}> ...
    </Card>
  )}
  <Card className={classes.root}> ...
</Card>
</Row>
```

Figura 3.10 Funciones dependiendo el rol

Como se observa **Figura 3.5** se presenta también un botón que nos permitirá recuperar la contraseña mediante el correo electrónico registrado en el sistema, este proceso de lo realizará con modales que se presentará sobre la aplicación reduciendo así la carga al controlador de rutas, al igual que el componente anterior utilizamos la misma librería y lógica de alertas, proceso que será demostrado en próximas iteraciones.



Figura 3.11 Recuperación de contraseña

Como se puede visualizar en la **Figura 3.11** si nosotros ingresamos un correo registrado como parte conjunta con el backend recibiremos un mensajes con el link que nos permitirá recuperar la contraseña, internamente este proceso envía una petición al funciones creadas en un *endpoint* que como respuesta recibiremos un correo con los datos necesarios para poder recuperar nuestra contraseña como los son el token del usuario y el correo ligado a este, en la **Figura 3.12** se puede observar el correo recibido que nos proporciona de un enlace para acceder a nuestro reseteo de contraseña.



Figura 3.12 Correo de notificación

Una vez que demos clic en este campo nos redirigirá nuevamente a nuestra aplicación junto a un formulario el cual nos pedirá que ingresemos una nueva contraseña para nuestro usuario. Para una buena experiencia para el usuario se presenta también los requisitos que deben tener las contraseñas para que sean aceptadas por el sistema, una vez que ingresemos nuestra contraseña seremos redirigidos al formulario para ingresar al sistema.



El formulario muestra un ícono de un ordenador con un candado y una llave, simbolizando la seguridad y el restablecimiento de la contraseña. El título es 'EMPRENDE' y el subtítulo es 'Establece una nueva contraseña'. Hay dos campos de entrada: 'Contraseña' y 'Confirmar Contraseña'. Debajo de los campos hay un botón 'ENVIAR'. En la parte inferior, se detallan los requisitos de la contraseña: 'La contraseña debe contener al menos una mayúscula y una minúscula, La contraseña debe contener al menos 8 caracteres'.

Figura 3.13 Restablecimiento de contraseña

Panel de Administración

En la interfaz gráfica de nuestra aplicación podremos observar en el caso de los superadministradores los apartados en los cuales podremos realizar el *CRUD* respectivo, para en esta aplicación se podrán encontrar dos roles, el Superadministrador y el Administrador mismos que tendrán diferentes funciones como se observa en la **TABLA VIII**.

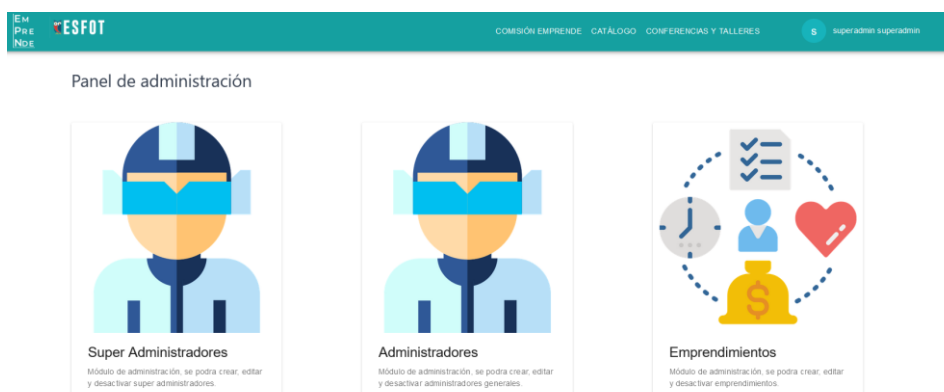


Figura 3.14 Panel de administración

TABLA VIII: Funciones según el rol

Rol de usuario	Funciones
Superadministrador	<ul style="list-style-type: none">• Crear usuarios• Cambio de información de los usuarios• Desactivar y desactivar usuarios• Creación de emprendimientos• Aprobar, activar y desactivar emprendimientos
Administrador	<ul style="list-style-type: none">• Creación de emprendimientos• Aprobar, activar y desactivar emprendimientos

En código escrito en JSX, que es una extensión de sintaxis para *JavaScript* que nos permite escribir código similar a HTML en los archivos *JavaScript* podemos representar una lista de elementos (en este caso, superadministradores) como tarjetas en la aplicación. Cada tarjeta contiene información sobre un superadministrador y los botones respectivos para poder editar, desactivar y crear, mismos que con el uso de modales como componentes reutilizables desplegaremos un formulario para crear nuevos administradores, editar y desactivarlos. Hay que tener en cuenta que el administrador que este con una sesión activa no podrá desactivar su propio usuario.

```
<Card.Body className="text-center">
  {admin.id !== user.id && (
    <Button
      variant="primary"
      className="text-center"
      onClick={() => deleteSuAdmin(admin.id)}
    > {admin.state === 1 ? 'Desactivar' : 'Activar'}</Button>
  )}
  {
    admin.id === user.id && (
      <ModalForm buttonLabel="Perfil" />
    )
  }
</Card.Body>
</Card>
```

Figura 3.15 Control de actividad por id de usuario

Este código se puede observar que según id del usuario activo se presentara un icono que tiene un formato “*svg*” que estilizado con *css* y tendrá cierta animación para que el usuario sepa que el puntero del mouse esta sobre él, la etiqueta “Activo” o “Inactivo” dependerá del valor del estado del usuario en sesión, para representar a un usuario se utilizaran estilos que darán forma a los contenedores y mediante el mapeo de la información obtenida del *endpoint* se cargara la información de los usuarios registrados, dependiendo el estado de cada uno los botones se podrán visualizar junto a su acción.

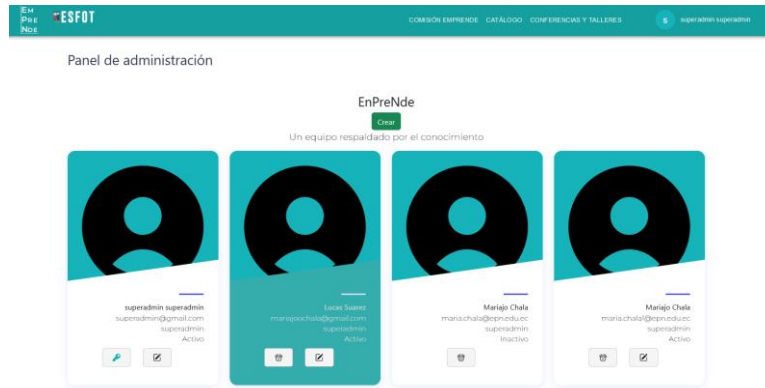


Figura 3.16 Vista de superadministradores

Para poder realizar el *CRUD* de los superadministradores creamos un componente funcional llamado *ModalForm* que muestra un modal con un formulario para crear o editar un nuevo administrador. Cuando se procesa el componente, muestra un botón con la etiqueta especificada en nuestro elemento llamado *buttonLabel* y le pasa las características esperadas de cada usuario y cuando se hace clic en el botón, se abre el modal.

El modal contiene un encabezado con la misma etiqueta que el botón, y el cuerpo del modal contiene el componente *EditForm* en el que esta codificado un formulario para este propósito. Cuando se llama al componente con el elemento llamado *buttonLabel* que está establecido en el botón "Crear", generará un formulario para crear un nuevo superadministrador.

El componente *Modal* es de la biblioteca *react-bootstrap* se usa para mostrar un modal en la página *web*. Los componentes *ModalHeader*, *ModalBody* y *Button* también son de *react-bootstrap* y se utilizan para mostrar el encabezado, el cuerpo y el botón, respectivamente, quienes forman parte del modal.

```
return (
  <div>
    <Button color="success" onClick={toggle}>
      {props.buttonLabel}
    </Button>
    <Modal isOpen={modal}
      toggle={toggle}
      className={props.className}
      backdrop="static"
      keyboard={false}
    >
      <ModalHeader toggle={toggle}>{props.buttonLabel}</ModalHeader>
      <ModalBody>
        <EditForm
          updateState={props.updateState}
          toggle={toggle}
          item={props.item}
          admins={admin}
        />
      </ModalBody>
    </Modal>
  </div>
);
```

Figura 3.17 Creación de Modal para creación de usuario

Este formulario presenta varios campos para el registro cada uno deberá ser llenados debido que son datos fundamentales para cada administrador, al momento de su envío la aplicación presentara alertas en el caso de que un campo este vacío o presente datos inválidos.



The image shows a web form titled "Formulario de Registro" (Registration Form) within a modal window labeled "Crear" (Create). The form contains the following fields and validation messages:

- Username:** Input field with placeholder "Ingrese el nombre". Validation message: "Debe ingresar un nombre válido!" (Must enter a valid name!).
- Apellido:** Input field with placeholder "Ingrese el apellido". Validation message: "Debe ingresar un apellido válido!" (Must enter a valid surname!).
- Email:** Input field with placeholder "Email". Validation message: "Email no válido!" (Email is not valid!).
- Teléfono personal:** Input field with placeholder "Teléfono". Validation message: "Solamente se aceptan números de 9 a 10 dígitos" (Only numbers from 9 to 10 digits are accepted).
- LinkedIn:** Input field with placeholder "LinkedIn".

At the bottom of the form are two buttons: "Crear" (Create) and "Limpiar" (Clear).

Figura 3.18 Formulario de registro

Para este y los demás procesos que tengan que ver con un *CRUD* existe una función que se llama cuando se envía el formulario. Realiza una validación básica para asegurarse de que se completen todos los campos obligatorios y luego realiza una *POST* solicitud al *endpoint* con los datos del formulario. Si él *endpoint* devuelve una respuesta de éxito, se muestra un mensaje de éxito utilizando una alerta. Si hay un error, se muestra el mensaje de error correspondiente.

Una vez creado el usuario recibirá un correo electrónico a la dirección registrada en la que se proporcionará una contraseña provisional el nombre, rol y un enlace que lo redirigirá a la página de inicio de sesión como se observa en la **Figura 3.19**, en este punto el usuario deberá cambiar su contraseña para poder tener mayor control sobre ella este paso se lo explicará en la sección de administradores.

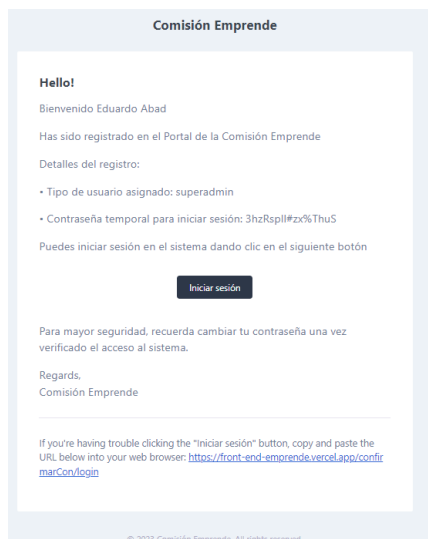


Figura 3.19 Mensaje de bienvenida

Para el caso de la edición de un usuario funcionará con la misma lógica ahora al “*buttonLabel*” se le enviara el nombre Editar tomando en cuenta que se tendrá que pasar un “*id*” de usuario para que se pueda direccionar la edición de datos relacionados al usuario y que se cargaran en el formulario. Una vez cargada la información el administrador podrá cambiar los datos de cualquier usuario registrado incluyéndose, con la ayuda de operadores ternarios el usuario que este iniciado sesión tendrá la oportunidad de cambiar su foto de perfil y su contraseña.

The image shows a form titled 'Editar' for editing user information. The form contains several input fields: 'Username' (superadmin), 'Apellido' (superadmin), 'Email' (superadmin@gmail.com), 'Teléfono personal' (0999999998), and 'Linkedin' (https://linkedin.superadmin.com). There is a blue 'Guardar' button. Below the form, there is an 'Imagen' section with a 'Seleccionar archivo' button and a 'Sin archivos seleccionados' status, along with a blue 'Subir imagen' button.

Figura 3.20 Edición de información usuario

En este módulo no se eliminarán usuarios, el proceso que se podrá realizar es la desactivación, esto para que los datos no se eliminen y en el caso de que el usuario requiera el acceso bastaría con la reactivación, este proceso se lo realiza juntamente con

un *endpoint* que nos recibe el “*id*” de usuario e internamente cambia su estado. En cada tarjeta de usuario se presenta un botón que al momento de colocar el puntero del ratón indicará la acción que el usuario podrá hacer.

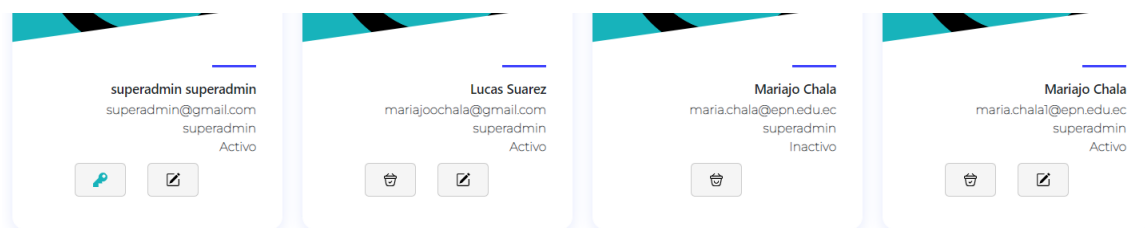


Figura 3.21 Acciones en función del estado.

Tomando en cuenta que los administradores tienen funciones reducidas no se podrá presentar la misma vista es por esto por lo que se creara una condición en la que se comparara su rol para mostrar una vista diferente acorde a su usuario como se observa en la **Figura 3.22**.

```
{user && user.rol === "admin" && (  
  <Card className={classes.root}>  
    <CardActionArea>  
      <CardMedia  
        className={classes.media}  
        image="https://cdn-icons-png.flaticon.com/512/1705/1705768.png"  
        title="emprendimiento"  
      />  
    </div className="d-flex justify-content-center w3-padding">  
      <ModalPswd />  
      <ModalAvatar />  
      <ModalInfo />  
    </div>  
    <CardContent>  
      <Typography gutterBottom variant="h5" component="h2">  
        Perfil de administrador  
      </Typography>  
      <Typography variant="body2" color="textSecondary" component="p">  
        Podra visualizar sus datos actualizar su perfil y cambiar su  
        contraseña.  
      </Typography>  
    </CardContent>  
  </CardActionArea>  
</Card>  
)}
```

Figura 3.22 Operador ternario administrador

De este código se genera la vista en la que el usuario podrá acceder a su información, para editarla utilizando la misma lógica que con los superadministradores explicada en la **Figura 3.17**, para este usuario el cambio de contraseña y la imagen de presentan formularios independientes para cada acción como se observa en la **Figura 3.22** que se activarán dependiendo de la necesidad del usuario como paso obligatorio en el caso de que sea un nuevo usuario deberá cambiar la contraseña. El modal que presenta el formulario se despliega luego de presionar el botón de cambio de contraseña como se observa en la **Figura 3.23**

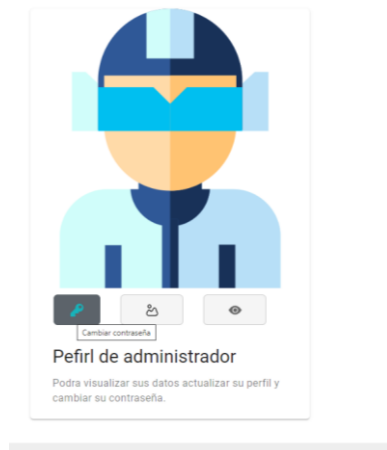


Figura 3.23 Cambio de contraseña

Una vez desplegado el formulario mismo que tendrá dos campos obligatorios, una vez que se ingrese la información adecuada se enviara a un *endpoint* que verificara si existe un usuario con sesión activa y a través de su *token* se realizara el cambio de contraseña, tomando en cuenta que este campo es parte fundamental de una sesión se procederá a cerrar la sesión para poder aplicar el cambio. Para una buena experiencia del usuario en el pie del modal se presentará un mensaje con información de la forma que debe tener la contraseña a modificar.

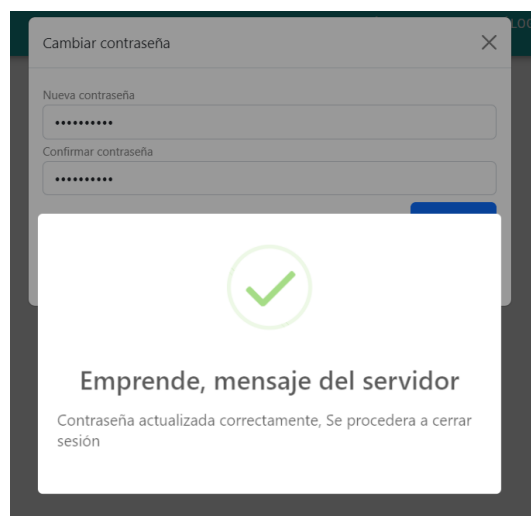


Figura 3.24 Cambio de contraseña

3.3 Iteración 2

La iteración se complementa con las siguientes historias de usuario como se observa en la **TABLA IX**

TABLA IX: Iteración 2

Iteración	Historia de usuario #	Descripción
2	HU006 – Administración de emprendimientos.	Como administrador se desea consumir un <i>endpoint</i> que permita autorizar, crear, visualizar y eliminar información de los emprendimientos y actualizar el catálogo de emprendimiento.
2	HU007 - Validación de campos en Formularios	Como administradores se desea validar los campos de los formularios antes de enviar al <i>endpoint</i> .

Administración de emprendimientos

Para este componente de Emprendimientos mostraremos una lista, una vez que tengamos *user* del *AuthContext* inicializaremos una variable de estado llamada *token* y recuperaremos el valor del token propio de cada usuario almacenamiento local.

El componente tiene dos funciones: *getData* y *deleteEmprendimiento*. La función *getData* envía una solicitud *GET* a un extremo de la *API* para recuperar una lista de emprendimientos y actualiza la variable *data* con los datos de respuesta. La función *deleteEmprendimiento* envía una solicitud *GET* a un extremo de la *API* con la especificación del *id* para eliminar un emprendimiento y luego llama a la función *getData* para actualizar la lista de emprendimientos.

```
const getData = async () => {
  try {
    const response = await axios.get(
      'https://backend-emprende.herokuapp.com/api/v1/emprendimiento',
      { headers: { 'accept': 'application/json', 'authorization': token } }
    );
    setData(response.data.data.emprendimientos)
  } catch (error) {
    console.log(error);
  }
}

const deleteEmprendimiento = async (id) => {
  try {
    const response = await axios.get(
      'https://backend-emprende.herokuapp.com/api/v1/emprendimiento/${id}/destroy',
      { headers: { 'accept': 'application/json', 'authorization': token } }
    );
    getData();
  } catch (error) {
    console.log(error);
  }
}
```

Figura 3.25 Lógica de CRUD emprendimientos

La tabla en la que se presentaran los emprendimientos actualmente se representa con el componente *Table* de la biblioteca *react-bootstrap*. El cuerpo de la tabla se llena con una fila para cada emprendimiento en el arreglo *data*. Cada fila muestra la categoría, nombre, descripción e imagen del emprendimiento junto a la columna de acciones, la cual presentará información relevante y permitirá la activación, desactivación y visualización de la información del emprendimiento registrado.






Emprendimientos				
<input type="text" value="Buscar"/>				
Categoría	Nombre	Descripcion	Imagen	Acciones
OTROS	HAMA BEEADS	EMPRESARIO DE HAMA BEADS		<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">🗑️</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">✎</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">👁️</div> </div>
Educación	NUEVO NOMBRE	Emprendimiento de Hama Beads		<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">🗑️</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">✎</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">👁️</div> </div>
Manualidades	Hama Beads	Emprendimiento de Hama Beads		<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">🗑️</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">✎</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">👁️</div> </div>
Manualidades	Hama Beads	Emprendimiento de Hama Beads		<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">🗑️</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">✎</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">👁️</div> </div>
Tecnología	Canoa Restaurante	Desayunos, almuerzos, bebidas y picaditas.		<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">🗑️</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">✎</div> <div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc;">👁️</div> </div>

Figura 3.26 Tabla administración de emprendimientos

Para cada emprendimiento, hay un botón que muestra "Desactivar" o "Activar", según el valor del estado del emprendimiento. Al hacer clic, icono activará la función *deleteEmprendimiento*, que envía una solicitud para eliminar o activar el emprendimiento en particular. También iconos que, al ser presionado, activará el componente *ModalEmp*, que muestra un formulario para editar la información de un emprendimiento y así mismo para poder visualizarlo.

```

{dato.map((item) => {
  <tr key={item.id}>
    <td>{item.nombre}</td>
    <td>{item.descripcion}</td>
    <td><img src={item.image} alt="" width="100px" /></td>
    <td > {item.estado === 1 ? (
      <button className='btndesactivar' onClick={() => deleteEmprendimiento(item.id)} title="
        <path d="M12.643 15C13.979 15 15 13.845 15 12.5V5H17.5C1 13.845 2.021 15 3.357 15H
      </svg></button>
    ) : (
      <button className="btn btn-danger" onClick={() => deleteEmprendimiento(item.id)} title="
        <path d="M12.643 15C13.979 15 15 13.845 15 12.5V5H17.5C1 13.845 2.021 15 3.357 15H
      </svg></button>
    )
  }
  <ModalEmp emprendimientos={item} data={data} />
  <ModalViewEmp emprendimientos={item} data={data} />
</td>
</tr>
)}

```

Figura 3.27 Acciones permitidas

Para el componente *ModalNewEmp* que muestra un modal con un formulario para crear un nuevo emprendimiento muestra un botón con la etiqueta especificada en la *buttonLabel* que con el uso de *props* enviamos las propiedades específicas de los emprendimientos y cuando se hace clic en el botón con la etiqueta Crear, se abre el modal.

El modal contiene un encabezado con la misma etiqueta que el botón, y el cuerpo del modal contiene el componente *EditForm*. Cuando el componente *ModalNewEmp* es llamado con el *buttonLabel* establecido en "Crear", renderizará un formulario para crear un nuevo emprendimiento.

Figura 3.28 Formulario nuevo emprendimiento

Para la función de editar el emprendimiento ya registrado se utilizará el mismo formulario con la diferencia que en el código cargaremos los datos del emprendimiento de forma automática solamente con el uso del id mediante peticiones *HTTP* con *AXIOS*.

```
const onSubmit = async (e) => {
  e.preventDefault();
  await axios.post(
    `https://backend-emprende.herokuapp.com/api/v1/emprendimiento/${emprendimientos.id}/update`,
    {
      ...form,
    },
    { headers: { accept: "application/json", authorization: token } }
  ).then(response => {
    const res = response.data;
    toggle();
  })
};

const handleUpload = (e) => {
  e.preventDefault();

  const formData = new FormData();
  formData.append('image', image);

  axios.post(`https://backend-emprende.herokuapp.com/api/v1/emprendimiento/${emprendimientos.id}/logo`, formData, {
    headers: { Authorization: `Bearer ${token}`, 'Content-Type': 'multipart/form-data' },
  })

  .then(response => {
    const res = response.data;
    //console.log(res);
    toggle();
  })

  .catch(error => {
    console.log(error);
  });
}
```

Figura 3.29 Actualización de emprendimientos

Para el proceso de eliminar tomaremos en cuenta que debemos tener guardado el registro de todos los emprendimientos registrado puesto que el usuario puede retomar las actividades, es por esto por lo que solamente lo deshabilitaremos, los elementos que están en la tabla junto con un botón con clase *"btn btn-info"* activará o desactivará un emprendimiento dependiendo de su estado. La función *onClick* del botón activa una función llamada *"deleteEmprendimiento"* y pasa la propiedad *"id"* del objeto *"emprendimientos"* como argumento. El operador ternario se usa para verificar el estado del emprendimiento y se traducirá como *"Activar"* o *"Desactivar"* dependiendo del estado.

```
<TableCell align="right"> {item.estado === 1 ? (
  <button className="btn btn-info" onClick={() => deleteEmprendimiento(item.id)}>Desactivar</button>
) : (
  <button className="btn btn-danger" onClick={() => deleteEmprendimiento(item.id)}>Activar</button>
)}
</td>
```

Figura 3.30 Activación y desactivación de emprendimientos

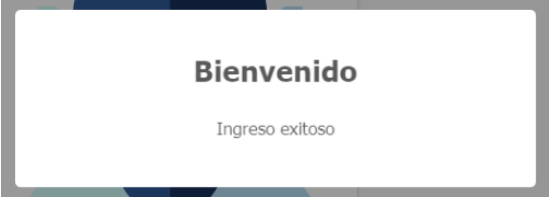
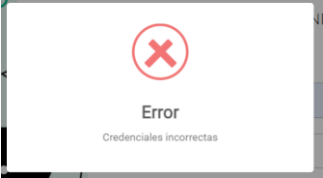
Validaciones

En la creación de formularios es muy importante la validación de los campos y así asegurar la integridad de los datos evitando que se introduzcan datos inválidos. Aportando también a la experiencia del usuario brindándole la oportunidad de corregir los errores sin la

necesidad de volver a completar el formulario, por ultimo y lo más importante, esto permite que evitemos la inyección de códigos malicioso manteniendo la seguridad del sistema.

Para el módulo de inicio de sesión se observará varios tipos de advertencias dependiendo la acción del usuario como se observa en la **TABLA X**, tomando de ejemplo el ingreso exitoso y el ingreso de credenciales incorrectas, las demás alertas están detalladas en el **ANEXO II** apartado Validaciones.

TABLA X. Validación de formulario *Login*

Acción	Validación
Ingreso de credenciales registradas en el sistema.	 <p>Figura 3.31 Ingreso al sistema</p>
Ingreso de credenciales incorrectas en el sistema.	 <p>Figura 3.32 Credenciales incorrectas</p>

Teniendo en cuenta que en el formulario de inicio de sesión existen solamente dos campos se optó por validación en función de alertas basadas en expresiones regulares, en el caso de los formularios creados para realizar los *CRUD*'s se cambiara la forma de validación, en estos se ingresaran distintos tipos de datos los cuales deben ser controlados de manera independiente antes de ser enviados. Para la validación de los datos utilizaremos estas expresiones de la **TABLA XI** para indicar en cada campo si presenta un error.

TABLA XI: Expresiones regulares

Expresión regular	Función
<code>/^[a-zA-ZÀ-ÿ\s] {3,16}\$/</code>	Controla que los caracteres ingresados sean letras, mayúsculas o minúsculas, acepta espacios, acentos y una longitud desde 3 a 16 caracteres.
<code>/^\d{9,10}\$/</code>	Controla que los caracteres ingresados sean números con una longitud de 9 a 10 caracteres.
<code>/^(https?:\w)?(\wda-z.-+)\.([a-z.]{2,6})([\w.-]*)*V?\$/</code>	Controla que los caracteres ingresados coincidan con las URL.
<code>/^[a-zA-Z0-9s_ll\:-:)]+(.jpg .jpeg .gif .png .bmp)\$/</code>	Controla que los archivos que se quieren enviar en el formulario tengan extensiones adecuadas para una imagen válida.
<code>^[a-zA-Z0-9_.\$!%-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2}\$</code>	Controla que los caracteres coincidan con una dirección de correo electrónica válida.

En el caso de la creación de usuarios se presenta un formulario en los que se deberá ingresar datos personales necesarios para poder generar la petición de creación del usuario, si en cualquier campo se genera un error ya sea por información inválida o campos vacíos se observará las alertas que se presentan en la **Figura 3.18** Formulario de registro, en el **Anexo II** apartado validaciones se presentará con más detalles las alertas en los demás formularios.

3.4 Iteración 3

En la **TABLA XII** esta iteración se complementa con las siguientes historias de usuario.

TABLA XII: Iteración 3

Iteración	Historia de usuario #	Descripción
3	HU008 - Visualización de emprendimientos	Como usuarios se desea ver una lista de emprendimientos con información relevante

3	HU009 - Formularios de solicitud.	Como administradores se desea consumir un <i>endpoint</i> que permita enviar solicitudes para poder formar parte del catálogo de la comisión.
---	-----------------------------------	---

Visualización de emprendimientos

Para que un usuario pueda visualizar los emprendimientos registrados y autorizados no será necesario que tenga un usuario bastaría solamente con ingresar al menú catalogo en la barra de navegación, este módulo este realizado con estilos basados en W3.css que nos dan la posibilidad de crear un carrusel de imágenes y tarjetas que presentaran los usuarios que tengan los estados de validación activos.

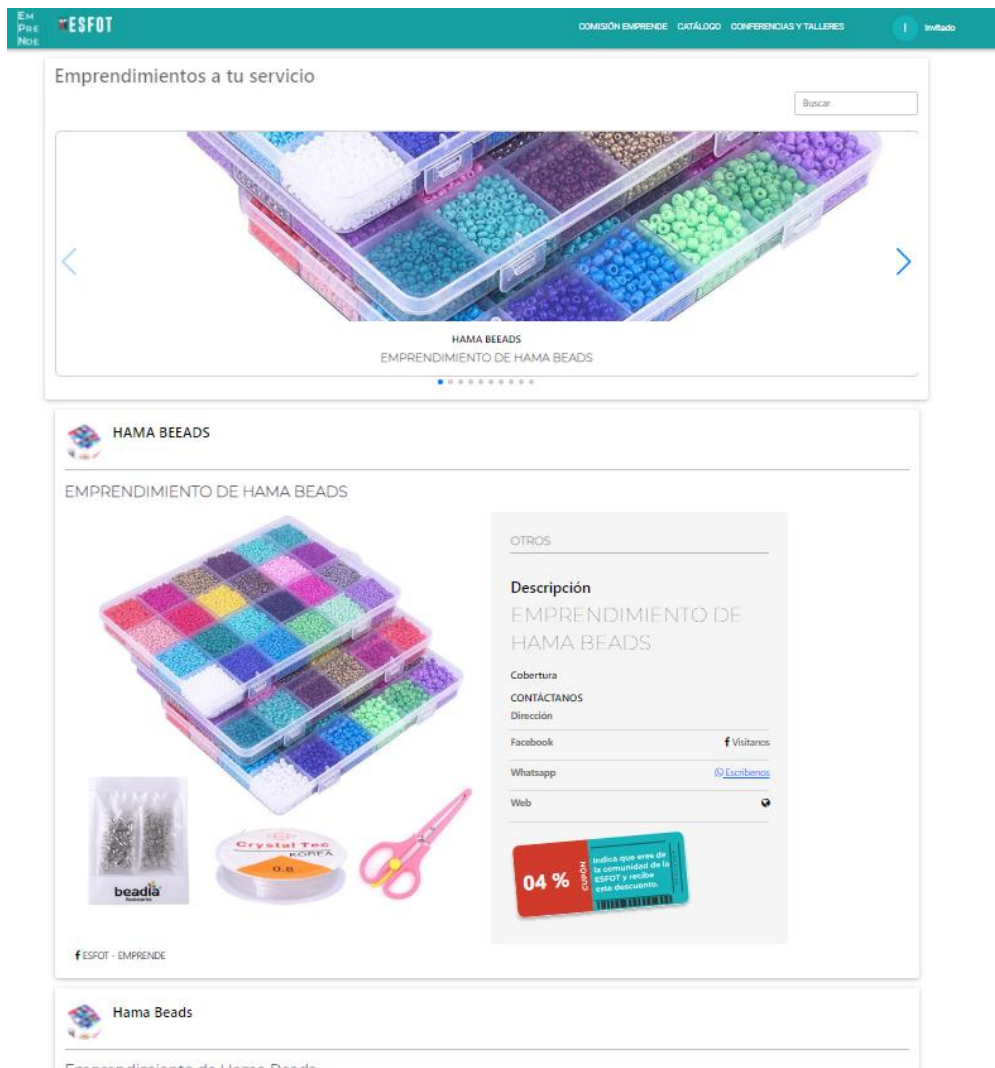


Figura 3.33 Catálogo de emprendimientos

Como se observa en la **Figura 3.34** la función `map()` toma una función argumento y la llama para cada elemento de la matriz que contiene los datos de los emprendimientos. El código dentro de la función comprueba si las propiedades `estado` y `estado1` del elemento actual son iguales a 1. Si esta condición es verdadera, se presenta un elemento que contiene otros elementos HTML que muestran información sobre el emprendimiento.

```
{dato.map((item) => You, hace 3 días • Descuentos
  item.estado === 1 && item.estado1 === 1 ? (
    <div>
      <div
        className="w3-container w3-card w3-white w3-round w3-margin"
        style={{ marginBottom: 10 }}
      >
```

Figura 3.34 Validación de estados

Formulario de solicitud

Los miembros de la comunidad de la ESFOT que tengan un emprendimiento y quieran dar a conocer algún producto tendrá a la mano un formulario en la página principal de la aplicación de la comisión *Emprende* por el cual podrá ingresar una petición para poder ingresar su emprendimiento a la base de datos de la comisión.

The image shows a web form titled "Petición Emprendimiento". It has a close button (X) in the top right corner. The form is organized into a grid of input fields:

- Row 1: "Rol Esfot" (dropdown), "Nombre" (text), "Descripcion" (text)
- Row 2: "Categoria" (dropdown), "Direccion" (text), "Cobertura" (dropdown)
- Row 3: "Pagina Web" (text), "Telefono" (text), "Whatsapp" (text)
- Row 4: "Facebook" (text), "Instagram" (text), "Descuento" (text)
- Row 5: "Imagen promocional" (file upload) with "Seleccionar archivo" and "Sin...dos" buttons.

At the bottom left is a blue "Guardar" button, and at the bottom right is a grey "Cerrar" button.

Figura 3.35 Formulario de petición

Este formulario tiene la misma lógica del formulario de creación de emprendimientos con la diferencia que en la petición de creación enviada al *endpoint* por defecto estará un segundo estado como desactivado, este se utilizará para poder separar la administración de los mismos que su valor que ira guardando al momento de realizar la petición al *endpoint* como se observa en la **Figura 3.36** el código recorre cada elemento de la matriz

emprendimientos y cuenta la cantidad de elementos donde estado1 es igual a 0. El conteo resultante se almacena en la variable peticiones variable Pendientes.

```
const getData = async () => {
  setLoading(false);
  try {
    const response = await axios.get(
      `https://backend-emprende.herokuapp.com/api/v1/emprendimiento`,
      {
        headers: { accept: "application/json", authorization: token },
      }
    );
    let pend = 0;
    setData(response.data.data.emprendimientos);
    for (let i = 0; i < response.data.data.emprendimientos.length; i++) {
      if (response.data.data.emprendimientos[i].estado1 === 0) {
        pend = pend + 1;
      }
    }
    setPendientes(pend);
    setLoading(true);
  } catch (error) {
    console.log(error);
  }
};
```

Figura 3.36 Formularios pendientes de aprobación

Para que el usuario pueda tener información sobre la cantidad de peticiones se presentara una notificación en la pestaña establecida para las peticiones esta toma el valor de los emprendimientos que tengan un estado específico y mediante un “*span*” y estilos css se presentara en la esquina superior derecha de la pestaña.



The screenshot shows a web interface with two tabs: 'Emprendimientos Aprobados' and 'Solicitudes'. The 'Solicitudes' tab is active and has a red notification badge with the number '3' in the top right corner. Below the tabs is a search bar labeled 'Buscar'. The main content is a table with the following data:

Categoría	Nombre	Descripción	Imagen	Acciones
Manualidades	Hama Beads	Emprendimiento de Hama Beads		  
Tecnología	Carnes La Familia	Carnicería familiar con variedad de productos cárnicos, huevos, y derivados.		  

Figura 3.37 Notificación de peticiones

Una vez que el administrador considere que la petición puede ser actualizada utilizara la misma función para la desactivación de emprendimientos, pero tomando en cuenta que es un estado al que no podrá regresar, es por esto por lo que también se presenta al usuario la posibilidad de editar y visualizar la petición para tomar una decisión acertada.

3.5 Iteración 4

Esta iteración se complementa con las siguientes historias de usuario como se observa en la **TABLA XIII**.

TABLA XIII: Iteración 4

Iteración	Historia de usuario #	Descripción
4	HU009 - Filtros en catálogo.	Como administradores se desea consumir un <i>endpoint</i> que permita filtrar el catálogo de emprendimientos por tipo de producto.
4	HU010 – Videoconferencias comisión Emprende.	Como administradores se desea consumir un <i>endpoint</i> que despliegue los videos de conferencias dictadas por la comisión.

Filtros de búsqueda de emprendimientos.

En este módulo se podrá contar con una barra de búsqueda que recibirá los datos que el usuario ingrese y la pasará al a función del "searcher" que se pasa como un controlador de eventos para el elemento de entrada este actualizará el estado de la variable de búsqueda con el valor del elemento de entrada como función básica de un "hook".

```
const [data, setData] = useState([])
const token = localStorage.getItem('token')

const [search, setSearch] = useState("")

const searcher = (e) => {
  setSearch(e.target.value)
}
```

Figura 3.38 Buscador React

Para poder filtrar según lo que el usuario quiera visualizar, se utiliza el operador ternario para filtrar una matriz de datos en función de un término de búsqueda. La variable "search" se usa para filtrar la matriz de "dato", si "search" es verdadera y necesariamente no es nula o indefinida, filtrará la matriz y solo devolverá los elementos donde la propiedad "nombre", en este caso sería el nombre del emprendimiento registrado del elemento incluye el valor de "search". Si la "búsqueda" es falsa es decir es nula o indefinida, se devolverá la matriz de "datos" original. El método "toLowerCase()" se usa tanto en el término de búsqueda

como en la propiedad "nombre" para que la comparación no distinga entre mayúsculas y minúsculas.

```
const dato = !search ? data : data.filter((dato) => dato.nombre.toLowerCase().includes(search.toLocaleLowerCase()))

useEffect(() => {
  |   getData();
  }, []);
```

Figura 3.39 Función base de filtrado

Módulo de videoconferencias

En esta sección se presentarán todos los videos que fueron grabados en los eventos realizados por la Comisión Emprende, para poder almacenarlos se creó un formulario que acepta tres campos que son: título del video, descripción y la dirección web en la que podemos ubicarla, en este caso la mayoría de videos esta alojados en la red social Facebook misma que nos provee de un enlace adecuado para nuestro propósito, esta información será enviada al endpoint que en el caso de tener una respuestas sin errores el video se podrá reproducir en la aplicación.

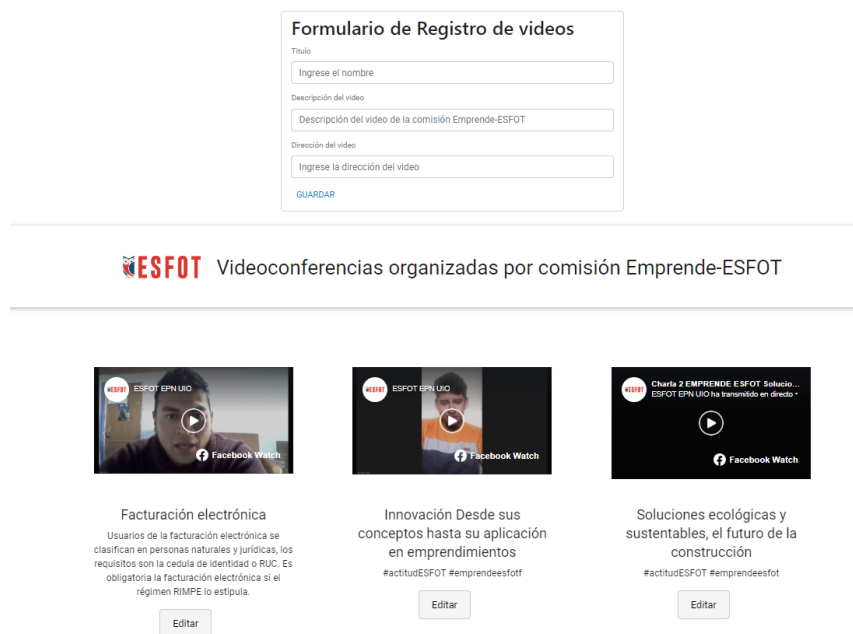


Figura 3.40 Repositorio de videos

Como se observa en la Figura 3.40 se presenta al usuario un botón para poder editar esta información en el caso de que el enlace no esté disponible o que la información ingresada este incompleta. Utilizando la misma lógica utilizada para los componentes anteriores se

enviará el “id” del video junto una petición de edición al *endpoint* misma que al procesarla y recibir una respuesta positiva nos presentará un mensaje de conformación.

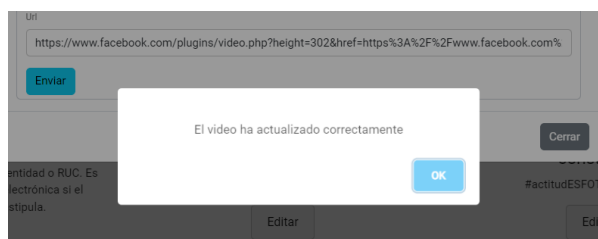


Figura 3.41. Editar información del repositorio

3.6 Iteración 5

Pruebas

Las pruebas en el frontend son una parte esencial al momento de la creación de aplicaciones *web* puesto que estas ayudaran a garantizar tanto la calidad como la estabilidad de la aplicación, además de ayudar a facilitar la mantenibilidad, a encontrar errores y sobre todo a asegurar la funcionalidad del sistema. Esta sección tiene como objetivo demostrar la funcionalidad del sistema mediante el uso de diversas pruebas que, aplicadas correctamente nos permitirán demostrar el cumplimiento de los requerimientos recopilados.

Pruebas de aceptación

Las pruebas de aceptación sirven para evaluar si un sistema cumple con los requerimientos especificados por el cliente y comprobando que cada componente este listo para su uso asegurando la calidad y la confianza en el sistema, estas pruebas servirán como referente en el caso de que existieren problemas antes de su implementación [18]. Las pruebas de aceptación para la aplicación *web* se realizaron junto al Ing. Abraham Loja representante de la comisión Emprende para este proyecto obteniendo como resultado un sistema funcional. Se puede evidenciar con más detalle el resultado de esta prueba en el **ANEXO II**. Apartado Pruebas de aceptación.

Pruebas de compatibilidad.

En Las pruebas de compatibilidad entre navegadores *web* es uno de los pasos necesarios dentro de la implementación de un sistema frontend. Estas pruebas se realizan para garantizar que la aplicación funcione correctamente en distintos navegadores comprobando que el diseño y la funcionalidad se muestran correctamente, esto se demuestra en la **TABLA XIV**.

TABLA XIV: Compatibilidad de navegadores

Navegador	Versión	Dispositivo	Resultado
Firefox	109.0 64-bit	Escritorio	Compatible
Chrome	109.0.5414.75 64-bits	Escritorio	Compatible
Microsoft Edge	109.0.1518.61 64-bits	Escritorio	Compatible
Chrome	109.0.5414.86	Tablet 7"	Compatible
Chrome	105.0.5195.102 32-bits	Escritorio	Compatible
Brave	1.47.171 Chromium:109.0.5414.87	Escritorio	Compatible

Pruebas Unitarias

Para esta sección se realizarán pruebas unitarias que ayudarán a comprobar el correcto funcionamiento de los módulos creados de forma individual, esto evitara que en el proceso del desarrollo del sistema se pueda evitar problemas graves en su codificación. Con el uso de *Cypress* automatizaremos estas pruebas observando el flujo que el usuario pueda generar.

Se iniciará con el módulo de inicio de sesión, se espera que el usuario ingrese sus credenciales proporcionadas por el sistema y que pueda observar la pantalla del panel de administracion.

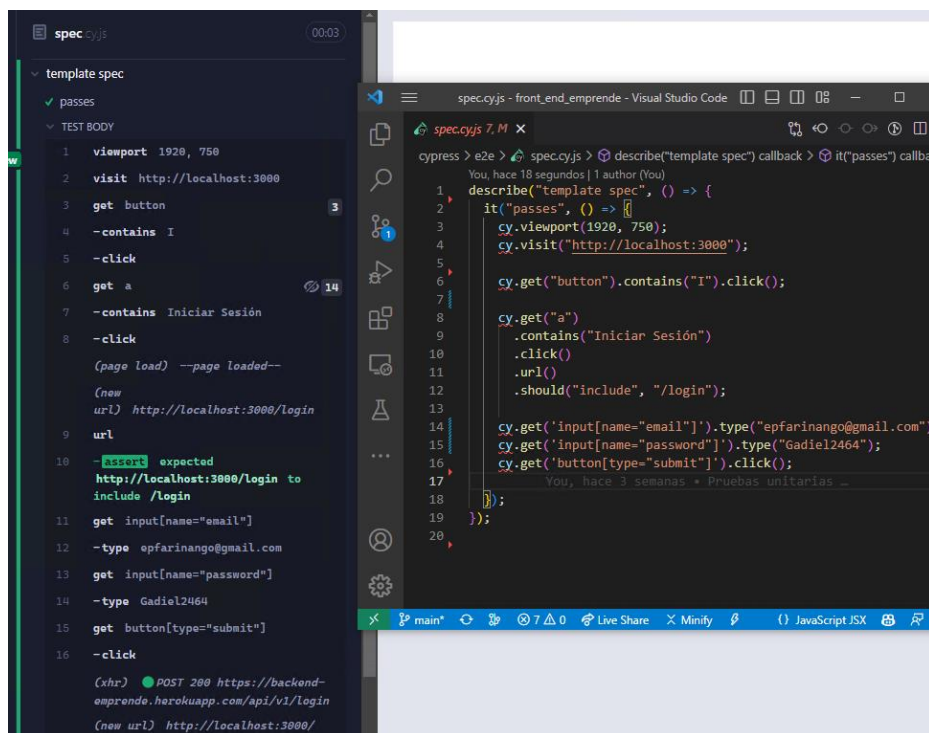


Figura 3.42 Prueba unitaria – Login exitoso

Este procedimiento se la realizó proporcionando un correo electrónico y una contraseña valida, *Cypress* nos muestra que la prueba fue exitosa y nuestro usuario inició sesión en el sistema rediriéndolo así a la ruta esperada, ahora se proporcionara los resultados de un ingreso de credenciales invalidas para poder validar su operatividad proporcionando un mensaje de error en el caso del correo electrónico invalido, en el **Anexo II** se podrá encontrar otras pruebas realizadas a los componentes principales

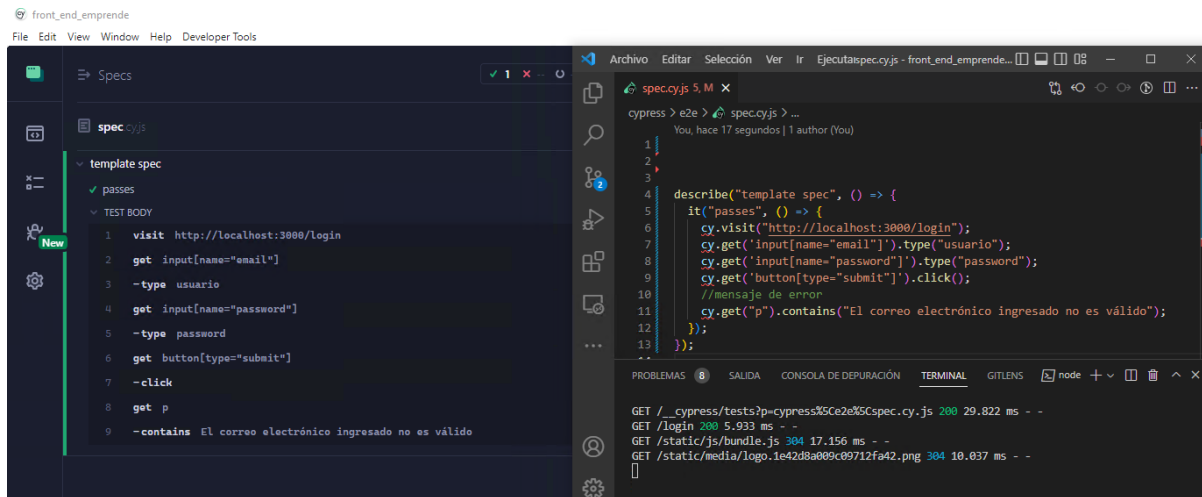


Figura 3.43 Prueba unitaria – Login fallido

3.7 Despliegue

Para que esta aplicación pueda ser utilizada se buscaron opciones que nos permitan publicarla en la *web*, *Vercel* es una plataforma basada en la nube de código abierto que mediante repositorios de *Github* se podrá publicar la aplicación como se observa en la **Figura 3.44**.

Para este propósito ingresaremos a nuestra cuenta y sincronizaremos el repositorio que deseemos publicar, este deberá estar actualizado y se debe tomar en cuenta que si se realiza algún cambio y es enviado a *Github* este automáticamente será desplegado en esta plataforma.

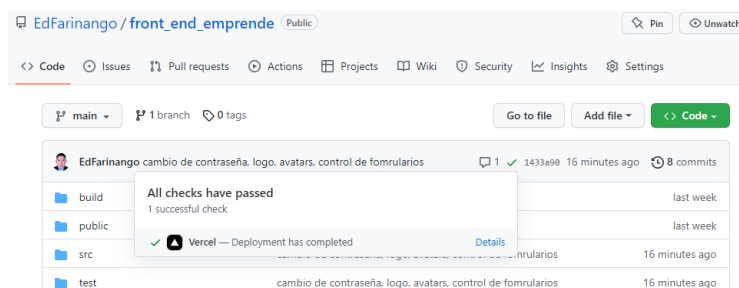
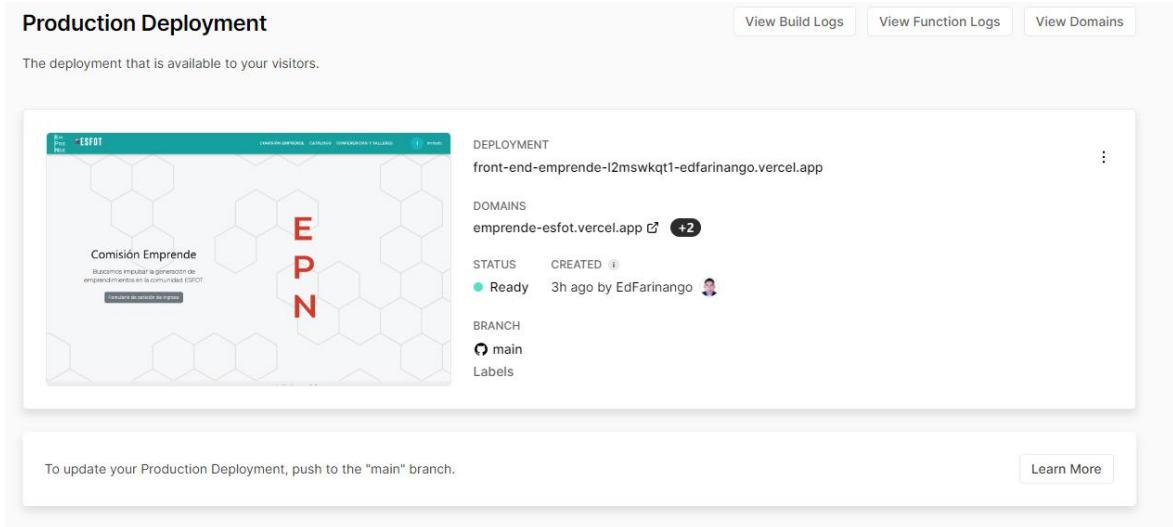


Figura 3.44 Sincronización Vercel – Github

El resultado de esta sincronización se la puede observar en la **Figura 3.45** Despliegue de aplicación Emprende. *Vercel* toma el repositorio y lo publica con un nombre referencial puesto por el usuario como dominio y con extensión “. *vercel*”.



The screenshot shows the Vercel Production Deployment interface. At the top, it says "Production Deployment" and "The deployment that is available to your visitors." There are three buttons: "View Build Logs", "View Function Logs", and "View Domains". Below this is a preview of the application, which is a landing page for "Comisión Emprende" with a hexagonal pattern background and the letters "EPN" in red. To the right of the preview, the deployment details are listed: "DEPLOYMENT" is "front-end-emprende-l2mswkqt1-edfarinango.vercel.app"; "DOMAINS" is "emprende-esfot.vercel.app" with a "+2" badge; "STATUS" is "Ready" (indicated by a green dot); "CREATED" is "3h ago by EdFarinango"; "BRANCH" is "main"; and "Labels" are listed below. At the bottom, there is a note: "To update your Production Deployment, push to the 'main' branch." and a "Learn More" button.

Figura 3.45 Despliegue de aplicación Emprende.

4 CONCLUSIONES

A continuación, se presentarán las conclusiones obtenidas dentro del marco de implementación del trabajo de integración curricular.

- Después de un detallado análisis de los requisitos funcionales y no funcionales, con el objetivo de cumplir con las expectativas de usuarios y administradores, se ha desarrollado una aplicación *web* que facilita la promoción de los emprendimientos de la comunidad de la ESFOT.
- El diseño de prototipos fue fundamental en la implementación de este componente, ya que ha permitido tener una ruta clara a seguir y considerar mejoras a lo largo del camino. Gracias a este proceso se pudieron probar diferentes estrategias para garantizar una implementación eficiente, recibir retroalimentación útil y un resultado final exitoso agradable a la vista del usuario.
- Para poder llegar a un producto final es necesario plantear un patrón arquitectónico que en este caso fue MVC, Modelo Vista-Controlador, teniendo en cuenta que React puede implementar esta arquitectura utilizando componentes como la vista. Con el uso de llamadas a una API en concreto se ha logrado enviar peticiones que tienen una respuesta válida consiguiendo así componentes capaces de tomar control sobre la información y presentarla de forma estructurada según cada petición del usuario.
- La metodología XP ha sido utilizada en este proyecto debido a su enfoque en iteraciones cortas que han producido resultados positivos, al considerar iteraciones regulares se pudo detectar y corregir errores de manera oportuna, evitando posibles retrasos en el cronograma de codificación y entrega.
- Se ha utilizado React como marco de desarrollo debido a su eficacia al momento de crear interfaces de usuario, React ofrece la capacidad de crear componentes con funcionalidades adaptables lo que permite la reutilización de su estructura, y, por ende, la mejora exponencial de la experiencia del usuario.
- Para poder comprobar el funcionamiento y evaluar el desempeño de la aplicación se realizaron pruebas de compatibilidad, aceptación y pruebas unitarias mismas que sirvieron para corregir errores en su estructura logrando así una aplicación modular y escalable de fácil mantenimiento.
- Se ha tomado la decisión de desplegar este componente en la plataforma de código abierto *Vercel* con el objetivo de facilitar el alcance de la información a una amplia gama de audiencia. *Vercel* es una aplicación gratuita que nos ha permitido utilizar

un dominio personalizado bajo su extensión, lo que ha resultado en una solución práctica y asequible para nuestro propósito.

5 RECOMENDACIONES

A continuación, se listan las recomendaciones obtenidas del proceso de desarrollo del trabajo de integración curricular.

- Para asegurar una implementación exitosa de una aplicación *web*, es recomendable comprender claramente la lógica del negocio a través de un levantamiento de requerimientos tanto funcionales como no funcionales de manera efectiva, estableciendo objetivos claros y específicos a fin de proporcionar una idea clara de los recursos necesarios que permitirán planificar el tiempo de desarrollo adecuado. Se sugiere además que se establezcan procedimientos claros y definidos para el seguimiento y evaluación de los avances del componente, lo que permitirá una implementación exitosa y oportuna de la aplicación.
- Para la implementación de una aplicación *web* de manera efectiva, es importante seleccionar una metodología de desarrollo adecuada que se adapte a los requisitos y objetivos del proyecto. Esta elección permitirá alinear los esfuerzos y recursos del equipo de desarrollo, facilitando el monitoreo constante del progreso y la identificación temprana de problemas. Se recomienda que la elección de la metodología se base en la evaluación de las necesidades del proyecto, las capacidades del equipo de desarrollo, el tipo de aplicación y los objetivos del cliente. Es recomendable también trabajar en estrecha colaboración con todo el equipo de desarrollo para asegurarse de que se está siguiendo los procedimientos y procesos acordados para asegurar el éxito del proyecto.
- Dentro del proceso de desarrollo es recomendable acudir a la documentación oficial para aclarar las dudas que puedan surgir en la codificación y reducir la probabilidad de cometer errores. La documentación oficial proporciona respuestas claras y en muchos casos más sencillas en comparación a las respuestas generadas por la comunidad de desarrolladores.
- Para asegurar una implementación fluida del proyecto, es fundamental definir las versiones de las herramientas a utilizar, con la constante actualización muchas van a dejar de funcionar o van a trabajar de manera diferente a una versión conocida. Es recomendable trabajar con las versiones más actualizadas de las herramientas de desarrollo para mejorar su rendimiento y garantizar su compatibilidad con otros programas, de esta forma se podrá asegurar una implementación eficiente y se reducirá el riesgo de problemas en el futuro.

- Se recomienda que la implementación de un sistema en el que Backend y el FrontEnd sean asignados a grupos diferentes, se fomente la comunicación y el trabajo en equipo, esto permitirá asegurar la integridad, eficiencia y calidad del producto final, ofreciendo una experiencia de usuario completa y satisfactoria. Se sugiere establecer canales de comunicación claros y regulares para garantizar una colaboración estrecha y efectiva entre los equipos de desarrollo.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] C. C. W. Viviana Elizabeth Jiménez Chaves, Los estudios de casos como enfoque metodológico, Asusnción: ACADEMO Revista de Investigación en Ciencias Sociales y Humanidades , 2016.
- [2] V. Morles, «Sobre la metodología como ciencia y el método científico: un espacio polémico,» *Revista de Pedagogía*, vol. 23, nº 66, 2002.
- [3] S. Universidades, «Metodologías de desarrollo de software,» Santander Becas, 21 12 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 06 12 2022].
- [4] M. M. R. C. M. Ailin Orjuela Duarte, «Redalyc.org,» 24 05 2008. [En línea]. [Último acceso: 08 01 2023].
- [5] A. Carranza, «CREHANA,» CREHANA, 26 11 2021. [En línea]. Available: <https://www.crehana.com/blog/transformacion-digital/aplicacion-web-que-es/>. [Último acceso: 18 11 2022].
- [6] N. Chapaval, «Frontend y Backend,» Platzi, 10 11 2017. [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>. [Último acceso: 06 12 2022].
- [7] A. C. S. Lévano, Investigación cualitativa: diseños, evaluación del rigor metodológico y retos, Lima: Liberabit, 2007.
- [8] D. Calvo, «Metodología XP Programación Extrema (Metodología ágil),» Diego Calvo, 17 04 2018. [En línea]. Available: <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/#:~:text=Roles,Tester%3A%20Encargado%20de%20Pruebas..> [Último acceso: 2022 12 20].
- [9] T. d. L. e. S. y. Computación, «Metodologías de desarrollo de software,» [En línea]. Available: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>. [Último acceso: 2022 12 20].

- [10] C. P. Patricio Letelier, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 2006. [En línea]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
- [11] A. M. L. Alliey, «DISEÑO DE LA INTERFAZ GRAFICA WEB,» [En línea]. Available: https://www.palermo.edu/dyc/maestria_diseno/pdf/tesis.completas/43.luzardo.pdf. [Último acceso: 2022 12 20].
- [12] K. Bracey, «Web Dasing,» Envatotuts+, 26 11 2018. [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>. [Último acceso: 20 11 2022].
- [13] S. d. Informática, «Modelo vista controlador (MVC),» Universidad de Alicante, [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 10 12 2022].
- [14] Mozilla.org, «MVC - MDN Web Docs Glossary: Definiciones de términos relacionados con la Web,» Mozilla.org , 20 09 2020. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. [Último acceso: 11 12 2022].
- [15] Reactjs.org, «Higher-Order Components – React,» [En línea]. Available: <https://reactjs.org/docs/higher-order-components.html>. [Último acceso: 11 12 2022].
- [16] Axios-http.com, «Getting Started,» [En línea]. Available: <https://axios-http.com/docs/intro>. [Último acceso: 11 12 2022].
- [17] M. U. SAS, «Muévete más rápido con las herramientas intuitivas de la interfaz de usuario de React,» [En línea]. Available: <https://mui.com/>. [Último acceso: 12 01 2023].
- [18] R. L. C. D. C. L. C. Yamilis Fernandez Prez*, «Redalyc.org.,» 08 2007. [En línea]. Available: <https://www.redalyc.org/pdf/3783/378343633008.pdf>. [Último acceso: 2023 01 25].
- [19] D. Sánchez-Hernández, «Integración de pruebas remotas de usabilidad en Programación Extrema: revisión de literatura,» *Uniciencia*, vol. 34, nº 1, p. 12, 2020.

7 ANEXOS

ANEXO I. Turnitin

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO IV. Manual de instalación



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

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 1 de marzo de 2023

De mi consideración:

Yo, Juan Pablo Zaldumbide Proaño, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE UN FRONTEND** asociado al **DESARROLLO DE APLICACIÓN WEB DE PROMOCIÓN DE EMPRENDIMIENTOS DE LA COMUNIDAD DE LA ESFOT** elaborado por el estudiante **EDUARDO PAÚL FARINANGO MALES** de la carrera en Tecnología Superior 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 9%.

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,

Juan Pablo Zaldumbide Proaño
Profesor ocasional a tiempo completo
ESFOT

ANEXO II

Manual técnico

En el siguiente anexo se describe los componentes de la metodología XP que se utilizaron para analizar los requerimientos de los usuarios y crear las respectivas historias de usuarios, la interfaz gráfica, iteraciones y las pruebas necesarias para comprobar el funcionamiento de la aplicación.

Historias de usuarios

En este apartado se presentan de forma secuencial las historias de usuarios, mismas que fueron parte fundamental para el desarrollo de la aplicación *web*.

TABLA XV: Historia de usuario 1 - Barra de navegación

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Super – Admin, Admin
Nombre de la Historia: Menú general (Barra de navegación)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Farinango Eduardo	
Descripción: Como administradores desean ver un menú en el que se tenga opciones: Comisión, catálogo de emprendimientos, repositorio de conferencias, panel de administración.	

TABLA XVI: Historia de usuario 2 – Información Comisión Emprende

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: General
Nombre de la Historia: Información Comisión Emprende	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Farinango Eduardo	
Descripción: Administradores y usuarios general desean visualizar información sobre la comisión Emprende, sus miembros, contactos y redes sociales.	

TABLA XVII: Historia de usuario 3 – Inicio de sesión

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Super – Admin, Admin
Nombre de la Historia: Inicio de sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
Responsable: Farinango Eduardo	
Descripción: Como administradores desean consumir un <i>endpoint</i> que le permita iniciar sesión, recuperar, y cambiar contraseña.	
Observaciones: En el caso de que el usuario cambie o recupere su contraseña deberá tomar en cuenta que esta debe cumplir con características específicas de la aplicación. Para el caso del cambio de contraseña el usuario recibirá un mensaje de cierre de sesión.	

TABLA XVIII: Historia de usuario 4 – CRUD administradores

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Super - Admin
Nombre de la Historia: CRUD usuarios Super administradores.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
Responsable: Eduardo Farinango	
Descripción: Como Super-Admin se desea consumir un <i>endpoint</i> que mediante un formulario permita crear, actualizar o eliminar un usuario administrador.	
Observación: Al momento de crear un usuario este recibirá un mensaje en su dirección de correo electrónico registrado, el cual contendrá una contraseña provisional y un enlace que lo redirigirá al inicio de sesión de la aplicación. Este nuevo usuario podrá cambiar la contraseña dentro del panel de administración dependiendo su rol.	

TABLA XIX: Historia de usuario 5 – CRUD administradores generales

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Super - Admin
Nombre de la Historia: CRUD usuarios administradores generales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 2	
Responsable: Eduardo Farinango	
Descripción: Como Super-Admin se desea consumir un <i>endpoint</i> que mediante un formulario permita crear, actualizar o eliminar un usuario administrador.	
Observaciones:	
Al momento de crear un usuario este recibirá un mensaje en su dirección de correo electrónico registrado, el cual contendrá una contraseña provisional y un enlace que lo redirigirá al inicio de sesión de la aplicación. Este nuevo usuario podrá cambiar la contraseña dentro de su perfil.	

TABLA XX: Historia de usuario 6 – Módulo emprendimientos

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Super - Admin
Nombre de la Historia: Administración de emprendimientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 2	
Responsable: Eduardo Farinango	
Descripción: Como administrador se desea consumir un <i>endpoint</i> que permita autorizar, crear, visualizar, eliminar información y actualizar el catálogo de emprendimiento que se presenta a la comunidad.	

TABLA XXI: Historia de usuario 7 - Validación de campos en formularios

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Administradores
Nombre de la Historia: Validación de campos en Formularios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 2	
Responsable: Eduardo Farinango	
Descripción: Como administradores se desea validar los campos de los formularios antes de enviar al <i>endpoint</i> .	
Observación: Se presentarán alertas en cada campo que no cumpla con el formato requerido.	

TABLA XXII: Historia de usuario 8 – Visualización de emprendimientos

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Usuario general
Nombre de la Historia: Visualización de emprendimientos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 2	
Responsable: Eduardo Farinango	
Descripción: Como usuarios se desea ver una lista de emprendimientos con información relevante	

TABLA XXIII: Historia de usuario 9 - formularios

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Admin
Nombre de la Historia: Formularios de solicitud.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 3	
Responsable: Eduardo Farinango	
Descripción:	

Como administradores se desea consumir un *endpoint* que permita recibir solicitudes para poder formar parte del catálogo de la comisión Emprende.

Observación:

Los administradores encargados podrán aprobar las solicitudes enviándolas al panel principal teniendo la posibilidad de activar o desactivar su visualización en el catálogo principal.

TABLA XXIV: Historia de usuario 10 - Filtros

HISTORIA DE USUARIO	
Identificador (ID): HU010	Usuario: Administradores
Nombre de la Historia: Filtros en catálogo.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 4	
Responsable: Farinango Eduardo	
Descripción:	
Como administradores se desea consumir un <i>endpoint</i> que permita filtrar el catálogo de emprendimientos por categoría.	

TABLA XXV: Historia de usuario 11 – Módulo videoconferencias

HISTORIA DE USUARIO	
Identificador (ID): HU011	Usuario: Administradores
Nombre de la Historia: Videoconferencias comisión Emprende.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 4	
Responsable: Eduardo Farinango	
Descripción:	
Como administradores se desea consumir un <i>endpoint</i> que despliegue los videos de conferencias dictadas por la comisión Emprende.	

Prototipado

En este apartado se lista los prototipos utilizados en el desarrollo de la aplicación *web* para la Comisión Emprende siguiendo el flujo que un usuario administrador, se iniciara con la configuración general de los colores y tipografías.

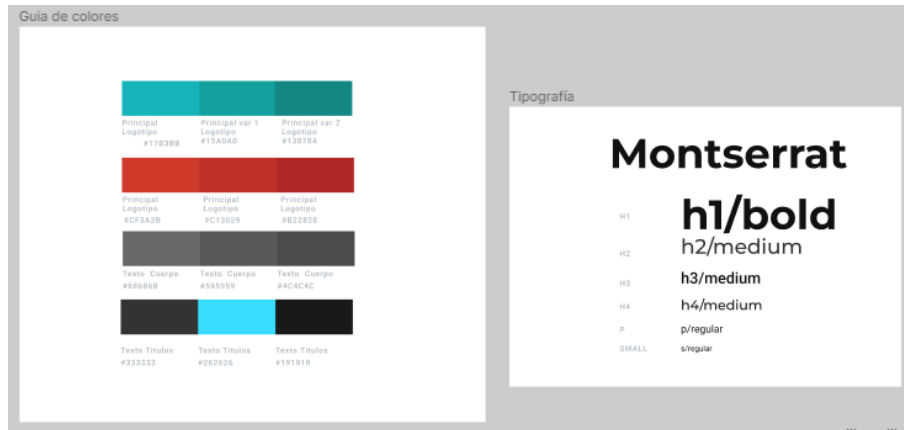


Figura 7.1. Paleta de colores y tipografía de la comisión Emprende



Figura 7.2 Variación del logo de la Comisión Emprende



Figura 7.3 Página de inicio de la aplicación *web*



Figura 7.4 Página de inicio de sesión



Figura 7.5 Pie de página aplicación web.



Figura 7.6 Comisión EmPrende

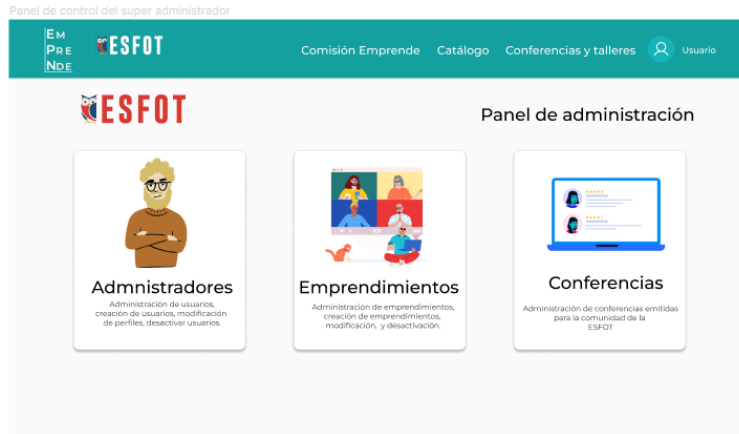


Figura 7.7 Panel de superadministradores

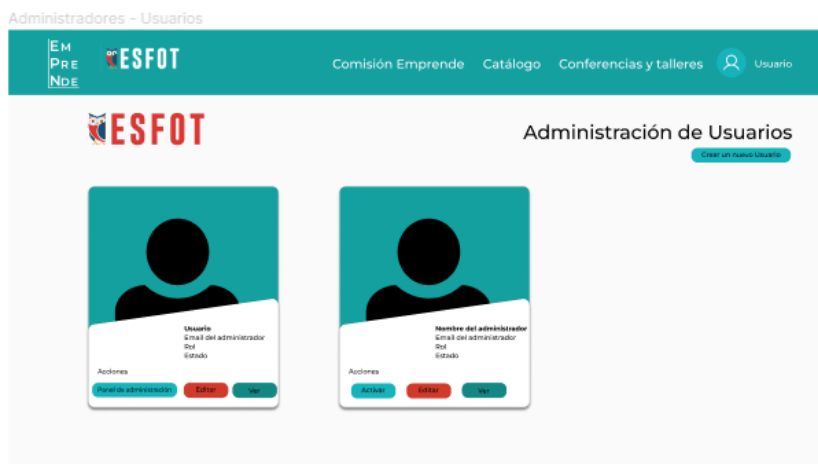


Figura 7.8 Vista superadministradores

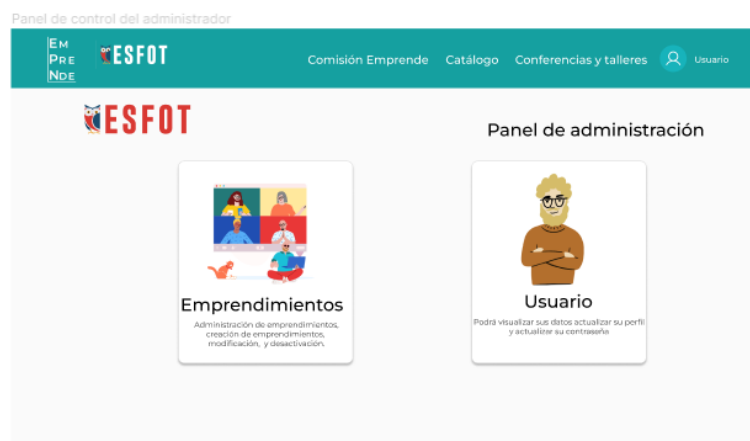


Figura 7.9 Vista administradores

Crear usuario

Crear X

Formulario de Registro

Nombre
Ingrese el nombre

Apellido
Ingrese el apellido

Correo Electrónico
Email

Teléfono personal
0999999999

Linkedin
Linkedin

Crear Limpiar

Figura 7.10 Formulario de creación y edición de usuario

Formulario de Emprendimiento

Crear Emprendimiento X

Rol Esfot
Administrativo ▼

Categoría
Agroindustria ▼

Página Web
no posee

Facebook
https://www.facebook.com/ysaca

Nombre
Carnes la Familia

Dirección
San Antonio de Pichincha

Teléfono
0968186418

Instagram
https://instagram.nohay.co

Descripción
Carricería familiar con variedad

Cobertura
Cumbaya-Tumbaco ▼

Whatsapp
0929945400

Descuento
10

Seleccionar archivo

Guardar

Figura 7.11 Formulario de creación de emprendimientos

Adición de emprendimientos

Editar Emprendimiento X

Rol Esfot
▼

Categoría
▼

Página Web
Página Web

Facebook
Facebook

Nombre
Nombre

Dirección
Dirección

Teléfono
Teléfono

Instagram
Instagram

Descripción
Facebook

Cobertura
▼

Whatsapp
Whatsapp

Descuento
Descuento

Seleccionar archivo Sin archivos seleccionados

Actualizar Imagen

Cerrar

Figura 7.12 Formulario de edición de emprendimiento

Vista de emprendimientos

 **Nombre del Emprendimiento**

Descripción

Descripción del emprendimiento

Dirección

Ubicación del emprendimiento

Teléfono

022222222

Sitio Web

www.emprendimiento.com

Facebook

facebook.emprendimiento.com

Instagram

instagram.emprendimiento.com

Whatsapp

0999999999

Categoría

Categoría del emprendimiento

Cobertura

Lugar donde distribuyen el producto

Inactivo

Figura 7.13 Vista emprendimientos

Emprendimientos

ESFOT

Comisión Empresa Catálogo Conferencias y talleres

ESFOT

Administración de Emprendimientos







Emprendimientos				
Categoría	Nombre	Descripción	Imagen	Acciones
Categoría 1	Nombre 1	Descripción 1	Imagen 1	 
Categoría 2	Nombre 2	Descripción 2	Imagen 2	 
Categoría 3	Nombre 3	Descripción 3	Imagen 3	 

Figura 7.14 Administración de emprendimientos

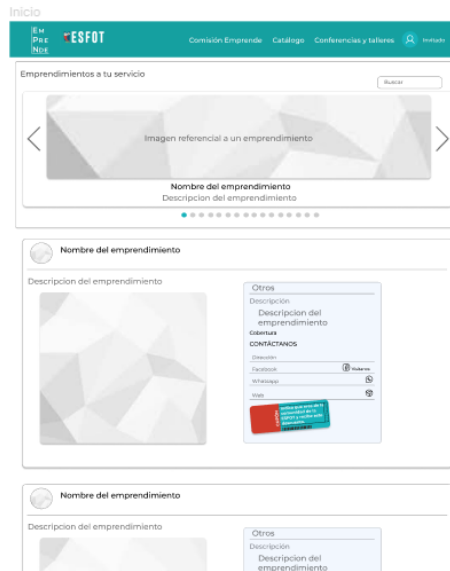


Figura 7.15 Vista de catálogo



Figura 7.16 Vista conferencias administrador



Figura 7.17 Vista conferencias usuario



Figura 7.18 Alertas



Figura 7.19 Cupón de descuento

En el siguiente enlace se podrá encontrar los diseños base creados en Figma.

- <https://www.figma.com/file/T1BNwNYlaWZ3uJpJaVWh33/Untitled?node-id=0%3A1&t=mOAaVqvKCA47DXJ0-1>

Herramientas

Para poder realizar configuración del ambiente de desarrollo iniciaremos con la descarga de Visual Studio Code del siguiente enlace: <https://code.visualstudio.com/download>

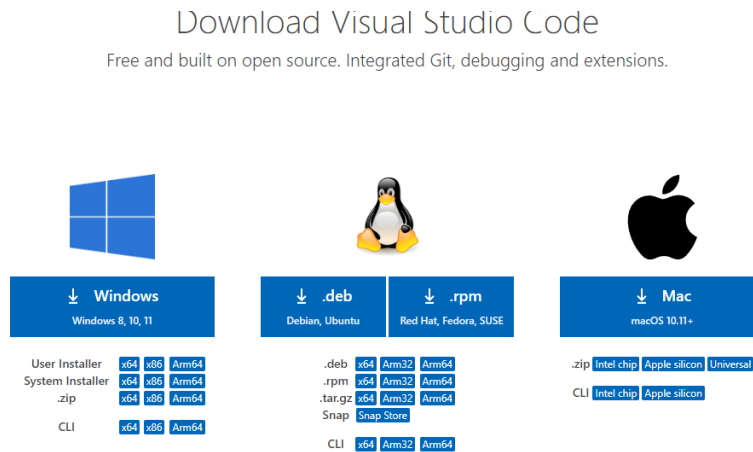


Figura 7.20 Instalación de VScode.

Una vez instalado esta herramienta procederemos a la descarga de Node.js del siguiente enlace: <https://nodejs.org/es/download/>



Figura 7.21 Descarga de Node.js

Ahora que ya tenemos instaladas nuestras herramientas principales de nuestro proyecto procederemos a crearlo utilizando React como se observa en la **Figura 7.22**

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ed\Desktop\Front-end\React-Fundamentos> code .

C:\Users\ed\Desktop\Front-end\React-Fundamentos>cd ..

C:\Users\ed\Desktop\Front-end>npx create-react-app emprende

Creating a new React app in C:\Users\ed\Desktop\Front-end\emprende.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1394 packages in 1m

213 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 72 packages in 10s

226 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

```

Figura 7.22 Creación de proyecto React emprende

Principales librerías utilizadas.

TABLA XXVI. Librerías

Librería	Comando de instalación	Descripción
AXIOS	<i>npm install axios</i>	Cliente HTTP basado en promesas
Material UI	<i>npm install @mui/material @emotion/react @emotion/styled</i>	Biblioteca de componentes precargados
React router dom	<i>npm add react-router-dom</i>	Navegación
sweetalert	<i>npm install sweetalert2</i>	Alertas precargadas
React Bootstrap	<i>npm install react-bootstrap bootstrap</i>	Componentes <i>bootstrap</i> para React
Cypress	<i>npm install cypress --save-dev</i>	Pruebas 2e2 y unitarias

Validaciones

Para poder saber si algo falla en nuestro sistema o si nuestro proceso fue exitoso se desplegarán las siguientes alertas y validaciones dentro de los formularios que serán un complemento a las validaciones creadas por los *endpoints* que se consumen.

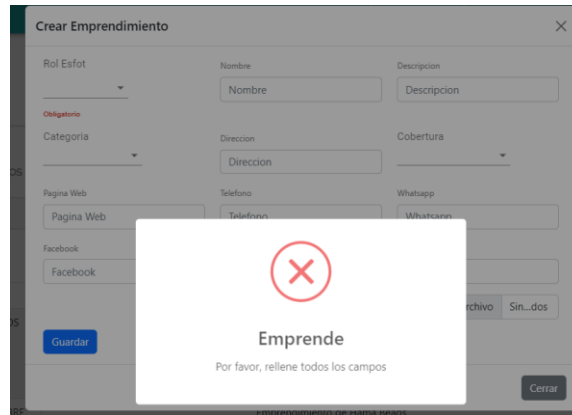


Figura 7.23 Validación de campos en formulario de creación de emprendimiento



Figura 7.24 Alerta de imagen que excede el peso admitido por el servidor

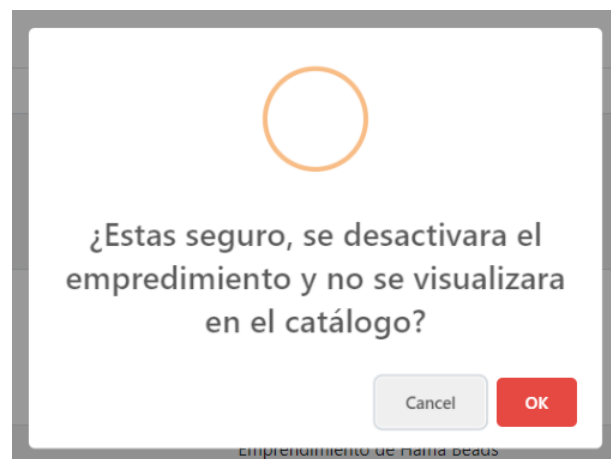


Figura 7.25 Espera de confirmación para enviar la petición

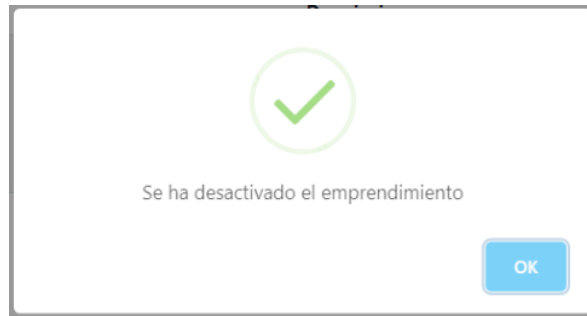


Figura 7.26 Confirmación de desactivación de emprendimientos



Figura 7.27 Alerta al momento de aprobación de emprendimientos

A registration form titled "Formulario de Registro" inside a window titled "Crear". The form has five input fields, each with a red border and a red error message below it. The fields are: "Nombre" (error: "Debe ingresar un nombre válido!"), "Apellido" (error: "Debe ingresar un apellido válido!"), "Correo Electrónico" (error: "Email no válido!"), "Teléfono personal" (error: "Solamente se aceptan números de hasta 10 dígitos"), and "LinkedIn". At the bottom of the form are two blue buttons: "Crear" and "Limpiar".

Figura 7.28 Validación de formulario de creación de usuarios

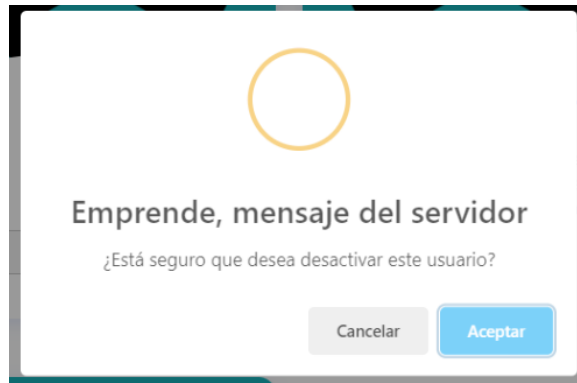


Figura 7.29 Mensaje de advertencia de desactivación de un usuario

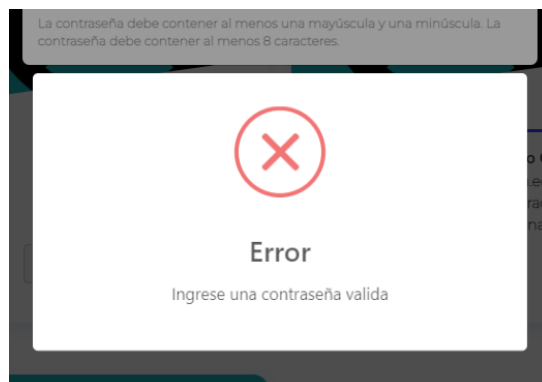


Figura 7.30 Ingreso de contraseñas invalidas en cambio de contraseña

Pruebas de aceptación

Se presentan el complemento de las pruebas de aceptación.

TABLA XXVII Prueba de aceptación 1

Pruebas de aceptación	
Numero: 1	HU000
Nombre: Página de inicio	
Descripción: Como usuario desea ver una página de inicio con la presentación de la comisión Emprende, en su contenido debe tener el logo de la marca Emprende y sus colores.	
Parámetros de ejecución: Se presentará una pantalla de inicio en la que se deberá observar el logo de la comisión su visión general debe estar acorde a los colores de la marca Emprende.	
Entrega: La pantalla de inicio cuenta con una barra de navegación un logo y colores establecidos por la marca Emprende	
Resultado esperado: El usuario al momento de ingresar a la URL establecida para la aplicación podrá observar una pantalla de inicio acorde a los requerimientos de la comisión Emprende.	
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda aumentar contenido visual (Fondos de tras del logo)	

TABLA XXVIII: Prueba de aceptación 2

Pruebas de aceptación	
Numero: 2	HU001
Nombre: Menú general (Barra de navegación)	
Descripción: Como administradores desean ver un menú en el que se tenga opciones: Comisión, catálogo de emprendimientos, repositorio de conferencias, panel de administración.	
Parámetros de ejecución: Se presentará una pantalla de inicio en la que se podrá observar una barra de navegación con los accesos a los módulos.	
Entrega:	

La pantalla de inicio cuenta con una barra de navegación que contiene el acceso directo a los diferentes módulos de la aplicación.
Resultado esperado: El usuario cuando presione en los diferentes componentes de la barra de navegación se generará una acción que lo llevara a la página solicitada.
Evaluación de la prueba: Prueba satisfactoria.

TABLA XXIX: Prueba de aceptación 3

Pruebas de aceptación	
Numero: 3	HU002
Nombre: Información Comisión Emprende	
Descripción: Administradores y usuarios general desean visualizar información sobre la comisión Emprende, sus miembros, contactos y redes sociales.	
Parámetros de ejecución: Se presentará un módulo en el que se deberá observar información relevante de la comisión emprende, así también se podrá conocer los miembros que conforman esta comisión junto al equipo colaborador.	
Entrega: En el módulo de Información de la comisión Emprende cuenta con una reseña de su creación, su objetivo principal y los miembros de la comisión.	
Resultado esperado: Cuando se ingrese al módulo de información el usuario podrá observar información relevante relacionada con la comisión Emprende	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA XXX: Prueba de aceptación 4

Pruebas de aceptación	
Numero: 4	HU003
Nombre: Inicio de sesión	
Descripción: Como administradores desean consumir un <i>endpoint</i> que le permita iniciar sesión, recuperar, y cambiar contraseña.	
Parámetros de ejecución:	

Los miembros de la comisión emprenden deberán ingresar al módulo de inicio de sesión e ingresarán el correo electrónico y una contraseña.
Entrega: El formulario de inicio de sesión cuenta con campos que aceptan un email y una contraseña acompañado en un botón para enviar este formulario e iniciar sesión.
Resultado esperado: Si el usuario ingresa con las credenciales proporcionadas se mostrará el panel de administración con los módulos a los que tenga acceso según el rol.
Evaluación de la prueba: Prueba satisfactoria.

TABLA XXXI: Prueba de aceptación 5

Pruebas de aceptación	
Numero: 5	HU004
Nombre: CRUD usuarios Super administradores	
Descripción: Como Super-Admin se desea consumir un <i>endpoint</i> que mediante un formulario permita crear, actualizar o eliminar un usuario administrador.	
Parámetros de ejecución: El Superadministrador con un usuario activo podrá visualizar los administradores registrados y podrá crear, ver, eliminar y actualizar la información.	
Entrega: El módulo de administracion se presenta con los usuarios administradores registrados y el respectivo acceso para realizar el CRUD.	
Resultado esperado: Si el administrador desea crear, ver, eliminar y actualizar la información deberá acceder a cada una de las acciones.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXII: Prueba de aceptación 6

Pruebas de aceptación	
Numero: 6	HU005
Nombre: CRUD usuarios administradores generales	
Descripción: Como Super-Admin se desea consumir un <i>endpoint</i> que mediante un formulario permita crear, actualizar o eliminar un usuario administrador.	
Parámetros de ejecución: El Superadministrador con un usuario activo podrá visualizar los administradores registrados y podrá crear, ver, eliminar y actualizar la información.	
Entrega: El módulo de administracion se presenta con los usuarios administradores registrados y el respectivo acceso para realizar el CRUD.	
Resultado esperado: Si el administrador desea crear, ver, eliminar y actualizar la información deberá acceder a cada una de las acciones.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXIII: Prueba de aceptación 7

Pruebas de aceptación	
Numero: 7	HU006
Nombre: Administracion de emprendimientos	
Descripción: Como administrador se desea consumir un <i>endpoint</i> que permita autorizar, crear, visualizar, eliminar información y actualizar el catálogo de emprendimiento que se presenta a la comunidad.	
Parámetros de ejecución: Los administradores generales con un usuario activo podrán autorizar, crear, editar, activar y desactivar los emprendimientos registrados.	
Entrega: El módulo de emprendimiento se presenta con la lista de emprendimientos registrados y el respectivo acceso para realizar el CRUD.	
Resultado esperado: Si el administrador desea crear, ver, eliminar y actualizar la información deberá tener una sesión activa para acceder a cada una de las acciones.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXIV: Prueba de aceptación 8

Pruebas de aceptación	
Numero: 8	HU007
Nombre: Validación de campos en Formularios.	
Descripción: Como administradores se desea validar los campos de los formularios antes de enviar al <i>endpoint</i>	
Parámetros de ejecución: Los administradores generales desean validar los campos utilizando alertas con mensajes informativos y así evitar el ingreso de código malicioso.	
Entrega: Los formularios presentaran un mensaje en pantalla cuando los datos ingresados en el campo no sean válidos.	
Resultado esperado: Si el administrador desea validar, los formularios presentan un mensaje cuando no son válidos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXV: Prueba de aceptación 9

Pruebas de aceptación	
Numero: 9	HU008
Nombre: Visualización de emprendimientos.	
Descripción: Como usuarios se desea ver una lista de emprendimientos con información relevante.	
Parámetros de ejecución: El usuario en general podrá visualizar la información de los emprendimientos autorizados y activos.	
Entrega: En el módulo catálogo de emprendimientos se presenta la información de los emprendimientos autorizados.	
Resultado esperado: El usuario puede ver la información del catálogo de la comisión Emprende	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXVI: Prueba de aceptación10

Pruebas de aceptación	
Numero: 10	HU009
Nombre: Formularios de solicitud	
Descripción: Como administradores se desea consumir un <i>endpoint</i> que permita recibir solicitudes para poder formar parte del catálogo de la comisión Emprende.	
Parámetros de ejecución: El Administrador podrá recibir solicitudes para formar parte del catálogo emprende por medio de un formulario.	
Entrega: En la página de inicio se presentará un botón que desplegará un formulario con los campos requeridos por la comisión Emprende.	
Resultado esperado: El Administrador recibirá la solicitud del emprendimiento y será capaz de administrarlo	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXVII: Prueba de aceptación 11

Pruebas de aceptación	
Numero: 11	HU010
Nombre: Filtros en catálogo.	
Descripción: Como administradores se desea consumir un <i>endpoint</i> que permita filtrar el catálogo de emprendimientos por categoría.	
Parámetros de ejecución: El Administrador podrá ingresar texto de un campo que le permitirá realizar una búsqueda.	
Entrega: se presenta un campo que le permite realizar una búsqueda	
Resultado esperado: Se podrá realizar búsquedas dentro de la página del catálogo y la administración de los emprendimientos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla XXXVIII: Pruebas de aceptación 12

Pruebas de aceptación	
Numero: 12	HU011
Nombre: Videoconferencias comisión Emprende	
Descripción: Como administradores se desea consumir un <i>endpoint</i> que despliegue los videos de conferencias dictadas por la comisión Emprende.	
Parámetros de ejecución: El usuario podrá visualizar las videoconferencias organizadas por la comisión Emprende	
Entrega: El usuario podrá ver la grabación de los videos realizados por la comisión Emprende	
Resultado esperado: El usuario podrá ingresar al repositorio de videos de las conferencias realizadas por la comisión Emprende	
Evaluación de la prueba: Prueba satisfactoria.	

Pruebas unitarias

Se procede a utilizar *Cypress* que nos permite escribir código de tal manera que se pueda comprobar el renderizado y funcionalidades que estas tengan.

Prueba unitaria de la barra de navegación.

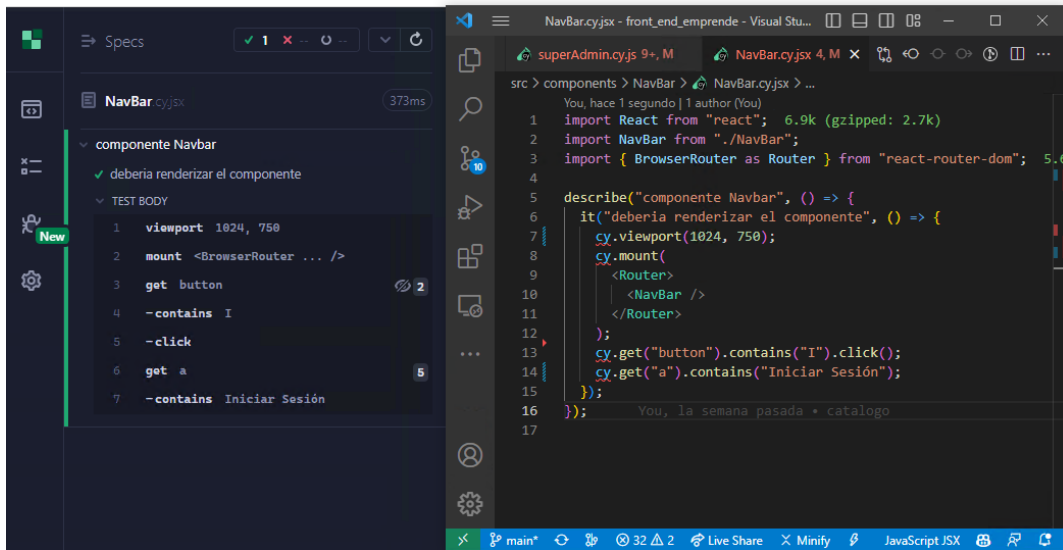


Figura 7.31 Prueba unitaria de la barra de navegación – botón inicio de sesión

Iniciamos estableciendo un tamaño de la pantalla adecuado para la prueba, establecemos que el componente de la barra de navegación realice el montaje dentro del componente enrutador, una vez que tenemos nuestro componente renderizado procedemos a probar si el botón de inicio por medio de una función de Cypress despliega la etiqueta Inicio de sesión, teniendo como resultado una prueba exitosa.

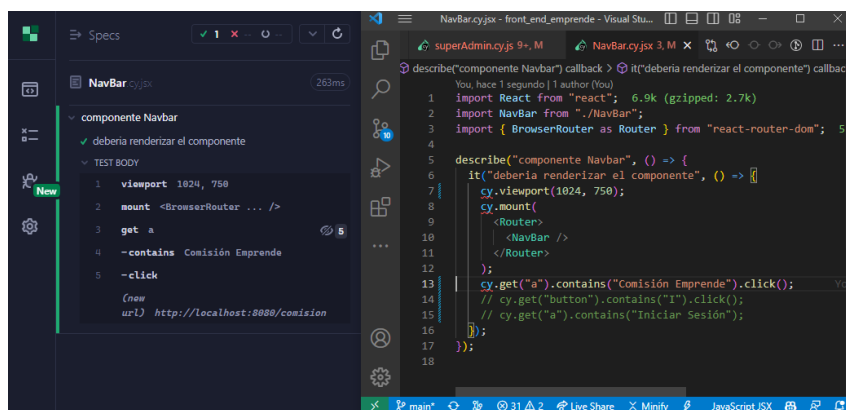


Figura 7.32 Prueba unitaria de la barra de navegación – Comisión Emprende

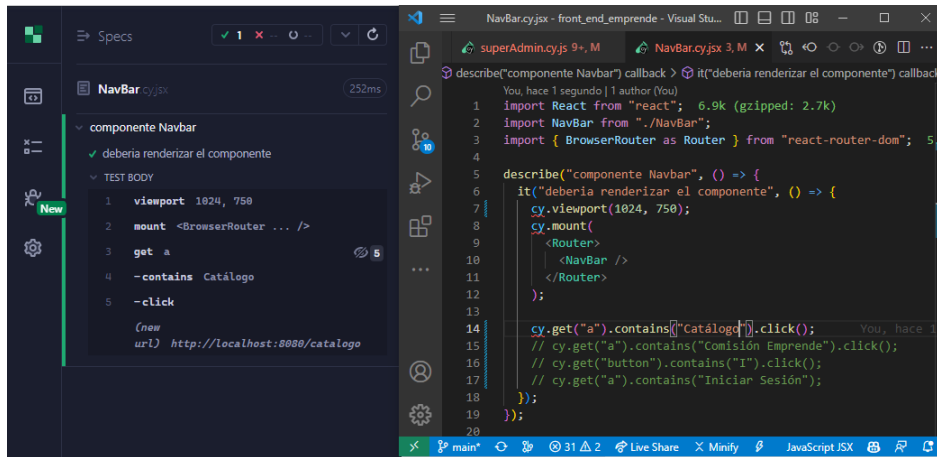


Figura 7.33 Prueba unitaria de la barra de navegación – Catálogo

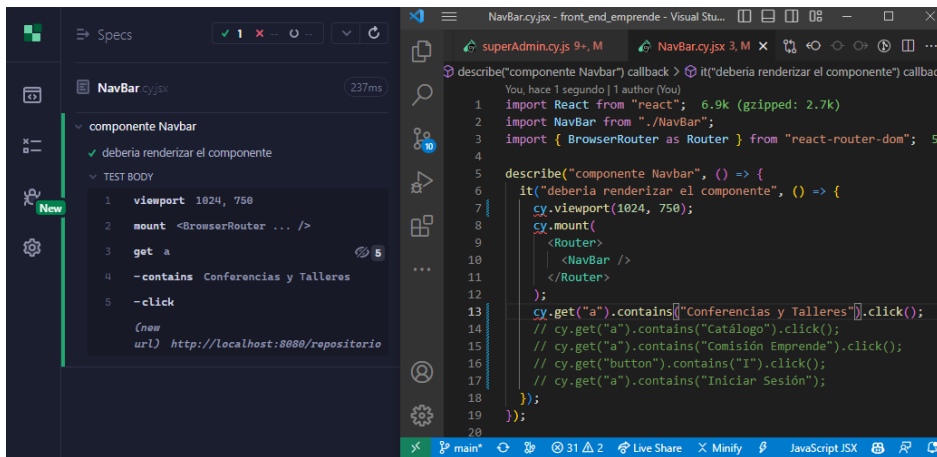


Figura 7.34 Prueba unitaria de la barra de navegación – Conferencias y Talleres

Como podemos observar en las **Figura 7.32**, **Figura 7.33** y **Figura 7.34** utilizamos el mismo método para poder obtener el acceso a los diferentes apartados de la aplicación obteniendo como test resultado las rutas establecidas para cada sección como lo son /comisión, /catalogo y /repositorio estableciendo esta prueba como exitosa.

Prueba unitaria de formulario de creación de usuarios.

Para esta prueba tomaremos en cuenta que no tenemos una sesión iniciada es por eso por lo que solo comprobaremos si nuestro componente se renderiza correctamente y que su formulario se presente y acepte datos.

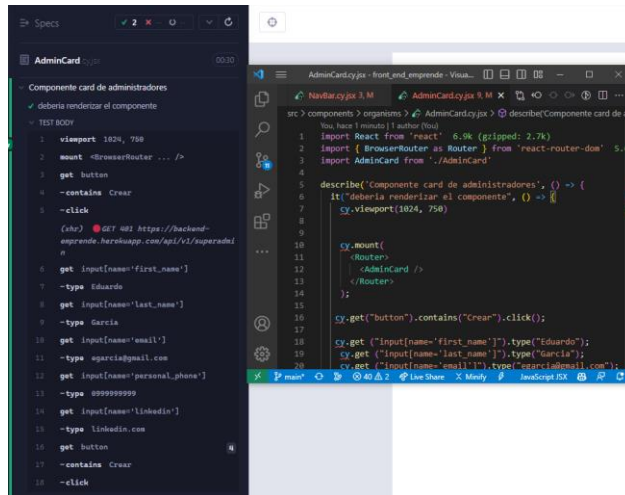


Figura 7.35 Prueba unitaria – Componente creación de usuarios

Prueba unitaria formularios de recuperación de contraseña

Para esta prueba se trata de verificar si el componente *ModalPswd* se pueda representar y que el campo de entrada de la contraseña se pueda escribir y tenga el valor esperado determinando una prueba exitosa.

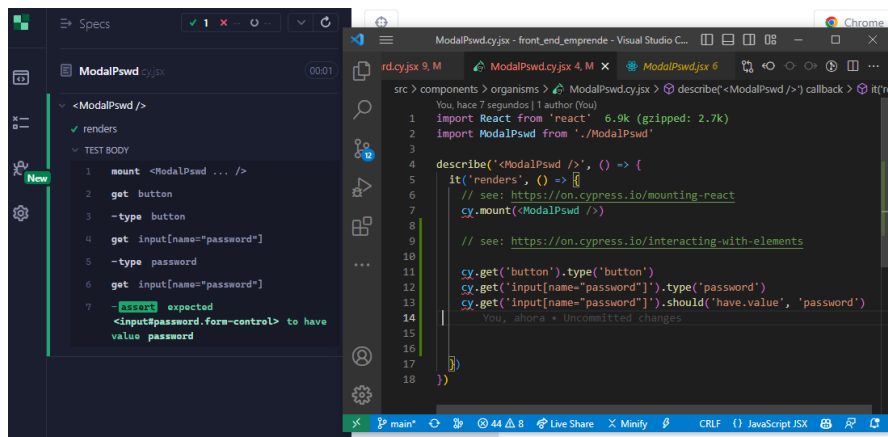


Figura 7.36 Prueba unitaria – Cambio de contraseña

Para la siguiente prueba se trata de comprobar si existe un mensaje de información sobre cómo debe estar estructurada la contraseña misma que si encuentra el mensaje se procederá a ingresar los datos y a enviar el formulario, como se observa en la **Figura 7.37** considerando que fue una prueba exitosa.

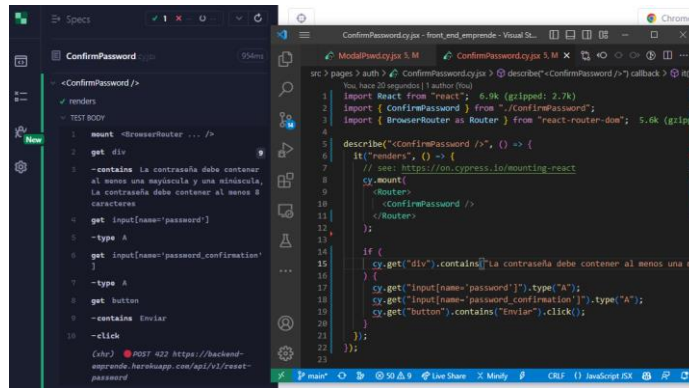


Figura 7.37– Recuperación de contraseña

Prueba unitaria de petición de nuevo emprendimiento

En esta prueba se comprobó el renderizado del formulario de petición de un nuevo emprendimiento mismo que se considera exitoso debido a que se comporta de la manera esperada, arbitrariamente no se envía la imagen de referencia recibiendo como resultado el mensaje de error que se configuro en el caso de que un campo no este completo o vacío como se observa en la **Figura 7.39**

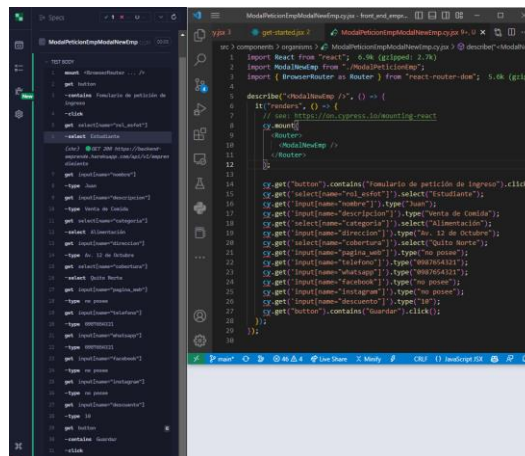


Figura 7.38 Prueba unitaria – Nueva petición emprendimiento

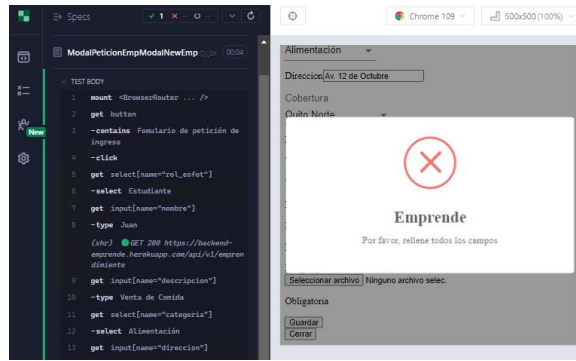


Figura 7.39 Prueba unitaria – Nueva petición emprendimiento error de campo

Prueba unitaria repositorio de videos

En esta prueba solamente se pide el renderizado del componente ‘Repositorio’ mismo que contiene los videos cargados en la base de datos desde el *endpoint* y que con una petición al mismo nos devuelve los enlaces para poder presentarlos en pantalla, esta respuesta del componente nos deja a consideración declarar una prueba exitosa.

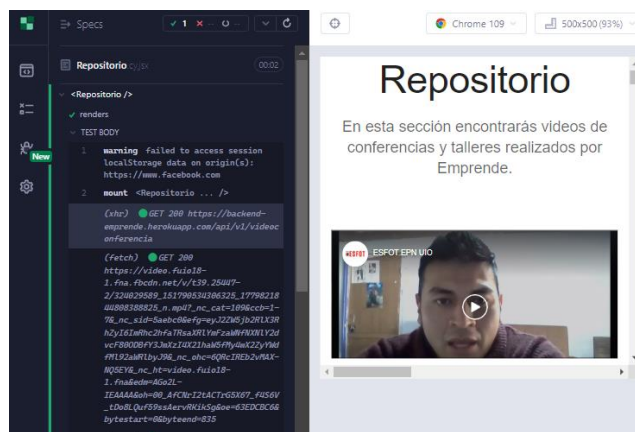


Figura 7.40 Prueba unitaria – Repositorio de videos Emprende

Prueba unitaria catálogo de emprendimientos

Para esta prueba se busca observar los emprendimientos registrados mediante el renderizado del componente ‘Catalogo’ mismo que durante la prueba no presento fallas ni retrasos.

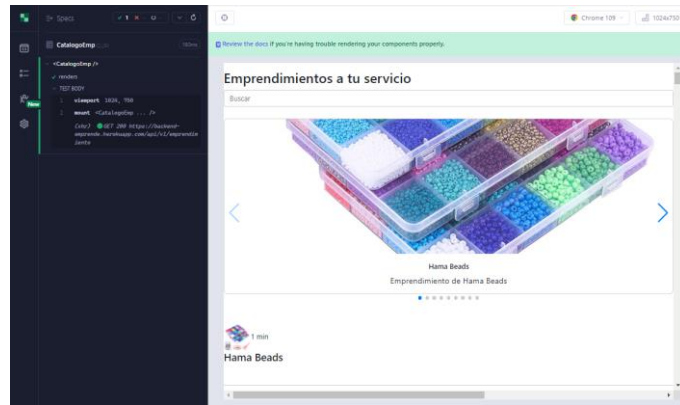


Figura 7.41 Prueba unitaria – Catalogo de emprendimientos

ANEXO III

Manual de usuario

En ese apartado se encontrará un enlace a un video que buscará explicar de la manera más sencilla los puntos más relevantes de este trabajo de integración curricular. Se cubrirán temas como: Inicio de sesión, creación de usuarios, activación y desactivación de usuarios, creación de emprendimientos, activación y desactivación de los emprendimientos, se explicará las actividades a las que tendrán acceso según su rol.

Para acceder al video explicativo, se proporcionará un enlace que dirigirá al usuario a la plataforma de YouTube donde podrá visualizar el contenido.

- <https://youtu.be/dtlhmq7dBfc>

ANEXO IV

Manual de instalación

Este apartado tratara de explicar el proceso que debe cumplir cualquier otra persona que desee utilizar este proyecto, esto incluye su instalación y credenciales para su uso.

Actualmente, la aplicación se puede acceder a través del siguiente enlace:

<https://emprende-esfot.vercel.app/>

Credenciales

En caso de necesitar ayuda, por favor comuníquese con el administrador del sistema a través del correo electrónico eduardo.farinango@epn.edu.ec para recibir asistencia.

Credenciales Perfil administrador

Usuario: emprende.esfot@gmail.com

Contraseña: Emprende2022B

El enlace al código fuente de la página que se encuentra alojado en *GitHub* es el siguiente

- https://github.com/EdFarinango/front_end_emprende.git