

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UN SISTEMA WEB Y APP DE E-COMMERCE
PARA LA TIENDA SATURNINA**

DESARROLLO DE UN FRONTEND

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

MIGUEL ANGEL CARAPAZ MENDOZA

DIRECTOR: BYRON GUSTAVO LOARTE CAJAMARCA

DMQ, marzo 2024

CERTIFICACIONES

Yo, **MIGUEL ANGEL CARAPAZ MENDOZA** 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.

Miguel Angel Carapaz Mendoza

miguel.carapaz@epn.edu.ec

macm.1726527755@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **MIGUEL ANGEL CARAPAZ MENDOZA**, bajo mi supervisión.

Ing. BYRON LOARTE, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

MIGUEL ANGEL CARAPAZ MENDOZA

DEDICATORIA

A mis amados padres, cuyo inquebrantable apoyo y amor han sido mi fuente constante de inspiración. Su sacrificio y aliento han iluminado mi camino hacia este logro.

A mis valiosos compañeros de tesis, quienes compartieron conmigo desafíos y triunfos, su colaboración y amistad han enriquecido esta experiencia. Este trabajo es también un tributo a la unidad y fortaleza que hemos construido juntos. A todos ustedes, gracias por ser parte fundamental de este capítulo significativo en mi vida.

MIGUEL ANGEL CARAPAZ MENDOZA

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mi estimado tutor y director de tesis. Su orientación experta, dedicación incansable y sabios consejos fueron cruciales en cada etapa de este arduo proceso.

MIGUEL ANGEL CARAPAZ MENDOZA

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.....	4
2 METODOLOGÍA.....	7
2.1 Metodología de Desarrollo	7
Roles.....	8
Artefactos	9
Recopilación de Requerimientos.....	9
2.2 Diseño de interfaces.....	12
Herramienta utilizada para el diseño	12
2.3 Diseño de la arquitectura.....	13
Patrón arquitectónico	13
2.4 Herramientas de desarrollo	14
3 RESULTADOS	17
Sprint 0. Establecimiento de la configuración para el entorno de desarrollo...17	
Sprint 1. Diseño de interfaces destinadas al usuario con perfil - administrador	22
Sprint 2. Diseño de interacciones destinadas al perfil cliente	31
Sprint 3. Evaluaciones dirigidas al desarrollo del frontend	35
Sprint 4. Ejecución del componente frontend en un entorno de producción ...39	
4 CONCLUSIONES	41
5 RECOMENDACIONES	42
6 REFERENCIAS BIBLIOGRÁFICAS	43

7 ANEXOS.....	47
ANEXO I.....	48
ANEXO II.....	48
ANEXO III.....	112
ANEXO IV.....	113

RESUMEN

“Saturnina” es un emprendimiento que surgió en la pandemia del 2020, destacando su enfoque en la personalización de productos de moda, desde gorras hasta camisetas y jeans, todos estos productos son creados de forma artesanal y a medida del cliente. A lo largo de los años, Saturnina ha experimentado un notable incremento en la cantidad de usuarios y que la gestión de todos los pedidos se los realice de forma manual mediante el uso de cuadernos, lo que ha generado diversos inconvenientes, tales como pérdida de información, errores humanos al registrar pedidos, demoras en la recuperación de datos de pedidos anteriores, duplicación de información y una comunicación deficiente con el equipo durante el proceso de personalización de prendas. Afectando considerablemente al modelo del negocio y la pérdida de clientes.

Para abordar la problemática antes citada, así como la creciente demanda y ampliar su alcance, se ha puesto a producción una solución tecnológica como parte de este Trabajo de Integración Curricular, el cual es el desarrollo de un frontend tipo e-commerce que permita gestionar la venta de productos personalizados, permitiendo que los clientes a través de un catálogo virtual puedan adquirir productos y que el administrador tenga un mayor control del inventario y una gestión adecuada de los pedidos.

La organización del presente documento se estructura de la siguiente manera en primer lugar, se aborda la problemática y se definen objetivos, alcance y el marco teórico. A continuación, se describe la implementación de la Metodología Scrum en el desarrollo del frontend, incluyendo la creación de prototipos y la integración de herramientas. Posteriormente, se presenta un análisis detallado de las tareas que se han realizado en cada módulo y los resultados que se han obtenido en cada iteración (Sprint). Finalmente, se ofrecen las conclusiones y recomendaciones basadas en el proceso de desarrollo, destacando el potencial impacto positivo tanto para Saturnina como para los amantes de la moda personalizada.

PALABRAS CLAVE: Frontend, Comercio electrónico, React, API, Scrum.

ABSTRACT

"Saturnina" is an undertaking that emerged during the 2020 pandemic, highlighting its focus on the customization of fashion products, from caps to T-shirts and jeans. All these products are crafted and tailored to the customer. Over the years, Saturnina has experienced a notable increase in the number of users, and the management of all orders has been done manually using notebooks, leading to various inconveniences such as loss of information, human errors in order registration, delays in retrieving data from previous orders, duplication of information, and poor communication with the team during the garment customization process. This has significantly affected the business model and resulted in customer loss.

To address the issues, meet the growing demand, and expand its reach, a technological solution has been put into production as part of this Curricular Integration Project. This solution involves the development of an e-commerce type frontend that allows managing the sale of personalized products. It enables customers to acquire products through a virtual catalog, while the administrator gains greater control over inventory and proper order management.

The organization of this document is structured as follows: first, the issues are addressed, and objectives, scope, and the theoretical framework are defined. Next, the implementation of the Scrum methodology in frontend development is described, including the creation of prototypes and tool integration. Subsequently, a detailed analysis of tasks performed in each module and the results obtained in each iteration (Sprint) is presented. Finally, conclusions and recommendations are provided based on the development process, highlighting the potential positive impact for both Saturnina and enthusiasts of personalized fashion.

KEYWORDS: Frontend, E-commerce, React, API, Scrum.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En la actualidad, Saturnina se destaca como una tienda física ubicada en la ciudad de Quito, especializada en la personalización de prendas de vestir. Su reputación se ha consolidado en el arte de convertir cualquier prenda en una auténtica obra de arte. Además de su enfoque en la personalización, la tienda también brinda una amplia gama de servicios que incluyen la restauración de prendas dañadas y la realización de bordados. A lo largo de los años, Saturnina ha experimentado un notable incremento en la cantidad de usuarios, lo que ha resultado que la gestión de pedidos se los realice de forma manual mediante el uso de cuadernos, con la integración ocasional de aplicaciones de mensajería instantánea [1].

La administración manual de pedidos ha generado diversos inconvenientes, tales como pérdida de información, errores humanos al registrar pedidos, demoras en la recuperación de datos de pedidos anteriores, duplicación de información y una comunicación deficiente con el equipo durante el proceso de personalización de prendas. Asimismo, la falta de un catálogo digital ha afectado negativamente la reputación de Saturnina en el mercado y su capacidad para atraer nuevos clientes [1].

En la actualidad, la relevancia de tener una presencia en línea y proporcionar una experiencia de compra sin contratiempos es crucial para el éxito en el sector de la personalización de prendas de vestir. La evolución tecnológica se presenta como un beneficio significativo para aquellos negocios que aprovechan este progreso. En este contexto, la tecnología facilita una conexión rápida y efectiva con los clientes. La adopción de una plataforma de comercio electrónico se vuelve imprescindible para que las empresas se mantengan competitivas y promocionen sus productos de manera eficaz [2]. Además, la presencia en línea permite la exhibición de catálogos, compras en línea desde la comodidad del hogar, comunicación en tiempo real, entre otros beneficios, llegando así a cualquier individuo con acceso a un dispositivo electrónico e internet.

Por lo antes citado, el actual Proyecto de Integración Curricular implica el desarrollo y puesta en marcha de un componente frontend como parte de un sistema web, el cual brinde a los clientes una experiencia memorable de compra a través de un

catálogo digital para explorar opciones de personalización, realizar pedidos a medida, obtener información relevante desde la comodidad de sus hogares, entre otras características relevantes al negocio. Por este motivo, el desarrollo y puesta a producción del frontend desempeña un papel fundamental en la modernización de Saturnina y su transformación en una empresa competitiva respaldada por la tecnología, asegurando la continuidad de prendas de vestir únicas y personalizadas, así como la satisfacción del cliente y una destacada presencia en el mercado.

1.1 Objetivo general

Desarrollar el frontend del e-commerce para la tienda SATURNINA.

1.2 Objetivos específicos

1. Establecer los requisitos funcionales del frontend.
2. Crear una interfaz intuitiva y amigable por medio de prototipos.
3. Codificar los módulos correspondientes en base a los usuarios.
4. Establecer una conexión segura con un componente backend que ha sido previamente desarrollado.
5. Realizar pruebas con el fin de verificar su adecuado rendimiento y funcionamiento.
6. Desplegar a producción para el uso de los administrativos y clientes de Saturnina.

1.3 Alcance

En la era actual, el comercio electrónico (e-commerce) desempeña un papel fundamental al permitir el posicionamiento de negocios on-line y la posibilidad de realizar transacciones desde la comodidad del hogar. La cual se ha visto remarcada especialmente durante los tiempos de pandemia, cuando las restricciones y precauciones de distanciamiento social han hecho que la capacidad de realizar compras en línea sea esencial [2]. En el caso de Saturnina, como un negocio especializado en la personalización de ropa, establecer una tienda virtual cobra aún

más relevancia ya que al ofrecer un catálogo electrónico de productos permite a los clientes seleccionar artículos a su gusto y completar transacciones electrónicas sin necesidad de visitar un espacio físico. Esto agrega un nivel adicional de comodidad y coherencia al proceso ya que los productos solicitados se entregan en el domicilio, brindando a los clientes una experiencia de compra segura y sin complicaciones, todo ello desde la seguridad de sus hogares [2].

Para el desarrollo del frontend además de presentar una interfaz visual atractiva, se tendrá en cuenta la accesibilidad, seguridad y rendimiento. Para lograr aquello, se ha empleado diversas herramientas, librerías y tecnologías modernas en conjunto con la integración de una metodología ágil de desarrollo, que supervisa y controla cada fase del proceso de desarrollo. Además, el frontend incluye dos perfiles de usuario, cada uno con autorizaciones distintas para acceder a los diversos módulos, como se muestra a continuación.

Perfiles: Administrador y Cliente.

El perfil de administrador puede:

- Iniciar sesión y cerrar sesión.
- Recuperar contraseña.
- Modificar su perfil de usuario.
- Gestionar categorías de productos.
- Gestionar productos.
- Gestionar pedidos de clientes.
- Gestionar comentarios y/o sugerencias.

El perfil de cliente puede:

- Registrarse.
- Iniciar sesión y cerrar sesión.
- Recuperar contraseña.
- Modificar su perfil de usuario.

- Visualizar las categorías.
- Visualizar los productos.
- Gestionar carrito de compras.
- Visualizar el historial de pedidos.
- Enviar comentarios y/o sugerencias.

1.4 Marco Teórico

Frontend

En el desarrollo de aplicaciones y sistemas software, el frontend tiene como objetivo fundamental proporcionar a los usuarios una experiencia agradable al interactuar con toda la información disponible. Actúa como la cara visible de cualquier aplicación, presentando la interfaz de usuario de manera intuitiva y visualmente atractiva. Un frontend bien diseñado garantiza que los usuarios puedan navegar sin dificultades, acceder a la información de forma organizada, personalizar sus elecciones y completar acciones de manera eficiente [3].

Figma

Es un instrumento para la creación de interfaces de usuario basada en la web de gran alcance que se ha vuelto indispensable para los diseñadores contemporáneos. Facilita la elaboración de prototipos interactivos, así como la colaboración simultánea y el diseño de cualquier tipo de aplicaciones. La característica más relevante es su accesibilidad desde la web y desde una aplicación de escritorio. Esto fomenta la colaboración y la revisión de proyectos en tiempo real, permitiendo a los diseñadores y equipos de desarrollo interactuar y dar forma a las interfaces de usuario de manera conjunta y organizada [4].

JavaScript

Es el lenguaje de programación de la web que proporciona la habilidad de añadir interactividad a las páginas web y se ha establecido como un estándar para el desarrollo web. Su utilidad se encuentra en su versatilidad para crear una amplia

variedad de funciones interactivas tales como animaciones, validación de formularios y hasta aplicaciones fullstack completas [5].

React

Es una biblioteca de JavaScript que transformó la manera en que se construyen las aplicaciones web. Permitiendo a los desarrolladores crear componentes reutilizables que forman la interfaz de usuario de una aplicación software. Además, permite actualizaciones eficientes de la interfaz en respuesta a cambios en los datos, lo que conduce a una experiencia de usuario sin contratiempos. Por último, su comunidad activa y la amplia gama de bibliotecas complementarias lo convierten en un recurso valioso para el desarrollo web moderno [6].

Node.js

Es un entorno de ejecución que permite que el código JavaScript se ejecute en el lado del servidor. Destacando su eficiencia y escalabilidad lo han convertido en una elección ideal para la construcción de todo tipo de aplicaciones que van desde aplicaciones en tiempo real y de alto rendimiento. Además, permite una programación unificada en el lado del servidor y del cliente, lo que simplifica la creación de aplicaciones completas tan solo utilizando un solo lenguaje de programación [7].

Tailwind CSS

Es una estructura de desarrollo que simplifica el diseño de las interfaces de usuario personalizadas. Su enfoque radica en clases de utilidad los que facilita la personalización de estilos y la creación de diseños coherentes. Además, la razón por la que Tailwind es tan apreciado radica en su capacidad para acelerar el desarrollo a través de una amplia gama de clases preestablecidas, esto posibilita que los desarrolladores se centren en el rendimiento en lugar de tener que crear CSS personalizado desde el principio [8].

Vite

Es una innovadora herramienta de construcción y desarrollo de aplicaciones web que se ha destacado por su velocidad y un entorno de desarrollo rápido y eficiente. Además, ofrece una carga rápida de módulos, lo que reduce significativamente el

tiempo de desarrollo y mejora la experiencia del desarrollador, por esta razón Vite es especialmente valioso para proyectos web basados en JavaScript y TypeScript en busca de un rendimiento óptimo [9].

API RESTful

Es un estilo de arquitectura de software que facilita la transferencia de datos entre varias aplicaciones utilizando el protocolo HTTP en conjunto con sus verbos estándar que realizan operaciones en los recursos de una aplicación software. Esta arquitectura proporciona una solución escalable, que es a la vez fácil de entender y aplicar, lo que permite a los desarrolladores diseñar aplicaciones ya sea para el entorno web o para dispositivos móviles, y establecer una comunicación eficiente con varios sistemas backend [10]. La importancia de una API RESTful radica en su capacidad para unir aplicaciones y servicios, permitiendo que estas compartan información y funcionen de manera conjunta, incluso si han sido desarrolladas con diferentes tecnologías o para distintos propósitos.

2 METODOLOGÍA

Clásicamente, la investigación a través del análisis de casos ha sido empleada como un método investigativo que se concentra en lograr los resultados esperados mediante un estudio eficaz. En otras palabras, se refiere a una estrategia que se sustenta en la observación y análisis de un caso específico, con el objetivo de alcanzar conclusiones y recopilar la evidencia necesaria para respaldar la validez de dichas conclusiones [11].

En el ámbito de este proyecto de tesis, se ha aplicado el estudio de casos para evaluar la presencia en línea de Saturnina. Los hallazgos mencionaban que la tienda carece de una plataforma virtual, desaprovechando las oportunidades que este enfoque puede ofrecer. En respuesta, se ha implementado un componente frontend con el propósito de permitir al equipo de Saturnina expandir su alcance, aumentar las ventas y gestionar de manera eficaz su modelo de negocio y procesos internos. Esta estrategia se orienta en abordar de manera específica los desafíos y oportunidades únicas que Saturnina enfrenta dentro del ámbito electrónico y la administración de solicitudes en línea.

2.1 Metodología de Desarrollo

Una metodología es un enfoque estratégico que se utiliza con el objetivo principal de potenciar la eficiencia en la entrega de un proyecto software. Este método permite la realización de entregas incrementales a lo largo de las diferentes fases del proyecto, eliminando la necesidad de esperar hasta la finalización del proyecto para obtener el producto final [12].

La técnica "ágil" es reconocida y ampliamente utilizada para garantizar una entrega eficaz en todos los proyectos. Esta metodología permite una rápida adaptación a cualquier cambio, ya sea tecnológico, de requisitos o de equipo durante la ejecución del proyecto. Además, la agilidad se distingue por su flexibilidad y habilidad para realizar entregas progresivas a través de las diferentes etapas del proyecto, eliminando la necesidad de esperar hasta su conclusión [13].

En concordancia con lo previamente expuesto, para este proyecto de tesis se ha implementado Scrum, el cual es una de las metodologías ágiles que promueve la

colaboración, la entrega gradual de funcionalidades y la capacidad de adaptación a cualquier eventualidad que pueda surgir durante la ejecución del proyecto [14]. A continuación, se proporciona una exposición más detallada sobre la incorporación de esta metodología ágil, abordando sus fases, roles y artefactos.

Roles

La metodología ágil Scrum comprende tres roles esenciales denominados *Product Owner*, *Development Team* y *Scrum Master*. Cada uno de estos roles desempeña funciones específicas y requiere habilidades particulares para fomentar la adaptabilidad, la innovación y la eficiencia a lo largo de la ejecución del proyecto [15]. Posteriormente, se detallan los integrantes asignados a cada rol.

Product Owner

Comúnmente conocido como el dueño del producto, este miembro del equipo tiene la responsabilidad de mejorar la eficiencia del grupo de desarrollo con el fin de maximizar su desempeño. Actúa como representante del cliente final y gestiona las prioridades en el *Product Backlog*. Además, revisa de manera conjunta los avances y resultados presentados por el equipo de desarrollo [16]. La **Tabla 2.1** presenta la identificación de la persona designada en este contexto.

Scrum Master

Garantiza y verifica la aplicación de las buenas prácticas de Scrum a lo largo de todo el desarrollo del proyecto. Su función principal consiste en eliminar cualquier obstáculo o impedimento que pueda afectar las habilidades del equipo. Además, impulsa cambios que mejoren la productividad y el éxito del proyecto, brindando motivación y apoyo constante [16]. En este contexto, la **Tabla 2.1** muestra la identificación de la persona designada.

Development Team

Conjunto de profesionales, con la responsabilidad de entregar un producto o funcionalidad terminada, asegurando una cantidad adecuada de integrantes para cada tarea. Estos profesionales se organizan y autoorganizan para gestionar eficientemente el trabajo en cada una de las iteraciones destinadas a la entrega del proyecto. Además, el grupo cuenta con diversas habilidades para desarrollar el

producto y cumplir con los entregables en los tiempos establecidos [16]. En la **Tabla 2.1** se proporciona la identificación de la persona designada en este contexto.

Tabla 2.1 Asignación de Roles.

ROL	INTEGRANTES
<i>Product Owner</i>	Sra. Shirley Rivadeneria.
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development Team</i>	Miguel Angel Carapaz Mendoza

Artefactos

En Scrum, los artefactos representan el trabajo ejecutado o el valor generado, buscando incrementar la transparencia de información esencial. Este enfoque permite que todos los integrantes accedan a la misma base de conocimientos y puedan realizar ajustes de manera correspondiente. De esta manera, los tres artefactos más significativos en Scrum incluyen la obtención de requisitos, el *Producto Backlog* y el registro del *Sprint Backlog*. [17]. Posteriormente, se describe su implementación como parte del proceso de desarrollo del componente frontend.

Recopilación de Requerimientos

Identificar los requerimientos del proyecto es una tarea crucial que juega un rol significativo en el éxito de un proyecto software. Asimismo, ayuda en la evaluación de los plazos y la extensión del proyecto, posibilitando una organización adecuada de los entregables y contribuyendo a los objetivos del proyecto [18]. Por este motivo, se ha registrado todos los requisitos funcionales del proyecto en la **Tabla 2.2**, la cual se ha compilado mediante varias reuniones con el *Product Owner*. Además, se puede consultar el **ANEXO II** para obtener una referencia más completa.

Tabla 2.2 Conjunto de requisitos.

EJEMPLO RECOPIACION DE REQUERIMIENTOS		
EJEMPLO SISTEMA	EJEMPLO - RR	EJEMPLO ENUNCIADO
Frontend	RR-005	Como administrador, es necesario consumir endpoints para: <ul style="list-style-type: none"> • Gestionar productos
	RR-006	Como administrador, es necesario consumir endpoints para: <ul style="list-style-type: none"> • Gestionar pedidos.

Historias de Usuario

Es una ficha empleada para detallar las características que el producto software debe poseer siendo considerada como la menor entidad de trabajo dentro de un contexto ágil. Estas fichas son simples y de fácil memorización, siguen una estructura y contienen elementos específicos [19]. Para el componente frontend se presenta un ejemplar como se presenta en la **Tabla 2.3**, por otra parte, en el **ANEXO II** se encuentra más detalles al respecto de las tablas que han sido elaboradas de acuerdo con los requerimientos funcionales previamente recopilados y detallados.

Tabla 2.3 Ejemplar de Historia de Usuario para el componente frontend.

EJEMPLO HISTORIA DE USUARIO	
Clave: HU-005	Rol: Perfil Administrador
Ejemplo Historia: Gestionar productos	
Urgencia: Alta	Riesgo desarrollo: Media
Asignación de iteración: 2	
Encargado (s): Miguel Carapaz	
Detalle: El administrador en el frontend debe consumir endpoints para:	

<ul style="list-style-type: none"> • Realizar el registro de productos. • Realizar cambios en productos. • Eliminar productos. • Visualizar productos.
<p>Análisis: El administrador ingresa a estas vistas después de iniciar sesión para administrar los productos.</p>

Product Backlog

Es un registro secuencial de actividades planificadas, las cuales tendrán lugar durante la ejecución del proyecto. En este listado se encuentran tanto las tareas fundamentales como las complementarias ordenadas según su importancia en el desarrollo [20]. Para la creación del *Product Backlog*, se ha utilizado el formato presentado en la **Tabla 2.4**, que proporciona información sobre la iteración, la condición y la importancia de las diversas actividades provenientes de las Historias de usuario. Además, para acceder a la tabla completa, se puede consultar el **ANEXO II**.

Tabla 2.4 Exhibición del *Product Backlog*.

HU	EJEMPLO HISTORIA DE USUARIO	ITERACIÓN	CONDICIÓN	IMPORTANCIA
HU005	Gestionar productos	2	Finalizado	Alta
HU006	Gestionar pedidos	2	Finalizado	Alta

Sprint Backlog

Después de la formulación del *Product Backlog* se genera una relación más pormenorizada de las actividades a realizar, las cuales deben ser estructuradas y categorizadas mediante Sprints o iteraciones. De esta manera, cada actividad posee un plazo definido para su conclusión [21]. La disposición establecida para la

presentación del *Sprint Backlog* de este proyecto se exhibe en la **Tabla 2.5** y la lista completa está disponible en el **ANEXO II**.

Tabla 2.5 Exhibición del *Sprint Backlog*.

EJEMPLO DEL SPRINT BACKLOG						
ID	ETIQUETA	MÓDULO	HU	HISTORIA	TAREA	TIEMPO ESTIMADO
SB-002	Diseño y desarrollo de los módulos para el usuario Administrador	Módulo Gestión de Productos	HU005	Gestionar Productos	<ul style="list-style-type: none"> Diseño e implementación de las interfaces para poder gestionar los productos Validación de los campos requeridos 	80 H

2.2 Diseño de interfaces

Es el procedimiento que los desarrolladores realizan para generar interfaces destinadas a sistemas software o dispositivos computarizados, destacando su aspecto y diseño. La meta consiste en concebir interfaces que resulten intuitivas y visualmente atractivas para los usuarios, simplificando su empleo y brindando una experiencia agradable [22]. Seguidamente, se expone el software que se ha seleccionado para desarrollar dichas interfaces.

Herramienta utilizada para el diseño

Figma, es una herramienta basada en la web reconocida por su utilidad en prototipados web y edición de gráficos vectoriales se ha seleccionado estratégicamente para la creación de interfaces dentro del contexto del proyecto Saturnina. Su característica distintiva, al operar a través del navegador, facilita la posibilidad de compartir el proyecto con todo el equipo y llevar a cabo ajustes colaborativos en un espacio de trabajo compartido [23]. En virtud de esta elección, se ha empleado Figma para la creación de todos los prototipos y muestra de ello

se presenta un ejemplar en la **Figura 2.1**, mientras que los demás prototipos relacionados con Saturnina se encuentran detallados en el **ANEXO II**.



Figura 2.1 Landing Page Saturnina.

2.3 Diseño de la arquitectura

El desarrollo de una aplicación o sistema de software involucra una interacción entre diversos elementos internos y este procedimiento se gestiona a través de una planificación estructura que permita su escalabilidad [24]. A continuación, se expone el modelo arquitectónico que se ha utilizado en el desarrollo y ejecución del frontend.

Patrón arquitectónico

La estructura de diseño denominada Modelo-Vista-Controlador (MVC) se emplea extensamente en el desarrollo de software que busca separar las interfaces, los datos y la lógica de control de manera independiente. Esta estructura procura una clara separación de preocupaciones, dividiendo la lógica del negocio de la visualización, lo que facilita tanto la organización del trabajo como el mantenimiento del código [25]. A continuación, se ofrece una explicación minuciosa de los estratos mencionados anteriormente:

- **Modelo:** encargado de gestionar la información y la lógica del negocio.
- **Vista:** encargado de mostrar de manera gráfica la información al usuario.

- **Controlador:** registra las acciones del usuario y dirige los comandos hacia el modelo o la vista correspondiente.

La **Figura 2.2** exhibe el modelo que se ha aplicado para desarrollar el frontend, asegurando la compatibilidad con otras herramientas y librerías.

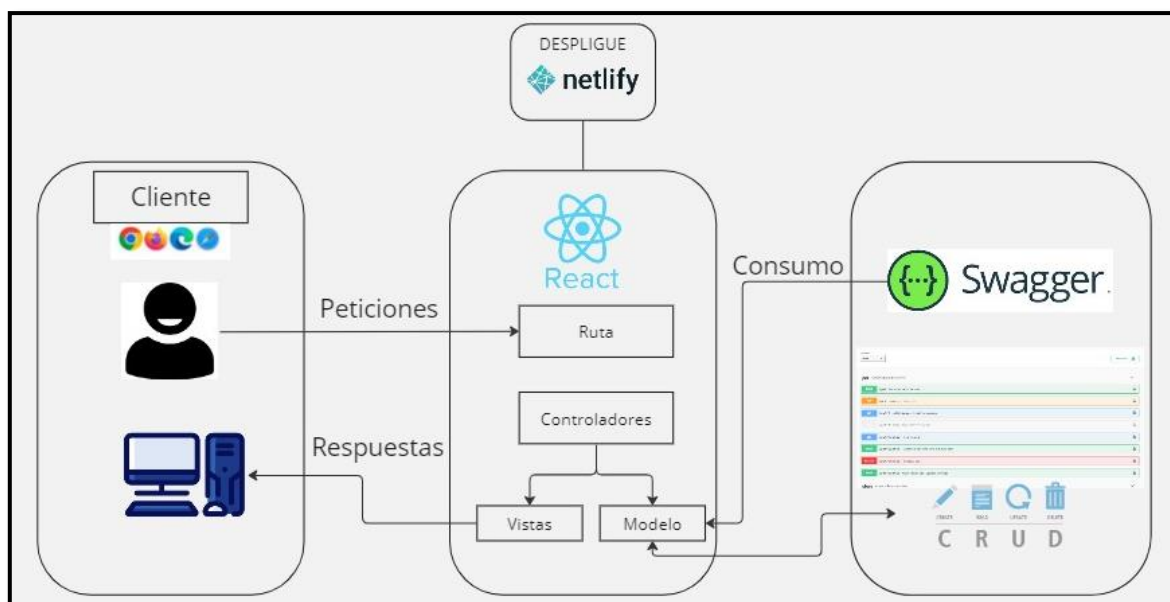


Figura 2.2 Esquema Modelo-Vista-Controlador (MVC).

2.4 Herramientas de desarrollo

Son aplicaciones informáticas que los equipos de desarrollo de software utilizan para diseñar, corregir errores, administrar, brindar soporte a aplicaciones y otros programas. Estas herramientas son comúnmente denominadas como herramientas de codificación de software [26]. Por lo antes citado, se han seleccionado minuciosamente las herramientas primordiales para la implementación de este componente, tomando en consideración la complejidad del proyecto, las cuales están detalladas en la **Tabla 2.6**.

Tabla 2.6 Herramientas para el componente Frontend.

HERRAMIENTAS	JUSTIFICACIÓN
GitHub	Facilita una administración eficaz del control de versiones del código en el proyecto frontend [27].

React	Permite construir interfaces de usuario modulares y reutilizables, facilitando el mantenimiento del código y proporcionando una experiencia de usuario más dinámica y eficiente para el frontend [28].
Vite	Acelera el proceso de desarrollo al proporcionar una compilación rápida y eficiente, lo que mejora significativamente el tiempo de carga de la aplicación frontend, brindando una experiencia más ágil a los usuarios [29].
Bootstrap	Facilita la planificación y apariencia de la interfaz de usuario al ofrecer componentes preestablecidos y un sistema de cuadrícula que simplifica la elaboración de páginas receptivas y estéticamente agradables en el frontend [30].

2.5 Librerías

En la etapa de desarrollo de software se utilizan diversas librerías que ofrecen numerosos beneficios, garantizando la inexistencia de fallos durante el proceso de compilación, pruebas e implementación en entornos de producción. A diferencia de un marco de trabajo, una librería no establece una estructura específica para llevar a cabo determinadas tareas, sino que se concibe como un complemento adicional [31]. En la **Tabla 2.7** se presenta un registro de las bibliotecas que se han utilizado en este contexto.

Tabla 2.7 Librerías para el componente Frontend.

LIBRERÍAS	DESCRIPCIÓN
form-data	Simplifica la construcción de objetos formulario para el envío de datos, especialmente útil en operaciones de carga de archivos [32].
jwt-decode	Permite decodificar tokens JWT, facilitando la gestión de la autenticación y autorización en el frontend [33].
react-redux	Facilita la integración de Redux con React, mejorando la gestión del estado en aplicaciones frontend complejas [34].

react-loading-skeleton	Facilita la incorporación fácil de estructuras de carga con el objetivo de optimizar la experiencia de usuario durante la carga de contenido en el frontend [35].
Axios	Simplifica las solicitudes HTTP, permitiendo una comunicación fluida entre el frontend y el servidor, facilitando la obtención y actualización de datos en tiempo real, mejorando así la interactividad y dinamismo de la aplicación [36].

3 RESULTADOS

Durante esta fase, se presentan los resultados que se han obtenido en cada uno de los aspectos del frontend mediante Sprints, los cuales se detallan de una manera amplia dentro del **ANEXO II**.

Sprint 0. Establecimiento de la configuración para el entorno de desarrollo.

En este Sprint, se describen las tareas que han sido determinadas en la planificación del Sprint Backlog, las cuales son:

- Restricciones sobre los requisitos.
- Elaboración de la estructura del frontend.
- Definición roles de usuarios.

Restricciones sobre los requisitos

Portal informativo

El componente frontend tiene la posibilidad de presentar una página informativa que incluye secciones como inicio, productos, contacto, registro y login. Además, es relevante destacar que no se requiere el inicio de sesión en el frontend para la visualización de toda la información.

Aplicación de endpoints para llevar a cabo la operación de registro de usuario

En el frontend, aquellos miembros identificados como clientes cuentan con la habilidad de llevar a cabo el procedimiento de registro. Para alcanzar este objetivo, resulta fundamental el consumo de endpoints que han sido integrados en la estructura de un backend previamente desarrollado.

Aplicación de endpoints para realizar las operaciones de iniciar sesión, cerrar sesión y restablecer contraseña

En el frontend, aquellos miembros identificados como administrador y clientes pueden obtener acceso mediante el uso del correo electrónico y la contraseña. Con el fin de alcanzar este objetivo, resulta fundamental que el módulo frontend utilice los endpoints que han sido proporcionados por el backend. De igual modo, para el cierre de sesión y modificar la contraseña, se requiere el uso de los endpoints respectivos por parte del backend.

Aplicación de endpoints para llevar a cabo la operación de modificar el perfil de usuario

En el frontend, aquellos usuarios identificados como administradores y clientes tienen la habilidad de ingresar a su perfil de usuario. Para este propósito, es necesario que el frontend haga uso de los endpoints que han sido proporcionados por el backend. De igual manera, para efectuar cambios en los perfiles de dichos usuarios, resulta esencial que el frontend haga uso del endpoint de manera apropiada.

Aplicación de endpoints para llevar a cabo la operación de gestión de categorías

En el frontend, aquellos usuarios identificados como administradores poseen la capacidad de gestionar las categorías. Para este propósito, es necesario que el frontend consuma los endpoints que han sido proporcionados por el backend, el cual posibilita la visualización de las categorías existentes. De manera similar, para llevar a cabo modificaciones en dichas categorías, resulta esencial que el frontend haga uso del endpoint correspondiente.

Aplicación de endpoints para llevar a cabo la operación de gestión de productos

En el frontend, aquellos usuarios identificados como administradores tienen la capacidad de gestionar los productos. Para este propósito, es esencial que el frontend consuma el endpoint generado por el backend, el cual posibilita la

visualización de los productos existentes. Del mismo modo, con el propósito de efectuar alteraciones en los productos previamente mencionados, se requiere que el frontend utilice su endpoint correspondiente.

Aplicación de endpoints para llevar a cabo la operación de gestión de pedidos

En el frontend, aquellos usuarios identificados como administradores y clientes tienen la capacidad de gestionar pedidos. Para este propósito, es esencial que el frontend consuma los endpoints desarrollados por el backend, que posibilita la visualización de pedidos existentes. De manera similar, con el fin de realizar cambios en los pedidos previamente expuestos, se requiere que el frontend utilice el endpoint correspondiente.

Aplicación de endpoints para llevar a cabo la operación de visualización del estado del pedido

En el contexto del entorno frontend, aquellos usuarios identificados como clientes pueden visualizar los estados para los pedidos. Con esta finalidad, es indispensable que el frontend consuma los endpoints que han sido generados por el backend, el cual facilita la visualización en los estados de los pedidos.

Aplicación de endpoints para llevar a cabo la operación de modificación del estado del pedido

En el contexto del entorno frontend, aquellos usuarios identificados como administradores pueden modificar los estados para los pedidos. Para ello, es indispensable que el frontend consuma los endpoints que han sido implementados por el backend, el cual facilita la modificación en los estados de los pedidos.

Aplicación de endpoints para llevar a cabo la operación de gestión del carrito de compras

En el frontend, aquellos usuarios identificados como clientes tienen la capacidad de administrar su carrito de compras. Para este propósito, es necesario que el frontend consuma los endpoints que han sido implementados por el backend, el cual posibilita la gestión del carrito de compras.

Aplicación de endpoints para llevar a cabo la operación de visualización del detalle del pedido.

En el contexto del entorno frontend, aquellos usuarios identificados como administradores y clientes pueden observar los detalles de los pedidos. Para ello, es indispensable que el frontend consuma los endpoints que han sido generados por el backend, el cual posibilita la visualización del detalle del pedido.

Aplicación de endpoints para llevar a cabo la operación de envío de comentarios y/o sugerencias

En el frontend, aquellos usuarios identificados como clientes pueden enviar comentarios y/o sugerencias. Con el fin de lograrlo, resulta crucial que el frontend haga uso de los endpoints que han sido generados por el backend, el cual posibilita el envío de comentarios y/o sugerencias.

Aplicación de endpoints para llevar a cabo la operación de gestión de comentarios y/o sugerencias

En el frontend, aquellos miembros identificados como administradores pueden gestionar los comentarios y/o sugerencias generadas por los usuarios clientes. Para ello, es necesario que el frontend consuma los endpoints que han sido generados por el backend.

Por último, la **Figura 3.1** presenta un gráfico que ilustra las diversas capacidades de aquellos usuarios categorizados como administradores y clientes. El propósito de este recurso ilustrativo es incrementar la comprensión e interpretación sobre las características disponibles.

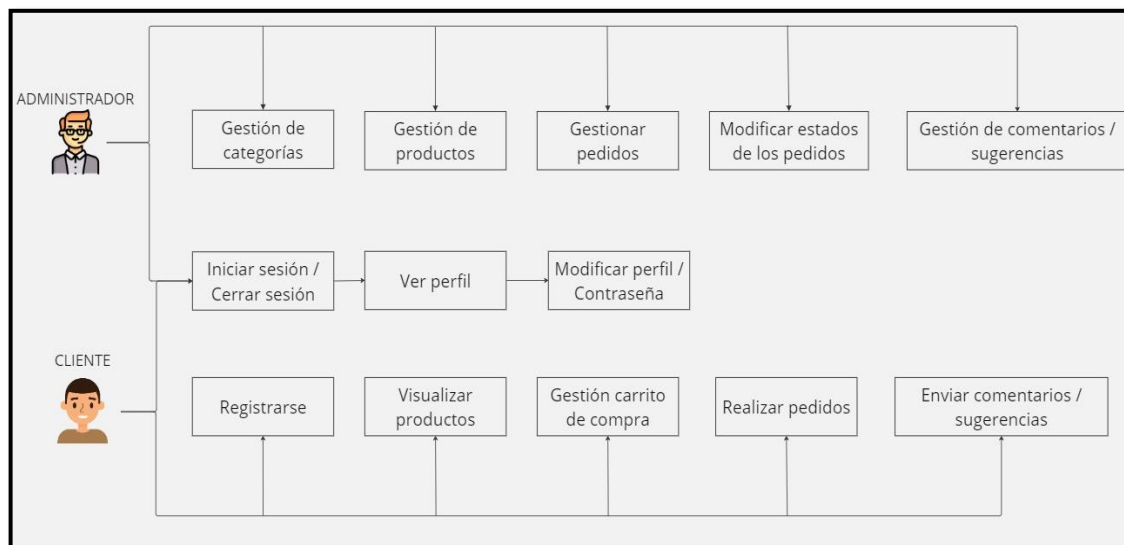


Figura 3.1 Categorías del perfil usuario y las tareas a realizar.

Elaboración de la estructura del proyecto frontend

La creación del módulo frontend se realiza empleando Visual Studio Code, una herramienta indispensable en la configuración del proyecto, incluyendo archivos de configuración y directorios fundamentales del frontend. Además, en la **Figura 3.2** se presenta la planificación detallada del proyecto.

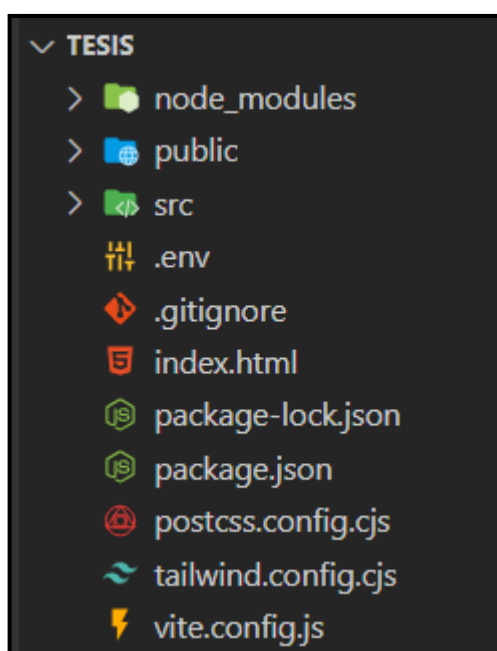


Figura 3.2 Configuración del proyecto en Visual Studio Code.

Definición de roles de usuario

En la **Figura 3.3** se ilustra los roles asociados al frontend que incluyen las figuras de administrador y cliente. El propósito de esta representación visual es elevar la comprensión y el conocimiento de las diversas funciones relacionadas con cada uno de estos roles.



Figura 3.3 Roles y tareas asignadas.

Sprint 1. Diseño de interfaces destinadas al usuario con perfil – administrador.

En este Sprint, se describen las tareas que han sido determinadas en la planificación del Sprint Backlog, las cuales son.

- Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de inicio de sesión, cierre de sesión y restablecimiento de contraseña.

- Aplicación de endpoints que han sido generados por el backend para llevar a cabo las operaciones de modificación y visualización en el perfil del usuario.
- Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de categorías.
- Aplicación de endpoints que han sido generados por el backend para llevar a cabo las operaciones de gestión de productos.
- Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de pedidos.
- Aplicación de endpoints que han sido generados por el backend para llevar a cabo la operación de visualización de detalles de pedidos.
- Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de comentarios y/o sugerencias.

Aplicación de endpoints generados por el backend para realizar las operaciones de inicio de sesión, cierre de sesión y restablecimiento de contraseña

En el frontend, aquellos miembros identificados como administradores tienen la capacidad de ingresar al componente con un correo electrónico y una contraseña suministrados por el backend. Además, para usuarios con perfil cliente pueden ingresar con las credenciales proporcionadas durante su registro. Una vez dentro del frontend, se ofrece a los usuarios la opción de cerrar sesión, además de la oportunidad de modificar o restablecer su contraseña según lo estimen necesario. Para llevar a cabo estas acciones, se les solicita completar campos que incluyen el correo electrónico, nueva contraseña y su confirmación. Tal y como se presenta en las **Figura 3.4**, **Figura 3.5** y **Figura 3.6**, mientras los detalles de las pruebas unitarias se pueden contemplar por las **Figura 3.7**, **Figura 3.8** y **Figura 3.9**, además, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

The screenshot shows the login page for 'saturnina'. At the top, there are navigation links: 'Inicio', 'Productos', and 'Comentarios'. The main heading is 'Iniciar sesión'. On the left, there is a decorative illustration of a woman sitting at a desk with a laptop and a coffee cup. On the right, there are two input fields: 'Correo electrónico' with the value 'ejemplo@gmail.com' and 'Contraseña' with masked characters '*****'. Below these fields are two links: '¿No tienes una cuenta? [Regístrate aquí](#)' and '¿Olvidaste tu contraseña? [Recupérala](#)'. At the bottom right, there is a dark button labeled 'Iniciar sesión'.

Figura 3.4 Inicio de sesión.

The screenshot shows a user profile dropdown menu. It features a user icon on the left, the text 'Miguel Carapaz Admin' in the center, and a dark button labeled 'Salir' on the right.

Figura 3.5 Cerrar sesión.

The screenshot shows the account recovery page for 'saturnina'. At the top, there are navigation links: 'Inicio', 'Productos', and 'Comentarios'. The main heading is 'Recuperar Cuenta'. A central white box contains the text: 'Introduzca su correo electrónico para el proceso de recuperación, le enviaremos un correo electrónico.' Below this text is an input field for 'Correo electrónico' with the value 'ejemplo@gmail.com'. At the bottom of the box is a dark button labeled 'Recuperar Cuenta'.

Figura 3.6 Recuperar cuenta.

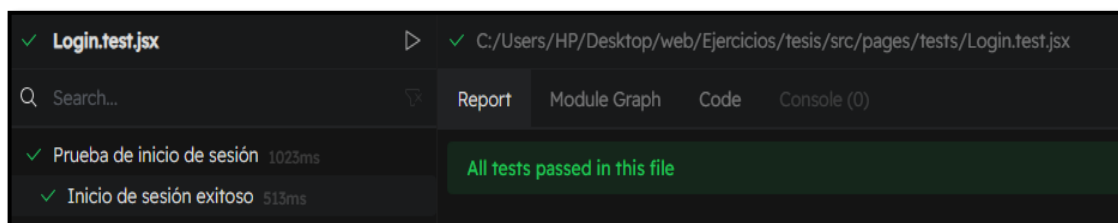


Figura 3.7 Test unitario - iniciar sesión.

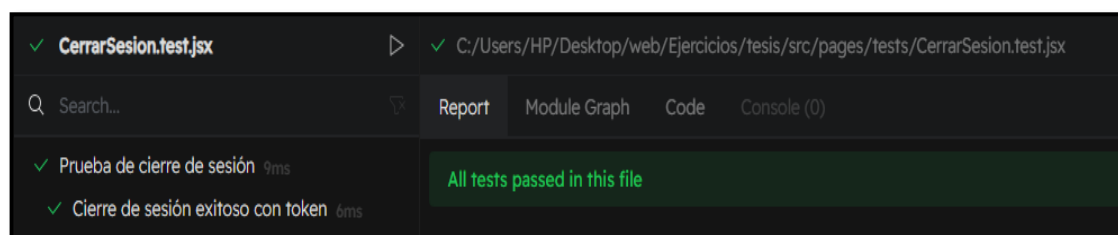


Figura 3.8 Test unitario - cerrar sesión.

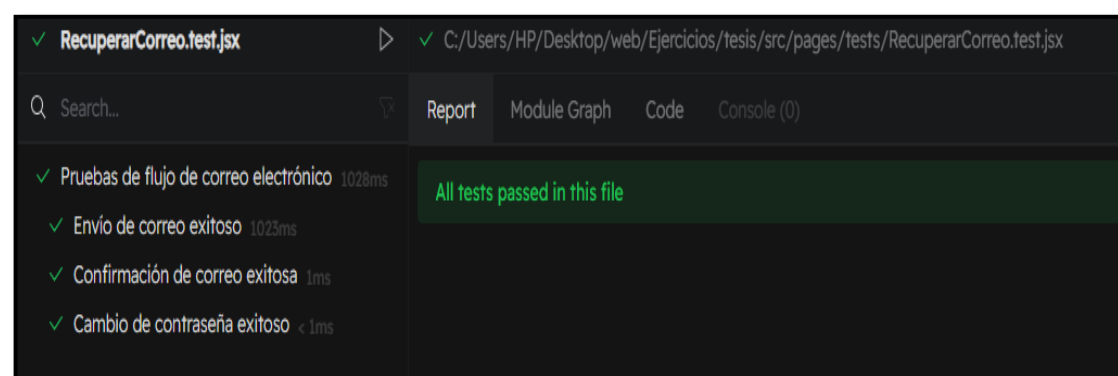


Figura 3.9 Test unitario - modificar contraseña.

Aplicación de endpoints que han sido generados por el backend para llevar a cabo las operaciones de modificación y visualización en el perfil del usuario

En el frontend, aquellos miembros identificados como administradores y clientes pueden visualizar y modificar su respectivo perfil de usuario. Los cuales abarcan información como nombres, apellidos, correo electrónico, número de teléfono y celular. Estas operaciones son realizadas a través de los endpoints que han sido proporcionados por el backend. El cual en la **Figura 3.10** se exhibe la parte visual

de la interfaz de usuario vinculada al perfil de usuario, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.11**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

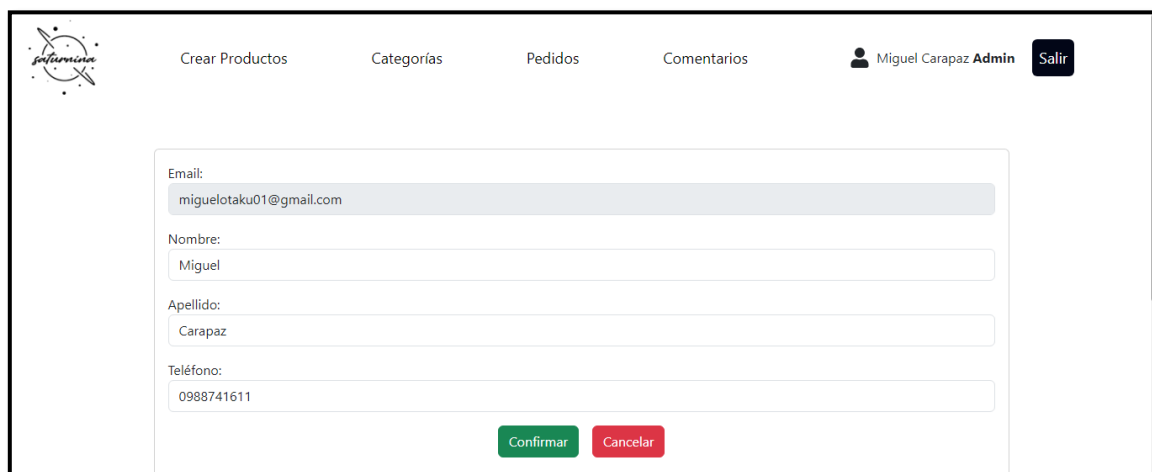


Figura 3.10 Modificar perfil.

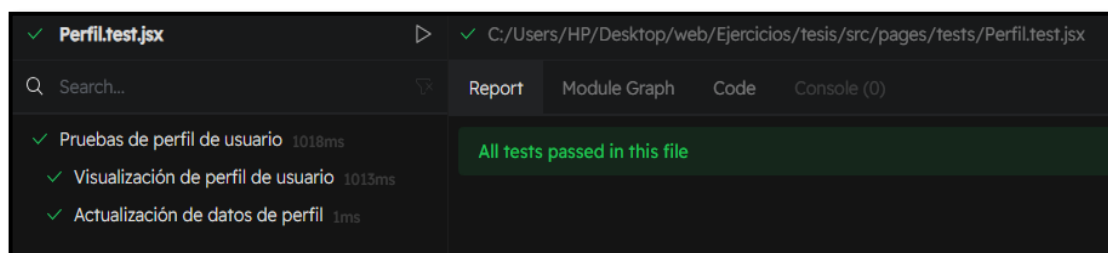


Figura 3.11 Test unitario - visualizar y modificar perfil.

Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de categorías

En el frontend, aquellos miembros identificados como administradores tienen la capacidad de gestionar (CRUD) categorías en el frontend, abarcando actividades asociadas a los productos, las cuales se llevan a cabo mediante los endpoints que han sido generados por el backend. La **Figura 3.12** muestra la parte visual de la interfaz de usuario vinculada a la funcionalidad de listar categorías, de igual modo, el detalle de la prueba unitaria se puede observar en la **Figura 3.13**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

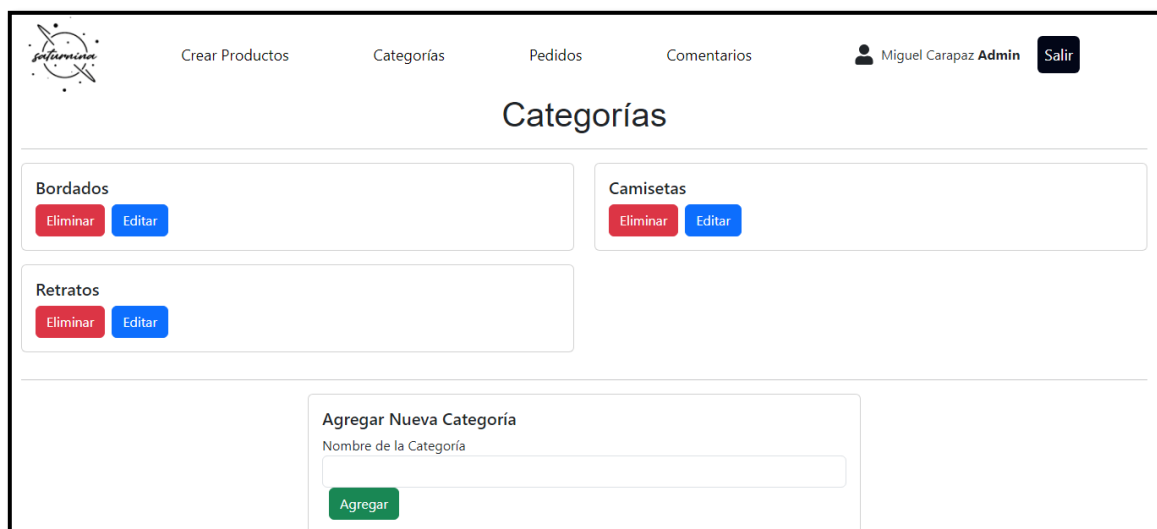


Figura 3.12 Listado de categorías.

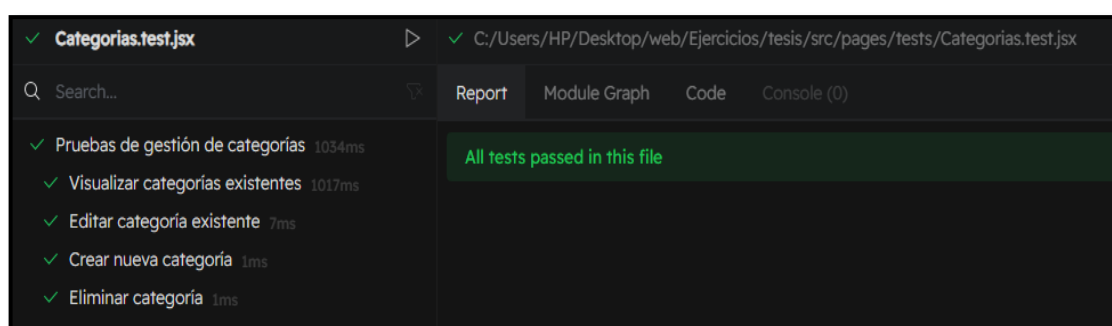


Figura 3.13 Test unitario - manejo categorías.

Aplicación de endpoints que han sido generados por el backend para llevar a cabo las operaciones de gestión de productos

Una vez dentro de los módulos del frontend, los usuarios clasificados como administradores poseen la capacidad para gestionar (CRUD) productos en el frontend. Este manejo implica actividades como visualizar, registrar, modificar o eliminar productos, los cuales son acciones realizadas mediante los endpoints que han sido generados en el backend. La **Figura 3.14** muestra la parte visual de la interfaz de usuario vinculada a la actividad de listar productos, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.15**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

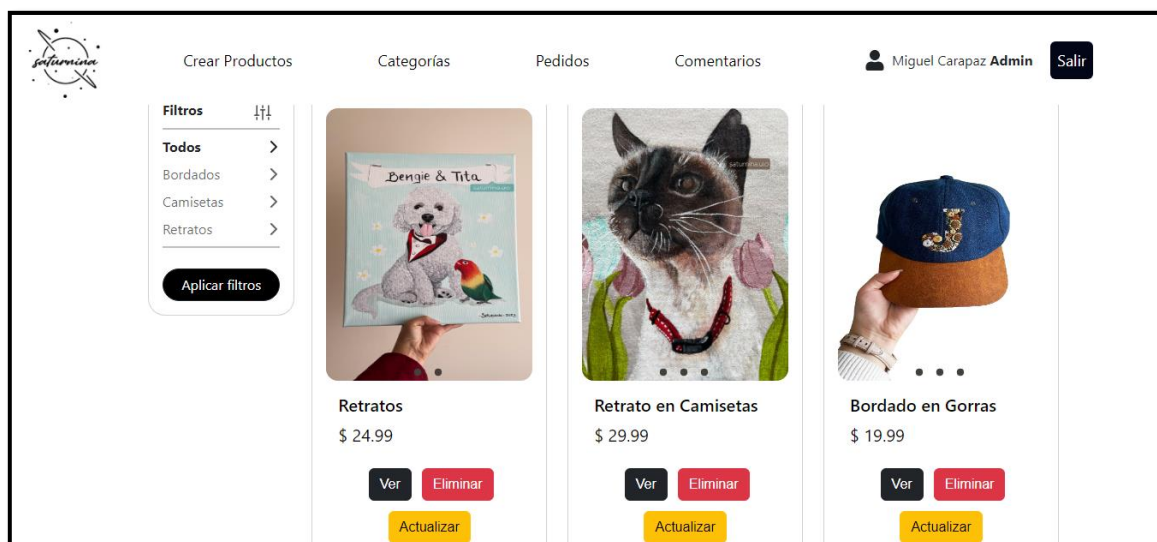


Figura 3.14 Listado de productos.

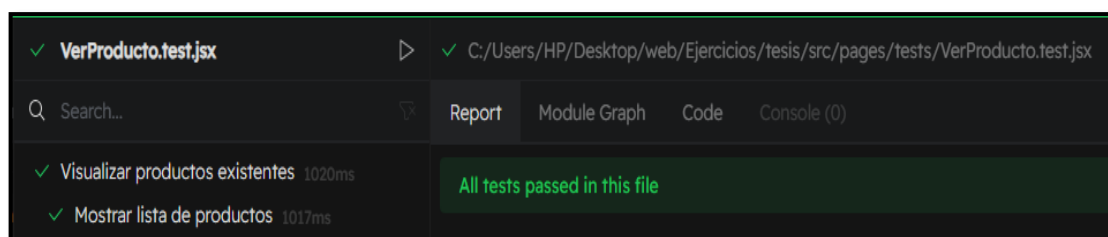


Figura 3.15 Test unitario - gestión productos.

Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de pedidos.

En el frontend, aquellos miembros identificados como administradores pueden gestionar los pedidos. Esto abarca acciones tales como listar y modificar el estado del pedido mediante los endpoints que han sido generados por el backend. La **Figura 3.16** muestra la parte visual de la interfaz de usuario vinculada a la actividad de listar pedidos, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.17**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.



Figura 3.16 Gestión pedidos.

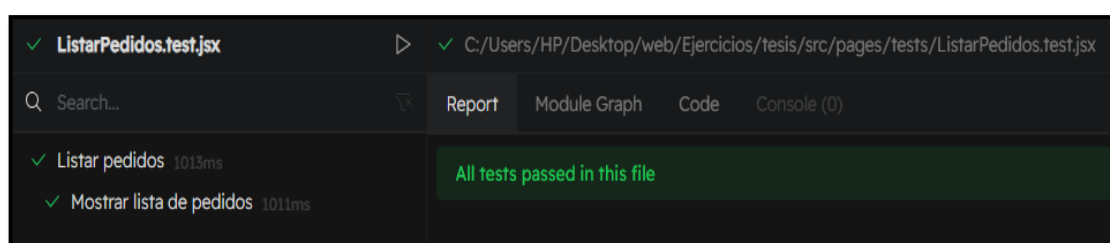


Figura 3.17 Test unitario - gestión de pedidos.

Aplicación de endpoints que han sido generados por el backend para llevar a cabo la operación de visualización de detalles de pedidos

En el frontend, los usuarios clasificados como administradores y clientes pueden visualizar los detalles de los pedidos. Esto involucra la capacidad de ver el detalle específico de cada pedido, los cuales se logran consumiendo los endpoints que han sido generados por el backend. La **Figura 3.18** muestra la parte visual de la interfaz vinculada a esta funcionalidad, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.19**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

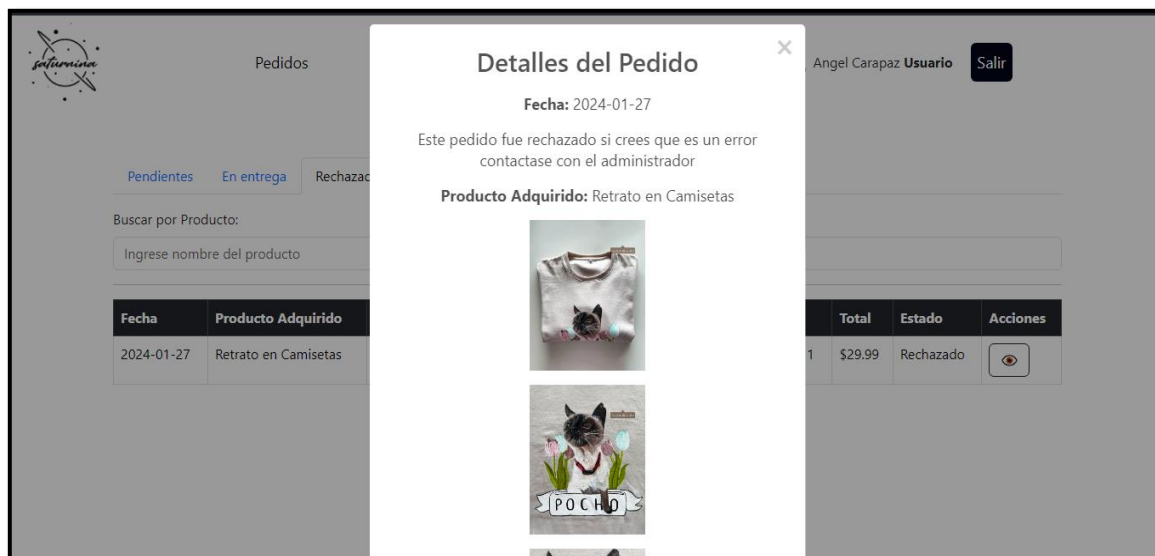


Figura 3.18 Detalles de un pedido.

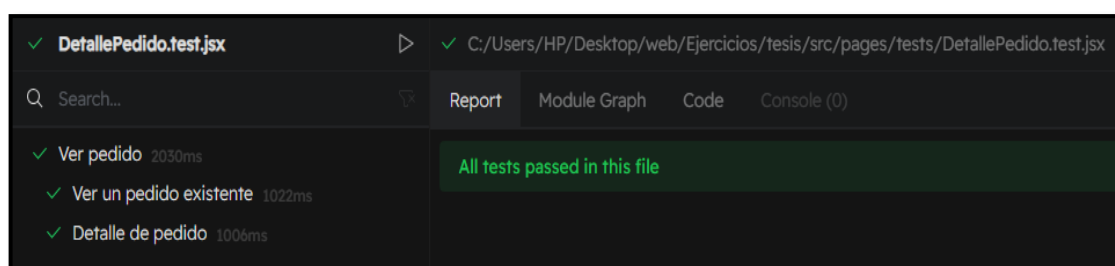


Figura 3.19 Test unitario - detalles de un pedido.

Aplicación de endpoints que han sido generados por el backend para realizar las operaciones de gestión de comentarios y/o sugerencias

En el frontend, aquellos miembros identificados como administradores pueden gestionar comentarios y/o sugerencias. Todo ello, mediante el consumo de endpoints que han sido generados por el backend. La **Figura 3.20** muestra la parte visual de la interfaz de usuario vinculada a la funcionalidad, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.21**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

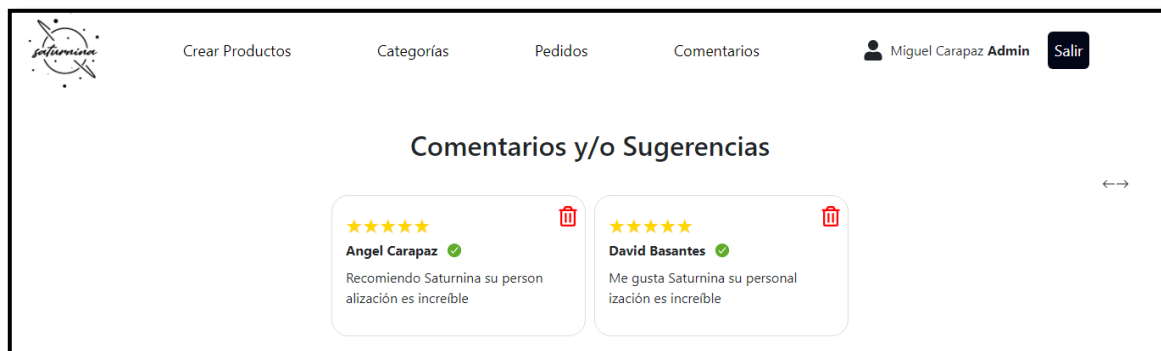


Figura 3.20 Gestión de comentarios.

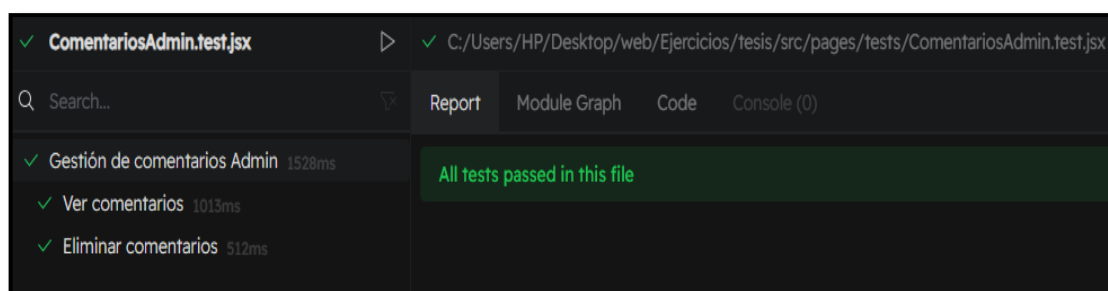


Figura 3.21 Test unitario - gestionar comentarios y/o sugerencias.

Sprint 2. Diseño de interfaces destinadas al usuario con perfil - cliente.

En este Sprint, se describen las tareas que han sido determinadas en la planificación del Sprint Backlog, las cuales son.

- Aplicación de endpoints para realizar la operación de visualización de productos.
- Aplicación de endpoints para llevar a cabo la operación de gestión del carrito de compras.
- Aplicación de endpoints para realizar la operación de crear/actualizar comentarios.

Aplicación de endpoints para realizar la operación de visualización de productos

En el frontend, aquellos miembros identificados como clientes pueden visualizar sus productos. Abarcando actividades tales como observar los artículos adquiridos por el cliente, mismos que son llevados a cabo a través del consumo de endpoints que han sido generados por el backend. La **Figura 3.22** muestra la parte visual de la interfaz de usuario vinculada a la funcionalidad, donde se pueden apreciar los detalles visuales relacionados con la visualización de productos. Por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.23**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

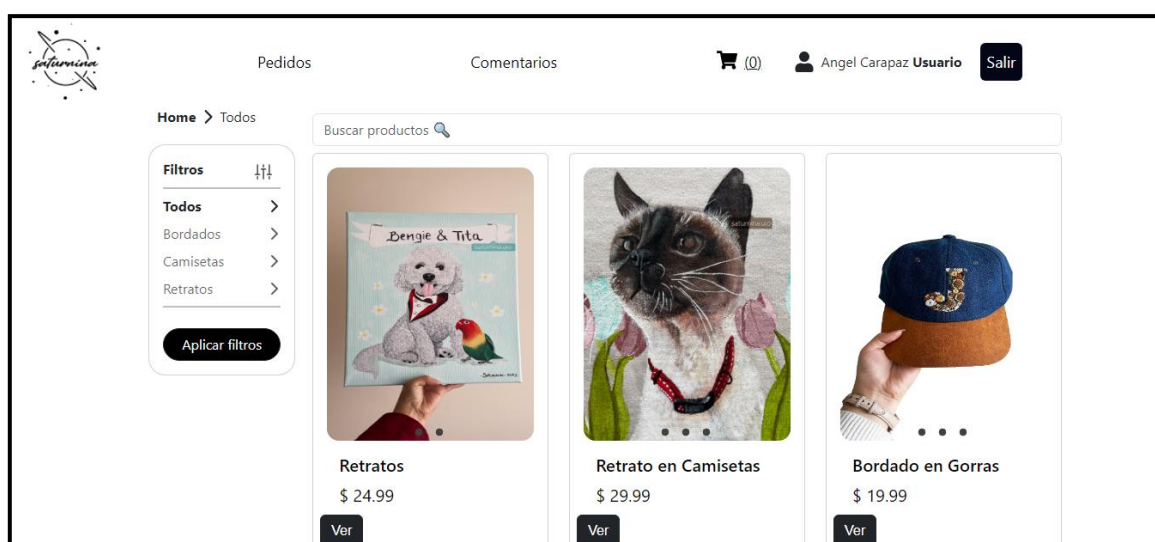


Figura 3.22 Visualización de productos.

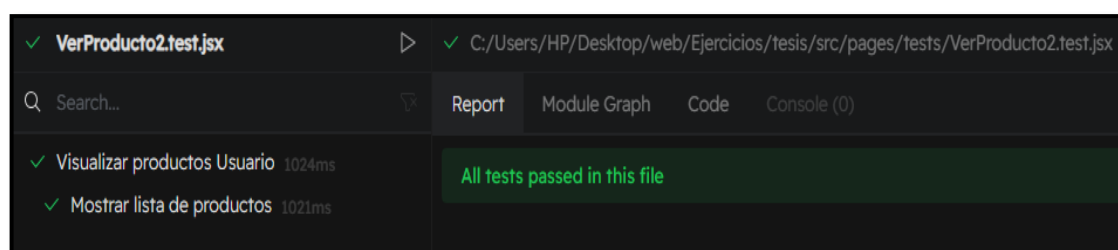


Figura 3.23 Test unitario - examinar productos.

Aplicación de endpoints para llevar a cabo la operación de gestión del carrito de compras.

En el frontend, aquellos miembros identificados como cliente pueden gestionar su carrito de compras. Abarcando actividades tales como visualizar, agregar y editar la cantidad o el producto, utilizando para ello el consumo de endpoints que han sido generados por el backend. La **Figura 3.24** muestra la parte visual de la interfaz de usuario vinculada a la funcionalidad, por otra parte, se detalla el resultado de la prueba unitaria en la **Figura 3.25**, también, se puede revisar el **ANEXO III** para obtener una comprensión más profunda sobre esta tarea.

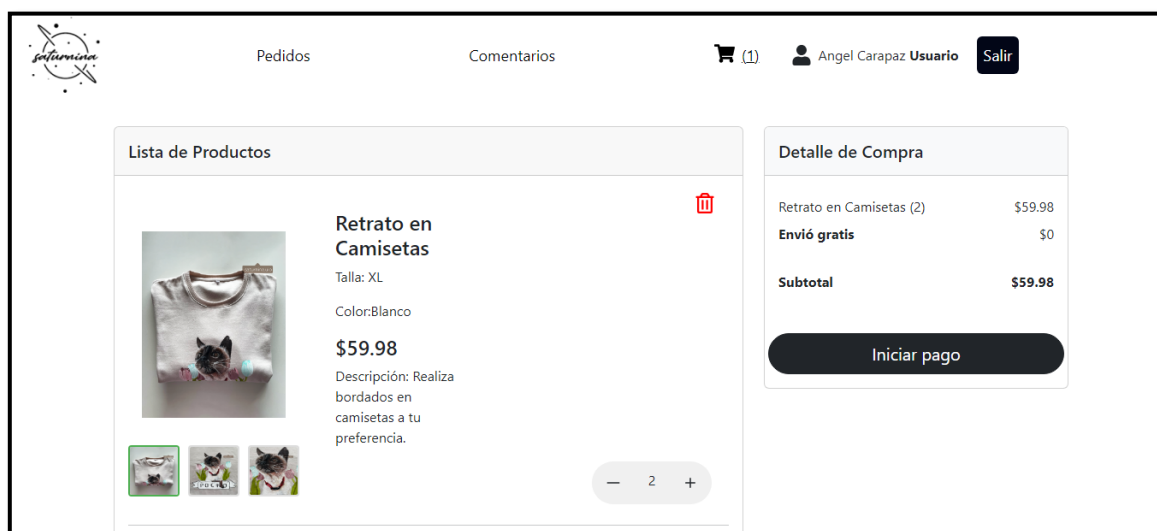


Figura 3.24 Gestión de productos al carrito de compras.

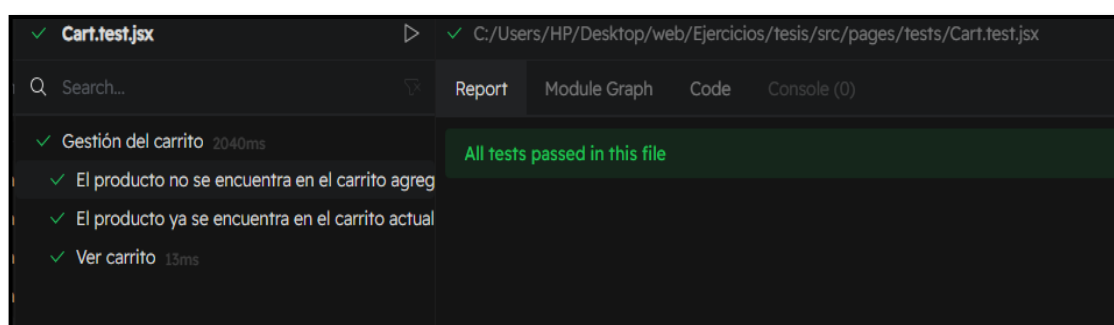


Figura 3.25 Test unitario – gestión de productos al carrito de compras.

Aplicación de endpoints para realizar la operación de crear/actualizar comentarios

En el frontend, los miembros clasificados como cliente pueden enviar comentarios. Abarcando actividades tales como realizar comentarios, mismas que el administrador puede revisar y tomar medidas según lo que sea más adecuado. Además, estas acciones se llevan a cabo mediante los endpoints que han sido generados por el backend. La **Figura 3.26** muestra la parte visual de la interfaz de usuario vinculada a la funcionalidad, además, se detalla el resultado de la prueba unitaria en la **Figura 3.27**, por otra parte, se puede consultar el **ANEXO III** para obtener una comprensión más profunda sobre estas tareas.



Figura 3.26 Crear/Actualizar comentario.

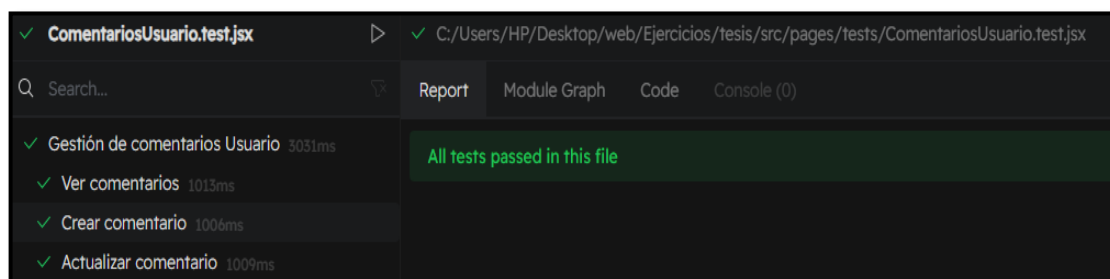


Figura 3.27 Test unitario – Crear/Actualizar comentario.

Sprint 3. Pruebas dirigidas al desarrollo del frontend.

En este Sprint, se describen las tareas que han sido determinadas en la planificación del Sprint Backlog, las cuales son.

- Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones unitarias.
- Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones de compatibilidad.
- Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones de aceptación.
- Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones de rendimiento.

Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones unitarias

Estas pruebas representan un procedimiento destinado a verificar la operatividad adecuada de cada componente de manera individual. Estas evaluaciones posibilitan la revisión del código, asegurando su fiabilidad. Además, al llevar a cabo las pruebas unitarias, se logra la detección de posibles fallos, garantizando que los componentes se comporten conforme a las expectativas y corrigiendo potenciales vulnerabilidades o errores en el código. Este proceso contribuye a fortalecer la solidez del sistema [37].

A continuación, a través de la **Figura 3.28** se muestra el código que se han empleado en una prueba unitaria, mientras que la **Figura 3.29**, exhibe el resultado que se ha obtenido de la mencionada prueba. Asimismo, se puede consultar el **ANEXO II** para un diagnóstico detallado de las evaluaciones restantes.

```

21 it('Inicio de sesión exitoso', async () => {
22   render(
23     <TestWrapper>
24       <Login />
25     </TestWrapper>
26   );
27
28   const { auth, setAuth } = useTestContext();
29
30   userEvent.type(screen.getByLabelText(/correo electrónico/i), 'miguelotaku01@gmail.com');
31   userEvent.type(screen.getByLabelText(/contraseña/i), 'Miguel1#c');
32
33   userEvent.click(screen.getByText(/iniciar sesión/i));
34
35   await waitFor(() => {
36     expect(screen.getByText(/iniciando sesión.../i)).toBeInTheDocument();
37   });
38 }

```

Figura 3.28 Ejemplo de código de prueba - inicio de sesión.

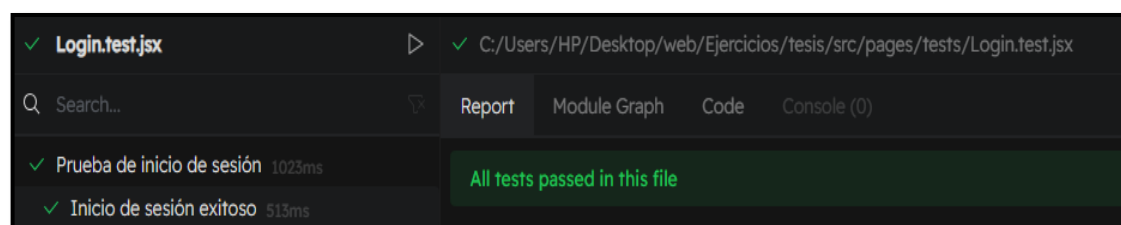


Figura 3.29 Resultado de prueba unitaria - iniciar sesión.

Tras llevar a cabo las pruebas unitarias, se evidencia que la operación de cada uno de los componentes es óptima, ya que no se han detectado inconvenientes.

Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones de compatibilidad

Al realizar estas evaluaciones, se debe enfocar en analizar la visualización de la información en diversos ambientes y dispositivos. Este procedimiento implica comprobar si el contenido multimedia, así como la distribución de contenido es apto para diferentes navegadores web, pantallas de dispositivos móviles, etc. [37] En este contexto específico, se ha procedido a realizar la evaluación en diferentes exploradores web como se puede apreciar en la **Tabla 3.1** y para obtener el resultado visual de esta evaluación se puede remitir al **ANEXO II**.

Tabla 3.1 Prueba de compatibilidad.

EXPLORADOR WEB	EDICIÓN	NOTA
Brave	120.1.61.120	Funcionando sin inconvenientes
Google Chrome	120.0.6099.225	Funcionando sin inconvenientes
Microsoft Edge	120.0.2210.133	Funcionando sin inconvenientes

Tras llevar a cabo las pruebas de compatibilidad, se constata que el funcionamiento adecuado de todos los módulos se ha verificado en diversos navegadores. Esta evaluación indica que las implementaciones han sido efectivas, alcanzando un desempeño apropiado de los componentes en distintos exploradores web.

Ejecutar y ofrecer información detallada sobre los resultados que se han logrado en las evaluaciones de aceptación

Al realizar estas evaluaciones, se debe enfocar en verificar si el sistema software cumple con los criterios y estándares definidos por el dueño del producto final [37]. En consecuencia, la **Tabla 3.2** presenta un ejemplar de la prueba de aceptación basada en los criterios que se han establecido por parte del usuario y en el **ANEXO II** se suministra información detallada sobre todas las evaluaciones restantes.

Tabla 3.2 Ejemplo para la prueba de aceptación.

EJEMPLO PRUEBA DE ACEPTACIÓN	
Identificador de la prueba: PA002	historia de Usuario: HU002
Nombre de la prueba: Iniciar Sesión, Cerrar Sesión, Restablecer Contraseña	
Explicación de la prueba: El administrador y cliente en el frontend deben utilizar diversos endpoints para: <ul style="list-style-type: none"> • Iniciar sesión • Finalizar sesión • Restablecer contraseña 	
Fases de realización:	

<p>Para iniciar sesión todos los usuarios:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • En lado derecho superior dar clic en Iniciar sesión. • Llenar los campos del correo electrónico y contraseña dependiendo del rol de usuario. • Dar clic en “Iniciar Sesión”. <p>Para salir de la sesión los usuarios:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En el lado derecho superior dar clic en “Cerrar sesión” <p>Para modificar contraseña:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • En lado derecho superior dar clic en “Iniciar Sesión”. • Dar clic en “Recupera tu cuenta”. • Llenar el formulario con tu correo electrónico. • Confirmar el correo enviado a tu correo. • Llenar el formulario respectivo para el cambio de contraseña. • Dar clic en “Cambiar contraseña”
<p>Resultado deseado:</p> <p>El componente frontend facilita el proceso de ingresar y salir de la sesión, así como la posibilidad de recuperar o modificar la contraseña con todos los usuarios.</p>
<p>Análisis de la prueba:</p> <p>El resultado cumple con las expectativas y el cliente expresa su plena satisfacción al 100%.</p>

Luego de llevar a cabo estas pruebas de aceptación, se ha verificado que cada módulo cumple con la finalidad deseada en base a los requisitos que se han recopilado al inicio del proyecto. Asimismo, todas las pruebas han recibido la aprobación del Product Owner.

Ejecutar y ofrecer información sobre los resultados que se han logrado en las evaluaciones de rendimiento

Estas evaluaciones consisten en realizar el análisis de rendimiento que tiene un sistema software bajo circunstancias específicas de operación [37]. En este sentido, se ha ejecutado la evaluación de rendimiento mediante la herramienta Lighthouse como se ilustra en la **figura 3.30** en donde se exhibe el resultado que se ha logrado.

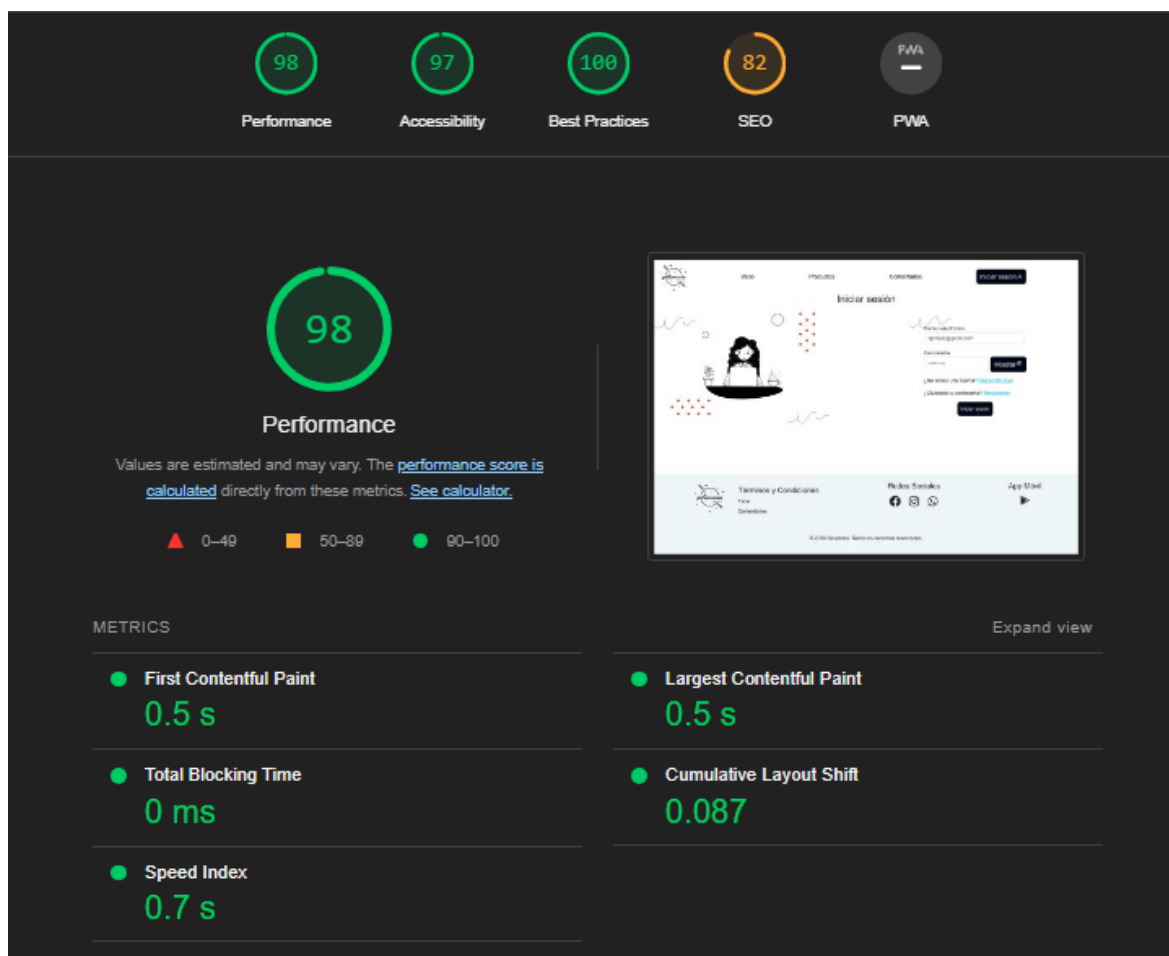


Figura 3.30 Evaluación – Lighthouse.

Tras realizar la mencionada evaluación, se verifica que todos los módulos funcionan con un rendimiento apropiado. El presente informe refleja que los ajustes que se han implementado han resultado favorables.

Sprint 4. Ejecución del componente frontend en un entorno de producción.

A continuación, se expone la única tarea asignada al Sprint 4. Una vez completada la etapa de codificación, pruebas y la aprobación por parte del propietario del producto, se procede a llevar a cabo la implementación del frontend a un entorno de producción utilizando la plataforma Vercel. Dicha plataforma se especializa en el desarrollo y alojamiento web, ofreciendo diversos servicios para facilitar la creación, implementación y gestión de sitios web y aplicaciones.

Por último, para acceder al componente frontend, se proporciona la URL correspondiente:

<https://saturnina.vercel.app/>

Finalmente, el administrador de "Saturnina" ha desarrollado un documento que evidencia el cumplimiento de todos los requisitos y tareas de la interfaz de usuario, según como se ha definido al inicio del proyecto. En el **ANEXO II** se proporciona el documento en cuestión.

4 CONCLUSIONES

A continuación, se lista todas las conclusiones extraídas en la ejecución de este Proyecto de Integración Curricular.

- Se ha cumplido satisfactoriamente los objetivos y el alcance que se ha definido en este proyecto, de este modo el administrador y los clientes pueden realizar distintas tareas en la interfaz de usuario, dependiendo de su rol. Garantizando de esta manera, una gestión efectiva mediante la integración de la tecnología.
- Gracias a la definición de requisitos al inicio del proyecto ha permitido la selección apropiada de artefactos, la planificación de cada uno de los sprints y la identificación de las herramientas necesarias para las diferentes fases de codificación, pruebas y despliegue.
- La implementación del modelo MVC, que se caracteriza por la asignación clara de responsabilidades, ha permitido un desarrollo estructural en cada una de las interfaces de usuario.
- La vinculación efectiva del frontend con el backend, mediante la utilización de APIs, ha facilitado un intercambio de información fluido, facilitando la transferencia y recepción de información en los módulos accesibles para cada usuario.
- Gracias a una amplia gama de componentes visuales listos para su implementación, han agilizado considerablemente el proceso de desarrollo del frontend, optimizando la eficiencia de cada una de las interfaces de usuario y mejorando la satisfacción general del usuario.
- Gracias a un conjunto de pruebas que se han realizado en el frontend, se ha verificado la correcta operación de cada módulo, lo que implica una buena distribución del contenido, buenas prácticas de codificación y la aprobación del producto final.

5 RECOMENDACIONES

A continuación, se lista todas las recomendaciones extraídas en la ejecución de este Proyecto de Integración Curricular.

- Se sugiere mantener un enfoque proactivo en la optimización del frontend, utilizando para ello auditorías periódicas para identificar posibles mejoras y garantizar una experiencia de usuario fluida.
- Se recomienda integrar un sistema de comunicación en tiempo real con el fin de evaluar la situación de manera inmediata y recibir comentarios de los usuarios mientras interactúan con la plataforma. Esto permitirá realizar ajustes rápidos según las necesidades y preferencias de los usuarios, mejorando así la calidad del servicio.
- Se sugiere mantenerse al tanto de las tendencias tecnológicas emergentes en el contexto del comercio electrónico y con ello evaluar la viabilidad de integrar nuevas tecnologías que puedan ofrecer ventajas competitivas, como la realidad aumentada para la visualización de productos o métodos innovadores de pago.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Rivadeneira, «Saturnina,» Instagram, 15 Septiembre 2020. Disponible en: <https://www.instagram.com/saturnina.uio/>. [Accedido: 08-nov-2023].
- [2] A. Orbis, «1. Importancia del ecommerce para las empresas», Agencia de Marketing Digital, 26-ago-2021. Disponible en: <https://www.somosorbis.com/blog/importancia-del-ecommerce/>. [Accedido: 16-oct-2023].
- [3] «¿Qué es el front end y por qué es tan importante para un sitio web?», Naxer, 22-jul-2021. Disponible en: <https://www.naxer.es/noticias/que-es-el-frontend/>. [Accedido: 16-oct-2023].
- [4] «Figma», Figma. Disponible en: <https://www.figma.com/developers/api>. [Accedido: 16-oct-2023].
- [5] «JavaScript Guide», Mozilla.org. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>. [Accedido: 16-oct-2023].
- [6] «Getting started», Reactjs.org. Disponible en: <https://legacy.reactjs.org/docs/getting-started.html>. [Accedido: 17-oct-2023].
- [7] «Documentación», Node.js. Disponible en: <https://nodejs.org/es/docs>. [Accedido: 17-oct-2023].
- [8] «Installation - Tailwind CSS», Tailwindcss.com. Disponible en: <https://tailwindcss.com/docs/installation>. [Accedido: 17-oct-2023].
- [9] «Vite», Vitejs.dev. Disponible en: <https://vitejs.dev/guide/>. [Accedido: 17-oct-2023].
- [10] T. Fredrich y P. eCollege, «REST API Tutorial», Restapitutorial.com. Disponible en: <https://www.restapitutorial.com/>. [Accedido: 17-oct-2023].
- [11] Software DELSOL, «Metodología», Sdelsol.com, 01-abr-2019. Disponible en: <https://www.sdelsol.com/glosario/metodologia/>. [Accedido: 20-nov-2023].

- [12] S. G. Sotomayor, «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa», Thinking for Innovation, 2023.
- [13] J. H. Canós y M. C. P. Letelier, «Metodologías Ágiles en el Desarrollo de Software», 2012.
- [14] «Qué es SCRUM», Proyectos Ágiles, 04-ago-2008. Disponible en: <https://proyectosagiles.org/que-es-scrum/>. [Accedido: 18-nov-2023].
- [15] M. Á. De Rios, «Scrum: qué es y cómo funciona este marco de trabajo,» Global Growth Agents, 9 Mayo 2022. Disponible en: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>. [Accedido: 18-nov-2023].
- [16] S. Monroy, «¿Cuáles son los roles de la metodología Scrum?,» APD, 14 Diciembre 2021. Disponible en: <https://www.apd.es/roles-metodologia-scrum/>. [Accedido: 18-nov-2023].
- [17] «Artefactos Scrum», <https://miro.com/>. Disponible en: <https://miro.com/es/agile/que-son-artefactos-scrum/>. [Accedido: 18-nov-2023].
- [18] «Qué es la recopilación de requisitos: definición y herramientas», Visure Solutions, 10-jul-2022. Disponible en: <https://visuresolutions.com/es/blog/requirements-gathering/>. [Accedido: 19-nov-2023].
- [19] D. M. Rehkopf, «Historias de usuario», Atlassian. Disponible en: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Accedido: 19-nov-2023].
- [20] A. Raeburn, «Qué es el product backlog y guía para hacer uno con ejemplo [2022] •», Asana, 25-ago-2022. Disponible en: <https://asana.com/es/resources/product-backlog>. [Accedido: 20-nov-2023].
- [21] Asana, «Backlog: qué es el trabajo pendiente del sprint y ejemplos», Asana, 17-ago-2022. Disponible en: <https://asana.com/es/resources/sprint-backlog>. [Accedido: 20-nov-2023].

[22] P. Segui, «Interfaz de usuario: Qué es y cómo diseñar una UI», OVACEN, 16-dic-2022. Disponible en: <https://ovacen.com/disenio-interfaz-usuario-ui/>. [Accedido: 20-nov-2023].

[23] «¿Qué es figma?», CEI: Escuela de Diseño y Marketing. Disponible en: <https://cei.es/que-es-figma/>. [Accedido: 20-nov-2023].

[24] Wordpress.com. Disponible en: <https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>. [Accedido: 20-nov-2023].

[25] «MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web», MDN Web Docs. Disponible en: <https://developer.mozilla.org/es/docs/Glossary/MVC>. [Accedido: 20-nov-2023].

[26] P. Huet, «Arquitectura de software: Qué es y qué tipos existen», Openwebinars.net, 24-ago-2022. Disponible en: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/#:~:text=Qu%C3%A9%20es%20la%20arquitectura%20de%20software,-Todo%20el%20mundo&text=La%20arquitectura%2C%20referida%20al%20software,paso%20previo%20a%20cualquier%20implementa>. [Accedido: 20-nov-2023].

[27] B. Gustavo, «Qué es GitHub y cómo empezar a usarlo», Tutoriales Hostinger, 12-abr-2019. Disponible en: <https://www.hostinger.es/tutoriales/que-es-github>. [Accedido: 20-nov-2023].

[28] A. Deyimar, «Qué es React: definición, características y funcionamiento», Tutoriales Hostinger, 04-feb-2020. Disponible en: <https://www.hostinger.es/tutoriales/que-es-react>. [Accedido: 20-nov-2023].

[29] «Vite», Vitejs.dev. Disponible en: <https://es.vitejs.dev/guide/>. [Accedido: 20-nov-2023].

[30] «Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo», Rock Content - ES, 12-abr-2020. Disponible en: <https://rockcontent.com/es/blog/bootstrap/>. [Accedido: 20-nov-2023].

[31] «Las librerías y los frameworks JavaScript más populares», IONOS Digital Guide, 01-mar-2023. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/frameworks-javascript-y-librerias-populares/>. [Accedido: 20-nov-2023].

[32] «Usando Objetos FormData», MDN Web Docs. Disponible en: https://developer.mozilla.org/es/docs/Web/API/XMLHttpRequest_API/Using_FormData_Objects. [Accedido: 20-nov-2023].

[33] «libreria JWT DECODE», Stack Overflow en español. Disponible en: <https://es.stackoverflow.com/questions/126360/libreria-jwt-decode>. [Accedido: 20-nov-2023].

[34] «Uso con React · Redux en Español», Redux.js.org. Disponible en: <https://es.redux.js.org/docs/basico/uso-con-react.html>. [Accedido: 20-nov-2023].

[35] «Npm: React-loading-skeleton», npm. Disponible en: <https://www.npmjs.com/package/react-loading-skeleton>. [Accedido: 20-nov-2023].

[36] F. G. de Zúñiga, «Axios Javascript: analizamos las características de este ligero cliente HTTP», Blog de arsys.es, 05-abr-2019. Disponible en: <https://www.arsys.es/blog/axios>. [Accedido: 20-nov-2023].

[37] A. Jacinto, «Optimización del desarrollo web con pruebas automatizadas de front-end», Startechup Inc, 30-jun-2023. Disponible en: <https://www.startechup.com/es/blog/4-front-end-automated-testing-tools/>. [Accedido: 09-dic-2023].

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo del frontend, los cuales se encuentran detallados de la siguiente manera:

- **ANEXO I.** Resultado del programa anti plagio Turnitin.
- **ANEXO II.** Manual de Usuario.
- **ANEXO III.** Manual de Instalación.
- **ANEXO IV.** Credenciales de acceso y despliegue.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.



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

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 18 de febrero de 2024

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de un frontend asociado al DESARROLLO DE UN SISTEMA WEB y APP DE E-COMMERCE PARA LA TIENDA SATURNINA elaborado por el estudiante MIGUEL ÁNGEL CARAPAZ MENDOZA 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 secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 02%.

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,



Firmado electrónicamente por:
BYRON GUSTAVO
LOARTE CAJAMARCA

**Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos**

ANEXO II

Recopilación de requerimientos

Todos los requisitos que se han establecido previamente al desarrollo del proyecto se encuentran registrados en la **Tabla I**.

Tabla I Recopilación de requerimientos.

RECOPIACION DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
Frontend	RR-001	Como usuario cliente necesita consumir varios endpoints para: <ul style="list-style-type: none"> • Registrarse
	RR-002	Como usuario administrador y cliente necesitan consumir varios endpoints para: <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión • Recuperar contraseña
	RR-003	Como usuario administrador y cliente necesitan consumir varios endpoints para: <ul style="list-style-type: none"> • Modificar perfil
	RR-004	Como usuario administrador necesita consumir varios endpoints para: <ul style="list-style-type: none"> • Gestionar categorías.
	RR-007	Como usuario administrador necesita consumir varios endpoints para: <ul style="list-style-type: none"> • Gestionar comentarios.
	RR-008	Como usuario cliente necesita consumir endpoints para: <ul style="list-style-type: none"> • Visualizar categorías
	RR-009	Como usuario cliente necesita consumir endpoints para: <ul style="list-style-type: none"> • Visualizar productos.

	RR-010	Como usuario cliente necesita consumir endpoints para: <ul style="list-style-type: none"> • Gestionar carrito de compras.
	RR-011	Como usuario cliente necesita consumir endpoints para: <ul style="list-style-type: none"> • Visualizar pedidos.
	RR-012	Como usuario cliente necesita consumir endpoints para: <ul style="list-style-type: none"> • Calificar el producto.

Historias de usuario

A continuación, se detalla la creación de las Historias de Usuario para este componente después de haber concluido la fase de Recopilación de Requerimientos. En total, se han generado 12 Historias de Usuario, que se detallan desde la **Tabla II** hasta **Tabla XII**.

Tabla II Historia – Registrarse.

HISTORIA DE USUARIO	
Clave: HU001	Rol: Cliente
Historia: Registrarse	
Prioridad: Alta	Riesgo en desarrollo: Baja
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	
Descripción: El usuario cliente en el frontend debe consumir varios endpoints para: <ul style="list-style-type: none"> • Registrarse. 	
Observación: El usuario cliente necesita registrarse para acceder al sistema. Por otro lado, el usuario administrador solo recibirá credenciales creadas por el equipo de TI para poder iniciar sesión.	

Tabla III Historia - Iniciar Sesión, Cerrar Sesión, Recuperar Contraseña.

HISTORIA DE USUARIO	
Clave: HU002	Rol: Administrador y Cliente
Historia: Iniciar Sesión, Cerrar Sesión, Recuperar Contraseña	
Prioridad: Alta	Riesgo en desarrollo: Baja
Asignación de iteración: 3	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario administrador y cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión • Recuperar contraseña 	
<p>Observación: Iniciar sesión es necesario para acceder a las funciones de administrador y cliente.</p>	

Tabla IV Historia - Modificar Perfil.

HISTORIA DE USUARIO	
Clave: HU003	Rol: Administrador y Cliente
Historia: Modificar Perfil	
Prioridad: Media	Riesgo en desarrollo: Media
Asignación de iteración: 3	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario administrador y cliente en el frontend debe consumir varios endpoints para visualizar y editar su perfil en base a los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre. 	

<ul style="list-style-type: none"> • Apellido. • Celular. • Correo.
<p>Observación: El usuario administrador y cliente accederán a dicha vista una vez hayan iniciado sesión para ver o editar su perfil a través de un formulario con los campos ya mencionados anteriormente. Por otra parte, el único campo que no se podrá editar será el correo.</p>

Tabla V Historia - Gestionar Categorías.

HISTORIA DE USUARIO	
Clave: HU004	Rol: Administrador
Historia: Gestionar Categorías	
Prioridad: Alta	Riesgo en desarrollo: Media
Asignación de iteración: 2	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Registrar categorías. • Modificar categorías. • Eliminar categorías. • Visualizar las categorías. 	
<p>Observación: El usuario administrador accede a dichas vistas para gestionar las categorías una vez que ha iniciado sesión.</p>	

Tabla VI Historia - Gestionar Pedidos.

HISTORIA DE USUARIO	
Clave: HU006	Rol: Administrador
Historia: Gestionar Pedidos	
Prioridad: Alta	Riesgo en desarrollo: Media
Asignación de iteración: 2	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Aceptar pedidos. • Listar pedidos. • Cambiar el estado del pedido. 	
<p>Observación: El usuario administrador accede a dichas vistas para gestionar los pedidos una vez que ha iniciado sesión.</p>	

Tabla VII Historia – Gestionar comentarios.

HISTORIA DE USUARIO	
Clave: HU007	Rol: Administrador
Historia: Gestionar comentarios	
Prioridad: Media	Riesgo en desarrollo: Media
Asignación de iteración: 2	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar comentarios y/o sugerencias. 	

<ul style="list-style-type: none"> • Eliminar comentarios y/o sugerencias.
<p>Observación: El usuario administrador accede a dichas vistas para gestionar los comentarios y/o sugerencias una vez que ha iniciado sesión. Además, esto se podrá realizar en el apartado de cada producto.</p>

Tabla VIII Historia - Visualizar Categorías.

HISTORIA DE USUARIO	
Clave: HU008	Rol: Cliente
Historia: Visualizar Categorías	
Prioridad: Media	Riesgo en desarrollo: Baja
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar las categorías. 	
<p>Observación: El usuario cliente podrá visualizar las categorías incluso sin iniciar sesión en el sistema y estas aparecerán de forma dinámica cuando el usuario administrador cree una nueva categoría.</p>	

Tabla IX Historia - Visualizar Productos.

HISTORIA DE USUARIO	
Clave: HU009	Rol: Cliente
Historia: Visualizar Productos	
Prioridad: Alta	Riesgo en desarrollo: Baja
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	

Descripción: El usuario cliente en el frontend debe consumir varios endpoints para visualizar los productos en cada producto se puede encontrar la siguiente información:

- Nombre del producto.
- Precio del producto.
- Descripción del producto.
- Imágenes del producto.
- Talla del producto.
- Color del producto.
- Categoría del producto.

Observación: El usuario cliente podrá visualizar los productos incluso sin iniciar sesión en el sistema y estas aparecerán de forma dinámica cuando el usuario administrador cree un nuevo producto.

Tabla X Historia - Gestionar Carrito de Compras.

HISTORIA DE USUARIO	
Clave: HU010	Rol: Cliente
Historia: Gestionar Carrito de Compras	
Prioridad: Alta	Riesgo en desarrollo: Alta
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Agregar productos al carrito de compras. • Visualizar productos agregados al carrito de compra. • Eliminar productos del carrito de compras. • Incrementar o disminuir la cantidad del producto. 	

<ul style="list-style-type: none"> • Visualizar el precio de cada producto y el valor total a pagar. • Realizar la compra mediante transferencia o deposito.
<p>Observación: El usuario cliente accede a dichas vistas para gestionar el carrito de compras una vez que ha iniciado sesión.</p>

Tabla XI Historia - Visualizar Pedidos.

HISTORIA DE USUARIO	
Clave: HU011	Rol: Cliente
Historia: Visualizar Pedidos	
Prioridad: Alta	Riesgo en desarrollo: Media
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	
<p>Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar los pedidos realizados. • Visualizar el estado actual de su pedido. 	
<p>Observación: El usuario cliente accede a dichas vistas para visualizar los pedidos una vez que ha iniciado sesión.</p>	

Tabla XII Historia - Calificar el Producto

HISTORIA DE USUARIO	
Clave: HU012	Rol: Cliente
Historia: Calificar el Producto	
Prioridad: Media	Riesgo en desarrollo: Media
Asignación de iteración: 1	
Encargado (s): Miguel Carapaz	

Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:

- Visualizar todos los comentarios y/o sugerencias de los productos.
- Crear un comentario y/o sugerencia del producto.
- Actualizar el comentario y/o sugerencia del producto.
- Calificar al producto.

Observación: El usuario cliente para poder calificar y comentar debe haber iniciado sesión. Por otra parte, para poder visualizar todos los comentarios lo podrá hacer sin necesidad de iniciar sesión.

Product Backlog

Se proporciona la descripción integral de todo el Product Backlog y se muestra la priorización asignada a cada Historia de Usuario, considerando su relevancia para el propietario del producto y su complejidad en el proceso de desarrollo como se presenta en la **Tabla XIII**.

Tabla XIII Elaboración del Product Backlog.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID-HU	HISTORIA	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Registrarse	1	Finalizado	Alta
HU002	Iniciar Sesión, Cerrar Sesión, Recuperar Contraseña	3	Finalizado	Alta
HU003	Modificar Perfil	3	Finalizado	Alta
HU004	Gestionar Categorías	2	Finalizado	Alta
HU007	Gestionar comentarios	2	Finalizado	Alta

HU008	Visualizar Categorías	1	Finalizado	Media
HU009	Visualizar Productos	1	Finalizado	Alta
HU010	Gestionar Carrito de Compras	1	Finalizado	Alta
HU011	Visualizar Pedidos	1	Finalizado	Alta
HU012	Calificar el producto	1	Finalizado	Media

Sprint Backlog

En la **Tabla XIV**. Se detallan los Sprints que se han desarrollado por el frontend, incluyendo una descripción de las actividades que se han realizado y el tiempo que se ha establecido para su cumplimiento, según lo establecido por el dueño del producto.

Tabla XIV Elaboración del Sprint Backlog.

ELABORACION DEL SPRINT BACKLOG						
ID-SB	NOMBRE	MÓDULO	HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB-001	Diseño y desarrollo de los módulos para el usuario cliente	Módulo de registro	HU001	Registrarse	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para registrarse • Validación de los campos requeridos 	80 H
		Módulo de Categorías	HU008	Visualizar Categorías	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar las categorías 	

		Módulo de Productos	HU009	Visualizar Productos	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para visualizar los productos • Diseño e implementación de las interfaces para visualizar el producto seleccionado 	
		Módulo de gestión de carrito de compras	HU010	Gestionar Carrito de Compras	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para visualizar los productos agregados al carrito de compras • Diseño e implementación de las interfaces para realizar los pedidos 	
		Módulo de Pedidos	HU011	Visualizar Pedidos	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para poder visualizar los pedidos realizados • Visualización del estado del pedido 	
		Módulo de Comentarios y/o Sugerencias	HU012	Calificar el Producto	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para poder visualizar, comentar los productos • Diseño e implementación de la interfaz para poder calificar los productos 	

SB-002	Diseño y desarrollo de los módulos para el usuario Administrador	Módulo de Gestión de Categorías	HU004	Gestionar Categorías	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para poder gestionar las categorías • Validación de los campos requeridos 	80 H
		Módulo de Gestión de Pedidos	HU006	Gestionar Pedidos	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para poder gestionar los pedidos • Implementación de estados para los pedidos 	
		Módulo de Gestión de Comentarios y/o sugerencias	HU007	Gestionar Calificación del Producto	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para poder gestionar los comentarios y/o sugerencias 	
SB-003	Diseño y desarrollo de los módulos para el	Módulo de Perfil	HU003	Modificar Perfil	<ul style="list-style-type: none"> • Diseño e implementación de la interfaz para visualizar y modificar el perfil • Validación de los campos requeridos 	20 H

	usuario Administrador y Cliente	Módulo de inicio de sesión	HU002	Iniciar Sesión, Cerrar Sesión, Recuperar Contraseña	<ul style="list-style-type: none"> • Diseño e implementación de las interfaces para el inicio de sesión, cierre de sesión y modificar contraseña • Validación de los campos requeridos 	
SB004	Pruebas del Frontend	<ul style="list-style-type: none"> • Pruebas unitarias. • Pruebas de compatibilidad. • Pruebas de aceptación. • Pruebas de rendimiento. 				20 H
SB005	Despliegue del Frontend	<ul style="list-style-type: none"> • Despliegue del Frontend en Vercel 				10 H
	Documentación	<ul style="list-style-type: none"> • Trabajo de Integración Curricular • Anexos 				30 H
TOTAL						240 H

Diseño de interfaces

A continuación, se presentan los diseños de cada uno de los módulos del frontend, que van desde la **Figura I** a la **Figura XVIII**.

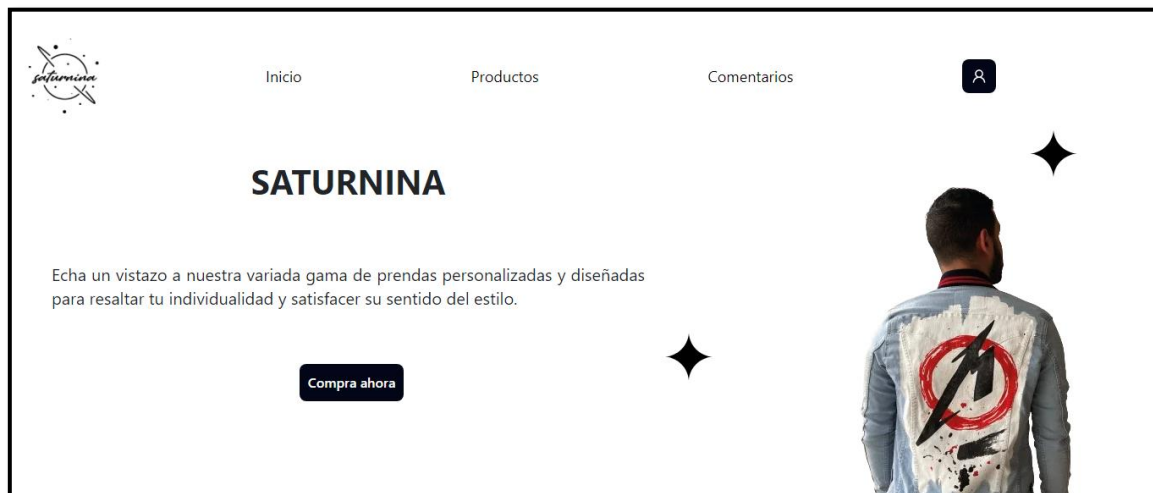


Figura I Pagina Informativa.

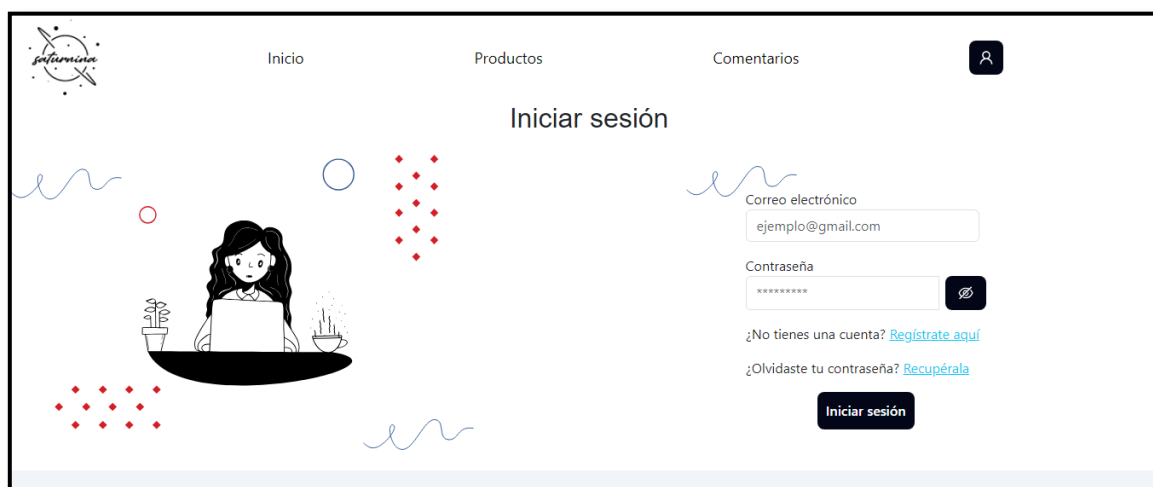


Figura II Inicio de sesión.

Inicio Productos Comentarios

Registrar

Nombre
Ingresa tus nombres completos

Apellido
Ingresa tu apellido

Correo electrónico
ejemplo@gmail.com

Contraseña

La contraseña debe tener entre 9 y 18 caracteres y contener al menos una letra mayúscula, un número y un carácter especial.

Teléfono

Figura III Registro.

Inicio Productos Comentarios

Recuperar Cuenta

Introduzca su correo electrónico para el proceso de recuperación, le enviaremos un correo electrónico.

Correo electrónico
ejemplo@gmail.com

Recuperar Cuenta

Figura IV Recuperar cuenta.

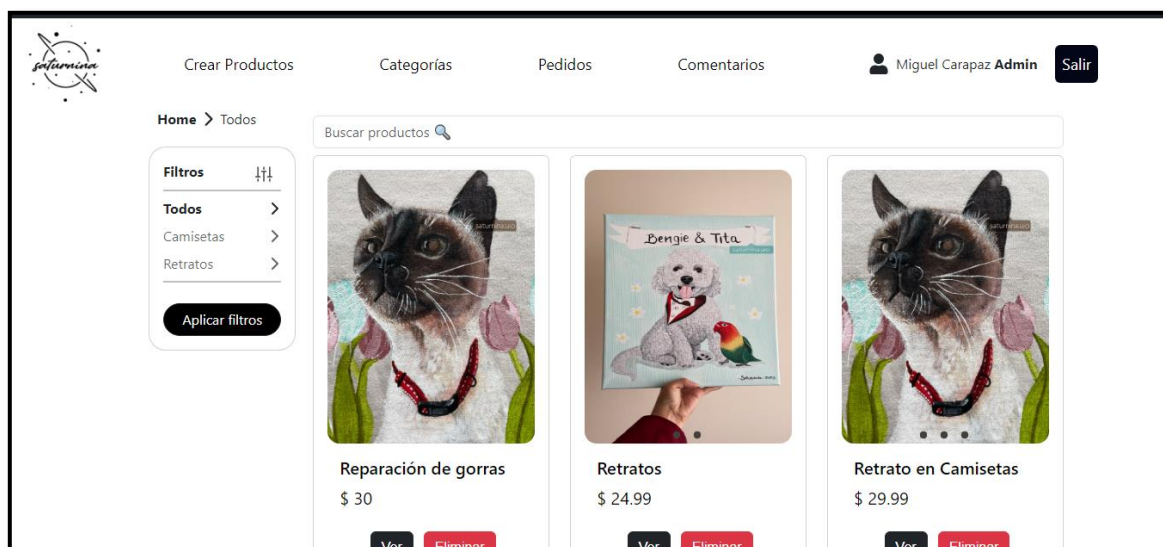


Figura V Administrador – Visualizar productos.



Figura VI Administrador – Gestionar producto.

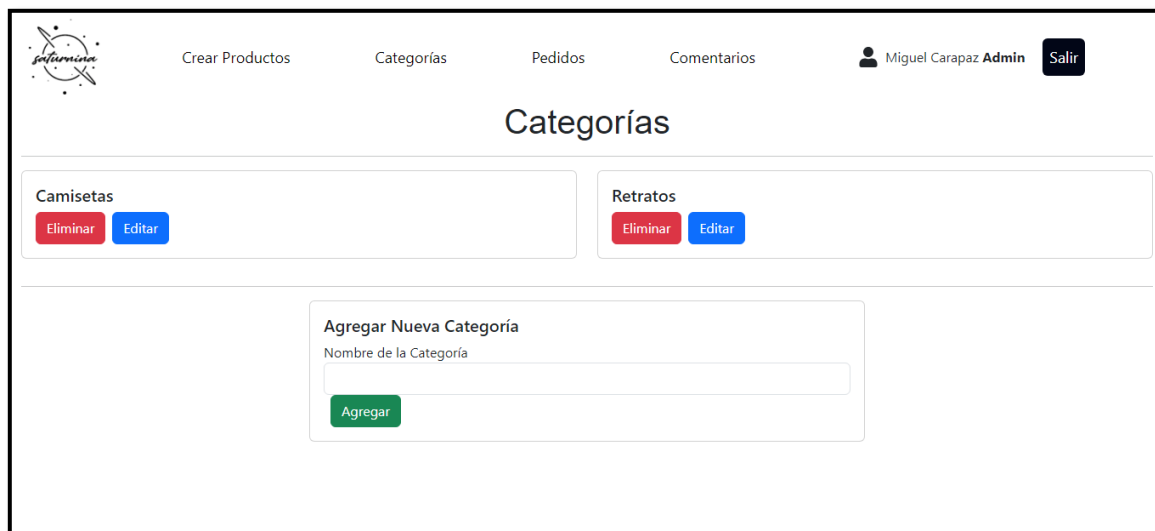


Figura VII Administrador – Gestionar categorías.

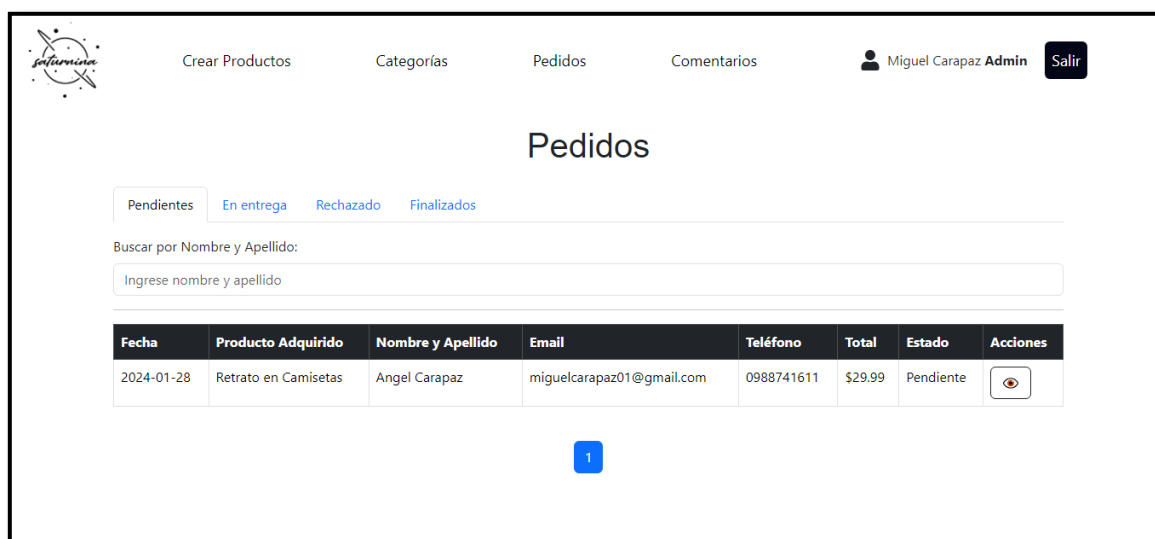


Figura VIII Administrador – Gestionar pedidos.



Figura IX Administrador – Gestionar comentarios y/o sugerencias.

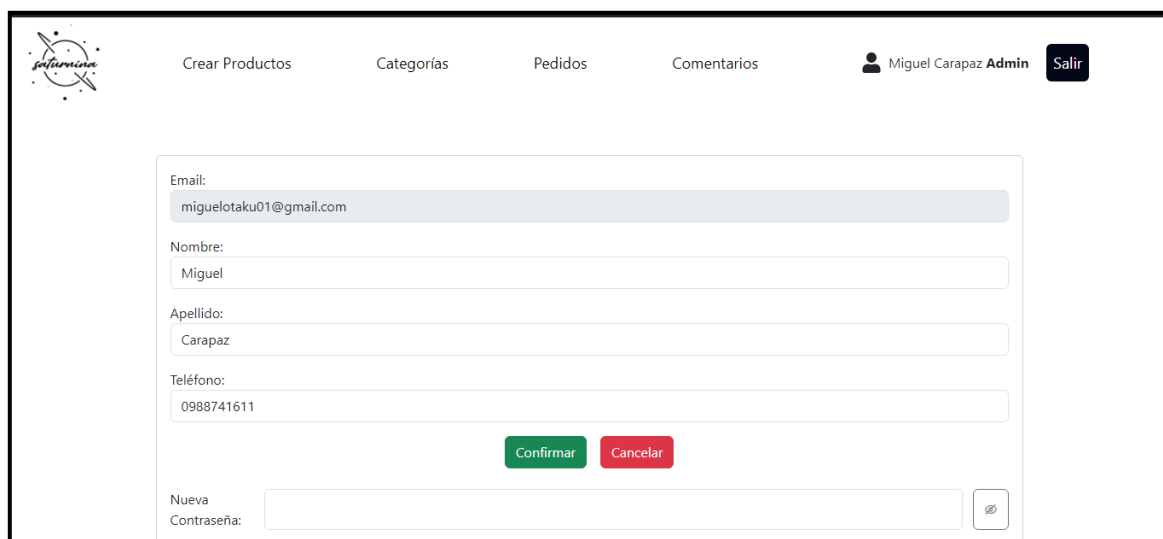


Figura X Administrador – Perfil.



Figura XI Administrador – Detalles del producto.

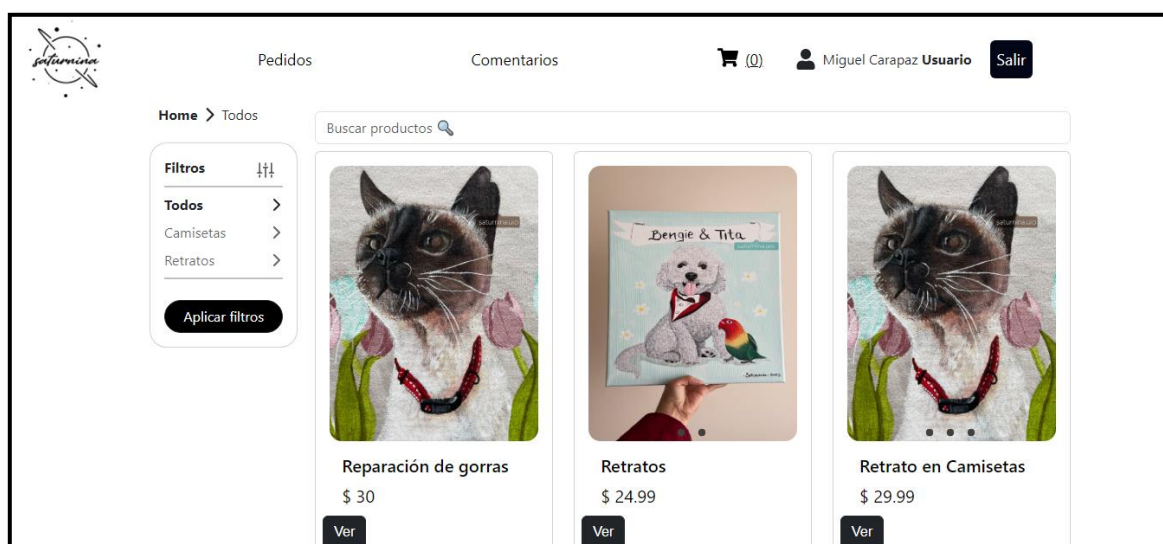


Figura XII Usuario – Visualizar productos.

Pedidos

Comentarios

(0) Miguel Carapaz Usuario Salir

Pedidos

Pendientes En entrega Rechazado Finalizados

Buscar por Producto:

Ingrese nombre del producto

Fecha	Producto Adquirido	Nombre y Apellido	Email	Teléfono	Total	Estado	Acciones
2024-01-28	Retrato en Camisetas	Angel Carapaz	miguelcarapaz01@gmail.com	0988741611	\$29.99	Pendiente	

1

Figura XIII Usuario – Historial de pedidos.

Pedidos

Comentarios

(0) Miguel Carapaz Usuario Salir

Comentarios

Agregar Comentario:

☆☆☆☆☆

Escribe tu comentario...

0/100 caracteres

Actualizar Comentario

←→

☆☆☆☆☆

Miguel Carapaz ✓

Me gusto el servicio brindando

☆☆☆☆☆

David Basantes ✓

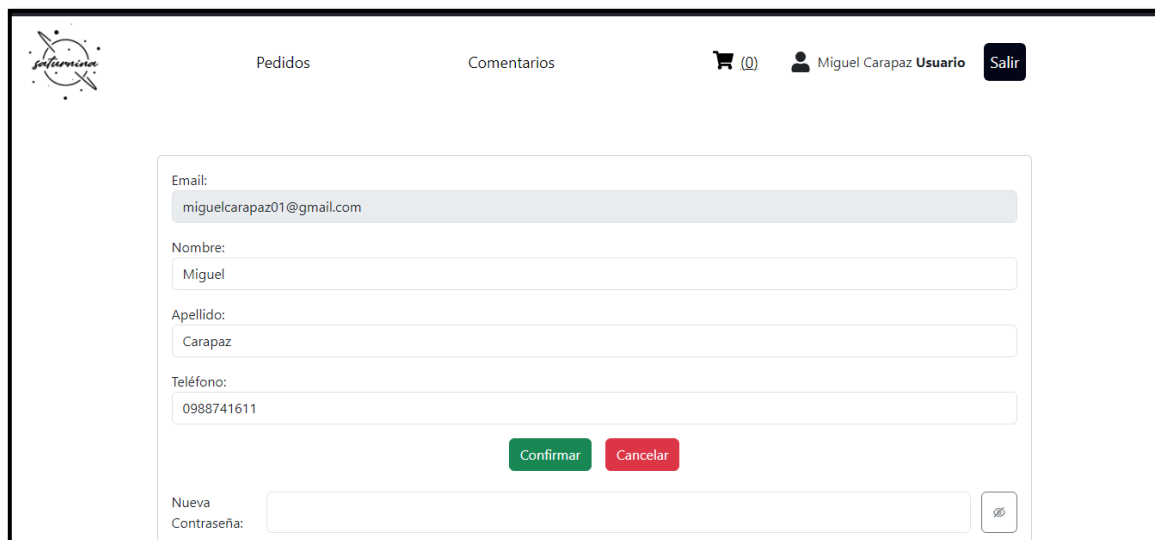
Ame mucho el Producto muchas g racias

☆☆☆☆☆


Jose Pinos ✓

Me gusta Saturnina su personal ización es increíble

Figura XIV Usuario – Enviar comentarios y/o sugerencias.



Logo: *safarinnia*

Pedidos Comentarios  (0) Miguel Carapaz **Usuario** **Salir**

Email: miguelcarapaz01@gmail.com

Nombre: Miguel

Apellido: Carapaz

Teléfono: 0988741611

Confirmar **Cancelar**

Nueva Contraseña:

Figura XV Usuario – Perfil.



Logo: *safarinnia*

Pedidos Comentarios  (1) Miguel Carapaz **Usuario** **Salir**



CAMISETAS
Reparación de gorras
\$30

Reparación de gorras

Comentarios:
No hay comentarios disponibles.

Figura XVI Usuario – Detalles del producto.

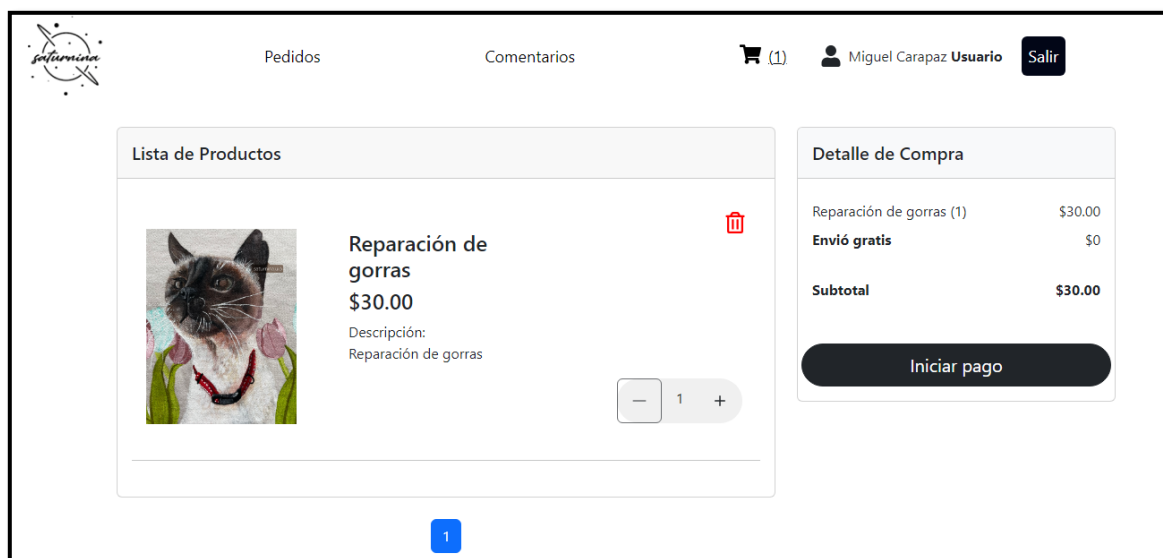


Figura XVII Usuario – Gestión carrito de compras.

The screenshot shows a checkout form titled 'Checkout'. It contains several input fields: 'Nombre:' with 'Miguel', 'Apellido:' with 'Carapaz', 'Correo Electrónico:' with 'miguelcarapaz01@gmail.com', 'Teléfono:' with '0988741611', and 'Dirección:' with the placeholder 'Ingresa tu dirección'. A red error message below the address field reads 'La dirección debe tener al menos 10 caracteres.' To the right, a 'Detalle de Compra' box shows a summary: 'Reparación de gorras (1)' for \$30, 'Envió gratis' for \$0, and a 'Total' of \$30.

Figura XVIII Usuario – Realizar pedido.

Pruebas

Tras finalizar la fase de codificación, se realizan diversas pruebas para validar el adecuado desempeño de las interfaces. Estas pruebas abarcan aspectos como las pruebas unitarias, compatibilidad, aceptación y de rendimiento.

Pruebas unitarias

En la **Figura XIX** se observa el comando en vitest, el cual resulta bastante práctico para realizar pruebas unitarias y en la **Figura XX** se observa todos los test que se han realizado. Además, se ha incorporado el detalle de todas las pruebas que van desde la **Figura XXI** hasta la **Figura LIV**.


```

PS C:\Users\HP\Desktop\web\Ejercicios\tesis> npm test

> saturnina@0.0.0 test
> vitest --ui

DEV v1.2.1 C:/Users/HP/Desktop/web/Ejercicios/tesis
UI started at http://localhost:51204/_vitest_/

✓ src/pages/tests/Categorias.test.jsx (4) 1038ms
✓ src/pages/tests/Perfil.test.jsx (2) 1021ms
✓ src/pages/tests/RecuperarCorreo.test.jsx (3) 1031ms
✓ src/pages/tests/ActualizarEstado.test.jsx (2) 2032ms
✓ src/pages/tests/Cart.test.jsx (3) 2043ms
✓ src/pages/tests/ComentariosUsuario.test.jsx (3) 3033ms
✓ src/pages/tests/Login.test.jsx (2) 1025ms
✓ src/pages/tests/CerrarSesion.test.jsx (2)
✓ src/pages/tests/ActualizarProduct.test.jsx (1) 1020ms
✓ src/pages/tests/Register.test.jsx (2) 3029ms
✓ src/pages/tests/CrearProducto.test.jsx (1) 1021ms
✓ src/pages/tests/ComentariosAdmin.test.jsx (2) 1530ms
✓ src/pages/tests/DetallePedido.test.jsx (2) 2032ms
✓ src/pages/tests/EliminarProducto.test.jsx (1) 517ms
✓ src/pages/tests/VerProducto.test.jsx (1) 1021ms
✓ src/pages/tests/VerProducto2.test.jsx (1) 1025ms
✓ src/pages/tests/ListarPedidos.test.jsx (1) 1014ms

```

Figura XIX Código para realizar el test con vitest.

The screenshot shows the Vitest UI interface. At the top, there is a search bar and a 'Report' tab. Below the search bar, there is a list of test files and their execution results. The status is 'PASS (17)'. The list includes the following files and their execution times:

- src/pages/tests/Categorias.test.jsx 1038ms
- src/pages/tests/Perfil.test.jsx 1021ms
- src/pages/tests/RecuperarCorreo.test.jsx 1031ms
- src/pages/tests/ActualizarEstado.test.jsx 2032ms
- src/pages/tests/Cart.test.jsx 2043ms
- src/pages/tests/ComentariosUsuario.test.jsx 3033ms
- src/pages/tests/Login.test.jsx 1025ms
- src/pages/tests/CerrarSesion.test.jsx 10ms
- src/pages/tests/ActualizarProduct.test.jsx 1020ms
- src/pages/tests/Register.test.jsx 3029ms
- src/pages/tests/CrearProducto.test.jsx 1021ms
- src/pages/tests/ComentariosAdmin.test.jsx 1530ms
- src/pages/tests/DetallePedido.test.jsx 2032ms
- src/pages/tests/EliminarProducto.test.jsx 517ms
- src/pages/tests/VerProducto.test.jsx 1021ms
- src/pages/tests/VerProducto2.test.jsx 1025ms
- src/pages/tests/ListarPedidos.test.jsx 1014ms

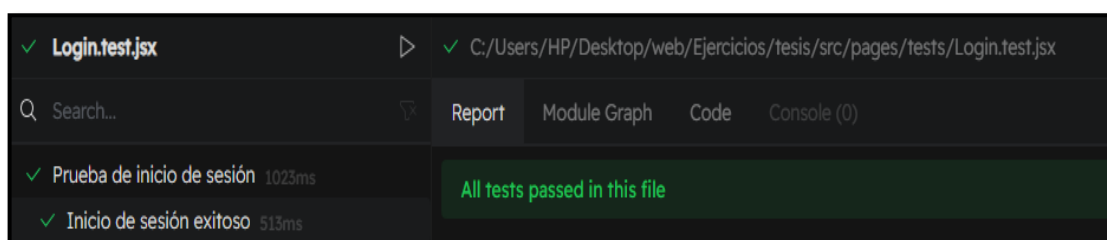
At the bottom right, there is a green box with the text 'All tests passed in this file'.

Figura XX Vitest UI con todas las pruebas.

```
describe('Prueba de inicio de sesión', () => {
  beforeEach(() => {
    globalThis.fetch = async () => ({
      ok: true,
      json: async () => ({
        detail: {
          token: 'fakeAuthToken',
          id: 'fakeUserId',
          rol: 'fakeRole',
          email: 'fake@example.com',
        },
      }),
    });
  });

  it('Inicio de sesión exitoso', async () => {
    render(
      <TestWrapper>
        <Login />
      </TestWrapper>
    );
  });
});
```

Figura XXI Código test inicio de sesión.



The screenshot shows a test runner interface with a dark theme. At the top, it displays 'Login.test.jsx' with a green checkmark and a play button icon, followed by the file path 'C:/Users/HP/Desktop/web/Ejercicios/tesis/src/pages/tests/Login.test.jsx'. Below this is a search bar and navigation tabs for 'Report', 'Module Graph', 'Code', and 'Console (0)'. The main area shows a list of test results: 'Prueba de inicio de sesión' with a green checkmark and '1023ms', and 'Inicio de sesión exitoso' with a green checkmark and '513ms'. A large green banner at the bottom right states 'All tests passed in this file'.

Figura XXII Prueba unitaria - inicio de sesión.

```
describe('Prueba de cierre de sesión', () => {
  beforeEach(() => {
    globalThis.fetch = async () => ({
      ok: true,
      json: async () => ({
        detail: {
          token: 'fakeAuthToken',
          id: 'fakeUserId',
          rol: 'fakeRole',
          email: 'fake@example.com',
        },
      }),
    });
  });

  it('Cierre de sesión exitoso con token', async () => {
    render(
      <TestWrapper>
        <Navbar2 />
      </TestWrapper>
    );
  });
});
```

Figura XXIII Código test - cierre de sesión.

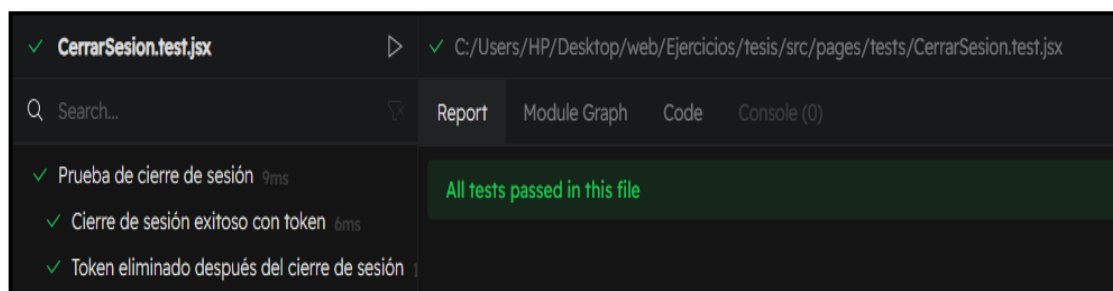


Figura XXIV Prueba unitaria - cierre de sesión.

```
it('Registro exitoso', async () => {
  const originalUseContext = globalThis.useContext;
  globalThis.useContext = () => ({
    auth: { isAuthenticated: false },
    setAuth: jest.fn(),
  });

  render(<Register />);
  userEvent.type(screen.getByLabelText(/nombre/i), 'John');
  userEvent.type(screen.getByLabelText(/apellido/i), 'Doe');
  userEvent.type(screen.getByLabelText(/correo electrónico/i), 'john.doe@example.com');
  userEvent.type(screen.getByLabelText(/contraseña/i), 'StrongPassword1#');
  userEvent.type(screen.getByLabelText(/teléfono/i), '1234567890');
  userEvent.click(screen.getByText(/registrar/i));
  await waitFor(() => {
    expect(screen.getByText(/registrando.../i)).toBeInTheDocument();
  });
});
```

Figura XXV Código test - registro.

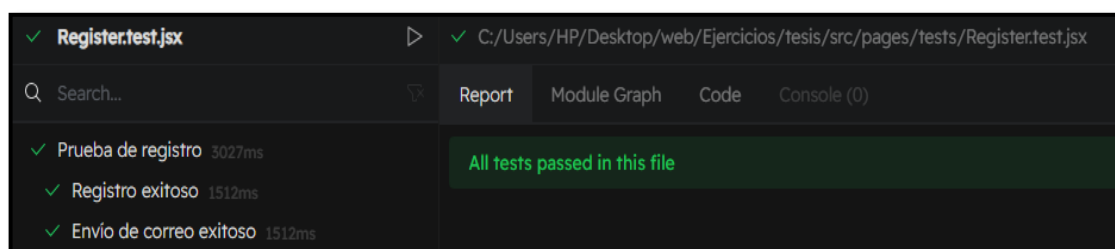


Figura XXVI Prueba unitaria – registro.

```
    },
  }),
});

it('Envío de correo exitoso', async () => {
  const originalUseContext = globalThis.useContext;
  globalThis.useContext = () => ({
    auth: { isAuthenticated: false },
    setAuth: jest.fn(),
  });

  render(<RecuperarCorreo />);

  userEvent.type(screen.getByLabelText(/correo electrónico/i), 'miguelcarapaz01@gmail.com');

  userEvent.click(screen.getByText(/recuperar cuenta/i));

  await waitFor(() => {
    expect(screen.getByText(/enviando correo.../i)).toBeInTheDocument();
  });

  expect(screen.getByText(/¡éxito!/i)).toBeInTheDocument();
});
```

Figura XXVII Código test - recuperar correo.

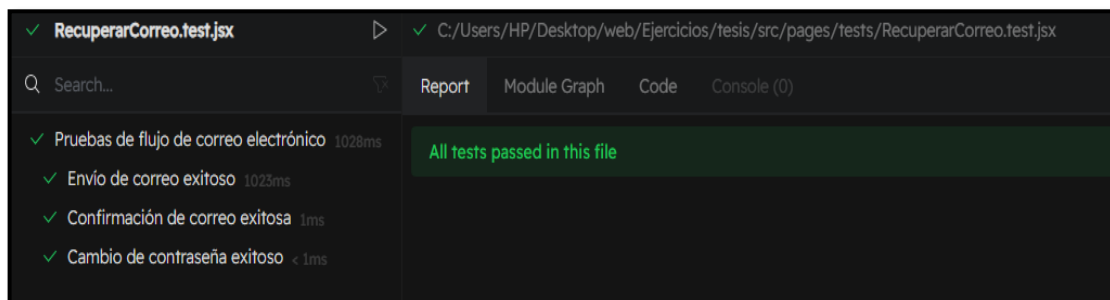


Figura XXVIII Prueba unitaria - recuperar correo.

```
it('Actualización de datos de perfil', async () => {
  render(<Perfil />);

  await waitFor(() => {
    expect(screen.getByText(/john.doe@example.com/i)).toBeInTheDocument();
    expect(screen.getByText(/John/i)).toBeInTheDocument();
    expect(screen.getByText(/Doe/i)).toBeInTheDocument();
    expect(screen.getByText(/1234567890/i)).toBeInTheDocument();
  });

  userEvent.click(screen.getByText(/actualizar perfil/i));

  userEvent.clear(screen.getByLabelText(/nombre/i));
  userEvent.type(screen.getByLabelText(/nombre/i), 'Jane');

  userEvent.click(screen.getByText(/confirmar/i));

  await waitFor(() => {
    expect(screen.getByText(/datos actualizados con éxito/i)).toBeInTheDocument();
  });
});
```

Figura XXIX Código test - perfil.

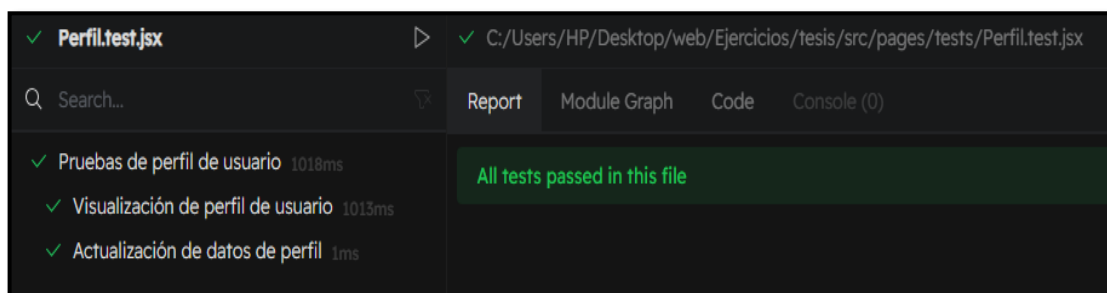


Figura XXX Prueba unitaria - perfil.

```

describe('Pruebas de gestión de categorías', () => {
  it('Visualizar categorías existentes', async () => {
    axios.get.mockResolvedValueOnce({ data: mockCategoriesData });

    render(<Categorias />);

    await waitFor(() => {
      expect(screen.getByText(/cargando categorías/i)).toBeInTheDocument();
    });

    await waitFor(() => {
      expect(screen.getByText(/categorías:/i)).toBeInTheDocument();
      expect(screen.getByText(/category 1/i)).toBeInTheDocument();
      expect(screen.getByText(/category 2/i)).toBeInTheDocument();
    });
  });

  it('Editar categoría existente', async () => {
    axios.get.mockResolvedValueOnce({ data: mockCategoriesData });
    render(<Categorias />);

    await waitFor(() => {
      expect(screen.getByText(/cargando categorías/i)).toBeInTheDocument();
    });
  });
});

```

Figura XXXI Código test – categorías.

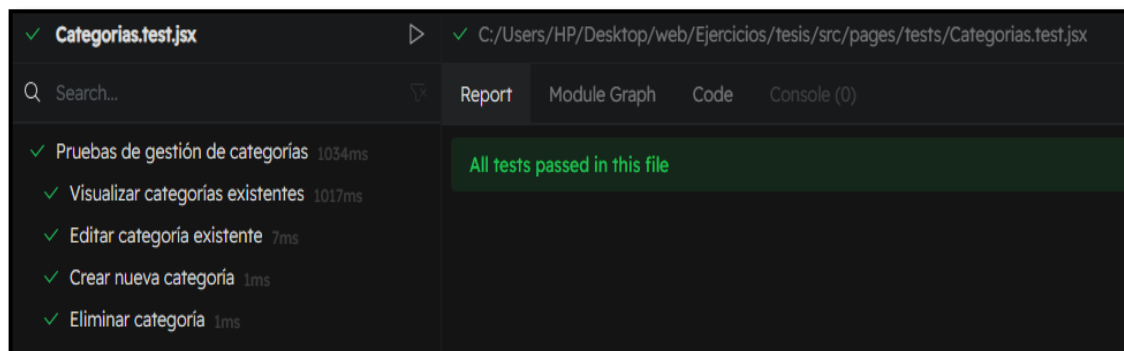


Figura XXXII Prueba unitaria – categorías.

```
beforeEach(() => {
  globalThis.fetch = async (url) => {
    if (url.includes('/products')) {
      return Promise.resolve({
        ok: true,
        json: async () => ({
          detail: mockProducts,
        }),
      });
    } else if (url.includes('/category')) {
      return Promise.resolve({
        ok: true,
        json: async () => ({
          detail: mockCategories,
        }),
      });
    }
  };

  jest.spyOn(window.localStorage.__proto__, 'getItem').mockImplementation(() => 'fakeAuthToken');
});

describe('Visualizar productos existentes', () => {
  it('Mostrar lista de productos', async () => {
    render(<Products3 />);
```

Figura XXXIII Código test - visualizar productos

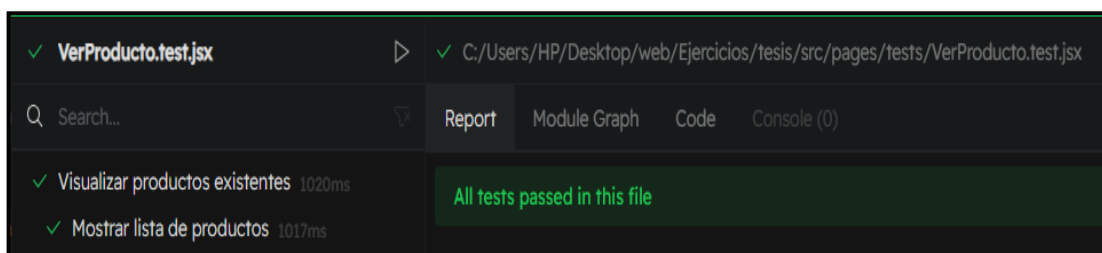


Figura XXXIV Prueba unitaria - visualizar productos.


```

describe('Crear nuevo producto', () => {
  it('Crear un nuevo producto', async () => {
    axios.get.mockResolvedValueOnce({ data: mockCategoriasData });
    render(<NuevoProducto />);
    await waitFor(() => {
      expect(screen.getByLabelText(/imágenes del producto/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/nombre del producto/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/descripción/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/precio/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/categoría/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/tallas/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/colores/i)).toBeInTheDocument();
    });

    fireEvent.change(screen.getByLabelText(/categoría/i), { target: { value: 'Categoría de prueba' }});
    fireEvent.click(screen.getByText(/agregar talla/i));
    fireEvent.change(screen.getByLabelText(/talla #1/i), { target: { value: 'S' } });
    fireEvent.click(screen.getByText(/agregar color/i));
    fireEvent.change(screen.getByLabelText(/nombre del color/i), { target: { value: 'Rojo' } });
    fireEvent.click(screen.getByText(/crear producto/i));
    expect(axios.post).toHaveBeenCalledWith(expect.stringContaining('/products'), expect.anything());
  });
});

```

Figura XXXV Código test - crear nuevo producto.

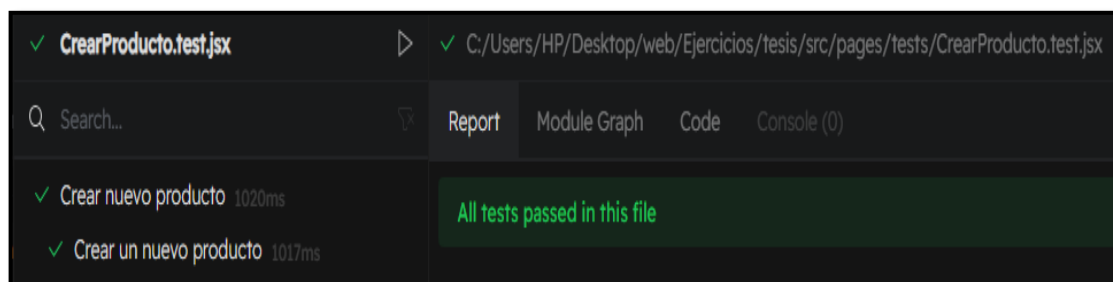


Figura XXXVI Prueba unitaria - crear nuevo producto.

```

describe('Editar producto', () => {
  it('Actualizar producto', async () => {
    render(
      <MemoryRouter>
      <ActualizarProducto />
      </MemoryRouter>
    );

    await waitFor(() => {
      expect(screen.getByLabelText(/nombre del producto/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/descripción/i)).toBeInTheDocument();
      expect(screen.getByLabelText(/precio/i)).toBeInTheDocument();
    });

    userEvent.clear(screen.getByLabelText(/nombre del producto/i));
    userEvent.type(screen.getByLabelText(/nombre del producto/i), 'Nuevo Nombre');

    userEvent.clear(screen.getByLabelText(/descripción/i));
    userEvent.type(screen.getByLabelText(/descripción/i), 'Nueva Descripción');

    userEvent.clear(screen.getByLabelText(/precio/i));
    userEvent.type(screen.getByLabelText(/precio/i), '15');
    userEvent.click(screen.getByText(/actualizar producto/i));
    await waitFor(() => {
      expect(screen.getByText(/producto actualizado exitosamente/i)).toBeInTheDocument();
    });
  });
});

```

Figura XXXVII Código test - actualizar producto.

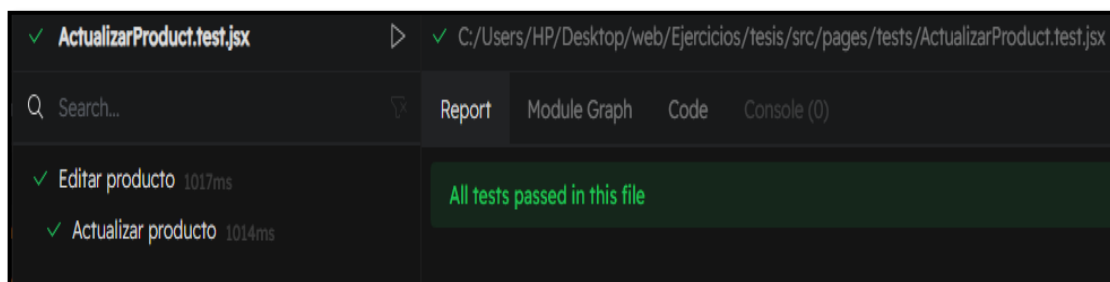


Figura XXXVIII Prueba unitaria - actualizar producto.

```
describe('Eliminar producto', () => {
  it('Eliminar producto', async () => {
    render(<Products3 />);
    await waitFor(() => {
      expect(screen.getByText(/Product 1/i)).toBeInTheDocument();
      expect(screen.getByText(/Category 1/i)).toBeInTheDocument();
      expect(screen.getByText(/$10/i)).toBeInTheDocument();
    });

    fireEvent.click(screen.getByText(/Eliminar/i));

    await waitFor(() => {
      expect(screen.getByText(/¿Estás seguro?/i)).toBeInTheDocument();
    });

    fireEvent.click(screen.getByText(/Sí, eliminarlo/i));

    await waitFor(() => {
      expect(screen.queryByText(/Product 1/i)).not.toBeInTheDocument();
    });
  });
});
```

Figura XXXIX Código test - eliminar producto.

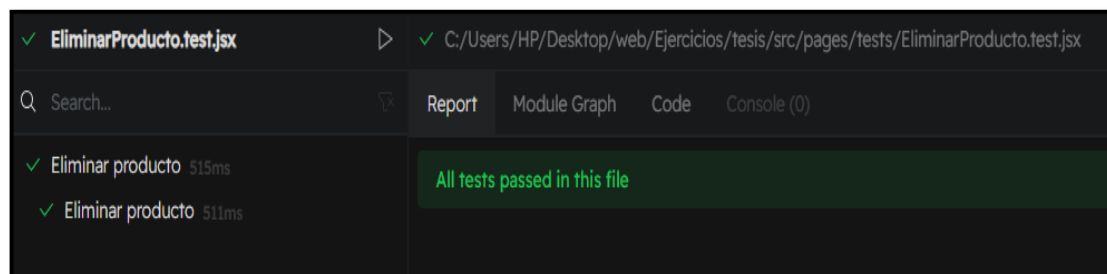


Figura XL Prueba unitaria - eliminar producto.

```
beforeEach(() => {
  globalThis.fetch = async (url, options) => {
    if (url.includes('/orders')) {
      return Promise.resolve({
        ok: true,
        json: async () => ({
          detail: [
            {
              id: 'fakeOrderId',
              fecha: '2024-01-18',
              id_producto: {
                name: 'Product Name',
                tallas: [{ name: 'S', status: 'true' }],
                colores: [{ name: 'Red', status: 'true' }],
                precio: 10,
              },
              id_orden: {
                nombre: 'John',
                apellido: 'Doe',
                email: 'john.doe@example.com',
                telefono: '1234567890',
                direccion: '123 Main St',
                descripcion: 'Order description',
              },
              status: 'Pendiente',
            },
          ],
        }),
      });
    }
  };
});
```

Figura XLI Código test - listar pedidos.

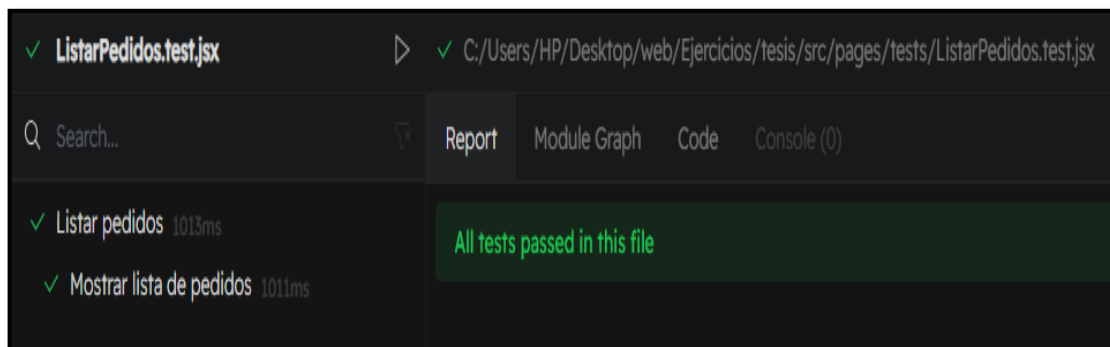


Figura XLII Prueba unitaria - listar pedidos.



Figura XLIII Código test - actualizar estado del pedido.

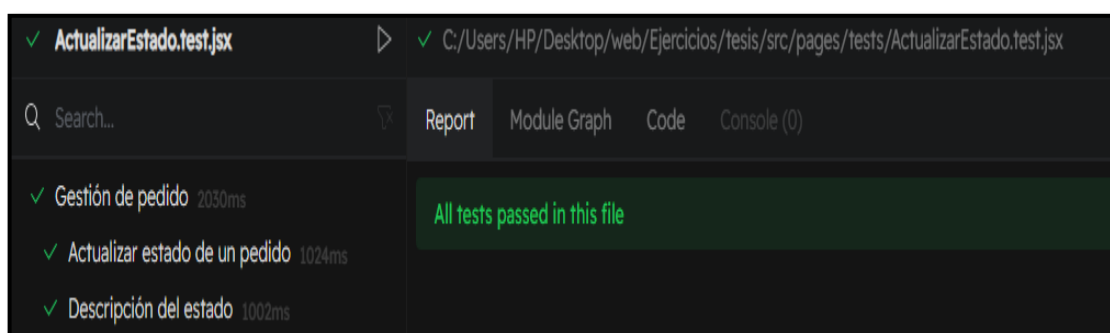


Figura XLIV Prueba unitaria - actualizar estado del pedido.

```
it('Detalle de pedido', async () => {
  render(<VerPedidosAdmin />);

  await waitFor(() => {
    expect(screen.getByText(/Nombre del Producto 1/i)).toBeInTheDocument();
    expect(screen.getByText(/Pendiente/i)).toBeInTheDocument();
  });

  fireEvent.click(screen.getByText(/👁/));

  await waitFor(() => {
    expect(screen.getByText(/Detalles del Pedido/i)).toBeInTheDocument();
    expect(screen.getByText(/Producto Adquirido:/i)).toBeInTheDocument();
    expect(screen.getByText(/Tallas Disponibles:/i)).toBeInTheDocument();
    expect(screen.getByText(/Colores Disponibles:/i)).toBeInTheDocument();
    expect(screen.getByText(/Nombre y Apellido:/i)).toBeInTheDocument();
    expect(screen.getByText(/Email:/i)).toBeInTheDocument();
    expect(screen.getByText(/Teléfono:/i)).toBeInTheDocument();
    expect(screen.getByText(/Dirección:/i)).toBeInTheDocument();
    expect(screen.getByText(/Descripción del pedido:/i)).toBeInTheDocument();
    expect(screen.getByText(/Total:/i)).toBeInTheDocument();
    expect(screen.getByText(/Estado:/i)).toBeInTheDocument();
    expect(screen.getByText(/Voucher:/i)).toBeInTheDocument();
  });
});
```

Figura XLV Código test - detalle del pedido.

The screenshot shows a test runner interface with the following elements:

- File name: `DetallePedido.test.jsx` (with a green checkmark)
- Path: `C:/Users/HP/Desktop/web/Ejercicios/tesis/src/pages/tests/DetallePedido.test.jsx` (with a green checkmark)
- Search bar: "Search..."
- Navigation tabs: "Report", "Module Graph", "Code", "Console (0)"
- Test results list:
 - ✓ Ver pedido 2030ms
 - ✓ Ver un pedido existente 1022ms
 - ✓ Detalle de pedido 1006ms
- Summary message: "All tests passed in this file" (in a green box)

Figura XLVI Prueba unitaria - detalle del pedido.

```

it('Eliminar comentarios', async () => {
  render(
    <AuthContext.Provider value={mockAuthContext}>
      <Comentarios3 productId="product1" />
    </AuthContext.Provider>
  );

  await waitFor(() => {
    expect(screen.getByText(/cargando comentarios/i)).toBeInTheDocument();
  });

  await waitFor(() => {
    expect(screen.getByText(/comentarios:/i)).toBeInTheDocument();
  });

  fireEvent.click(screen.getByText(/X/));

  await waitFor(() => {
    expect(Swal.fire).toHaveBeenCalledWith({
      title: '¿Está seguro de eliminar este comentario?',
      showCancelButton: true,
      confirmButtonText: 'Sí, eliminar',
      cancelButtonText: 'Cancelar',
    });
  });
});

```

Figura XLVII Código test - comentarios administrador.

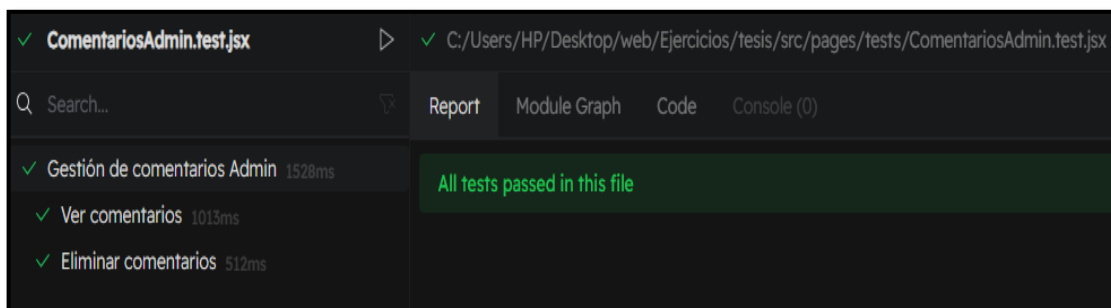


Figura XLVIII Prueba unitaria - comentarios administrador.

```
import Products2 from '../usuario/Products2';

beforeEach(() => {
  globalThis.fetch = async (url) => {
    if (url.includes('/products')) {
      return Promise.resolve({
        ok: true,
        json: async () => ({
          detail: [
            {
              id: '1',
              name: 'Product 1',
              precio: 50.0,
              tallas: [{ name: 'S' }, { name: 'M' }],
              colores: [{ name: 'Red' }, { name: 'Blue' }],
              descripcion: 'Lorem ipsum dolor sit amet.',
              imagen: [
                { secure_url: 'image1.jpg' },
                { secure_url: 'image2.jpg' },
              ],
            },
            {
              id: '2',
              name: 'Product 2',
              precio: 60.0,
              tallas: [{ name: 'L' }, { name: 'XL' }],
            }
          ]
        })
      }
    }
  }
});
```

Figura XLIX Código test - ver productos usuarios.

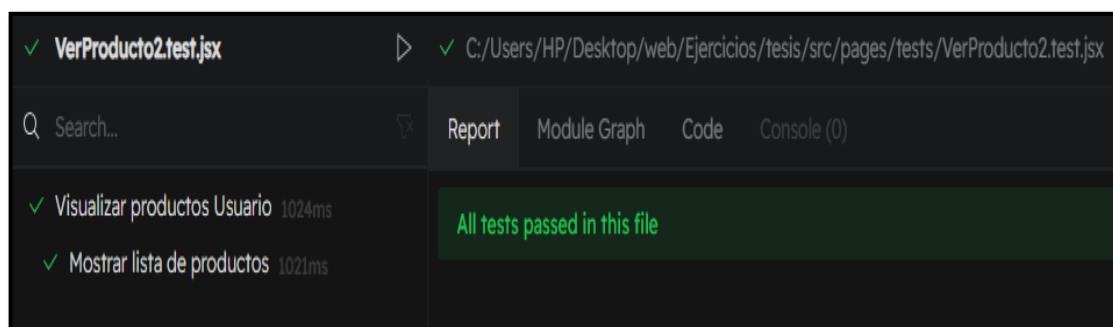


Figura L Prueba unitaria - ver productos usuarios.


```
    expect(screen.getByText(/Tu carrito está vacío/i)).toBeInTheDocument();
    expect(screen.getByText(/Continuar comprando/i)).toBeInTheDocument();
  });
});

it('Ver carrito', async () => {
  const initialState = {
    handleCart: [
      {
        id_producto: '1',
        nombre_producto: 'Product 1',
        precio: 50.0,
        cantidad: 2,
        imagen: [{ secure_url: 'image1.jpg' }],
      },
      {
        id_producto: '2',
        nombre_producto: 'Product 2',
        precio: 60.0,
        cantidad: 1,
        imagen: [{ secure_url: 'image2.jpg' }],
      },
    ],
  };
});
```

Figura LI Código test - carrito de compras.

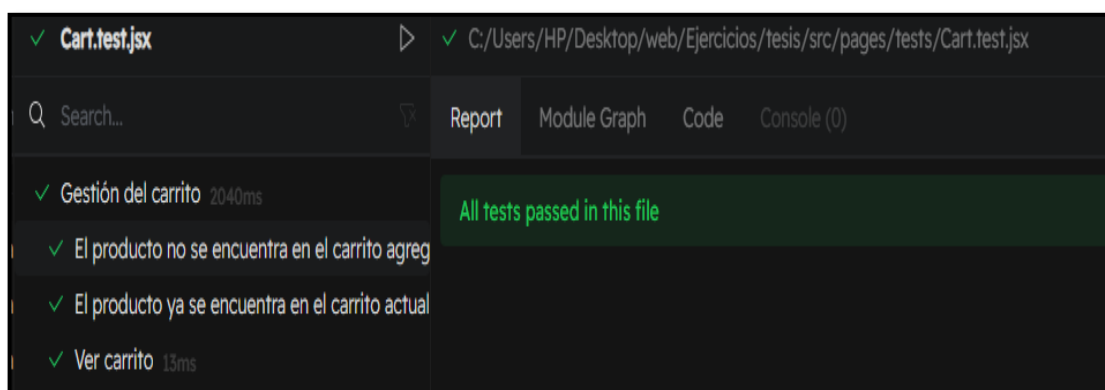


Figura LII Prueba unitaria - carrito de compras.

```

it('Crear comentario', async () => {
  render(
    <AuthContext.Provider value={mockAuthContext}>
      <Comentarios2 productId="product1" />
    </AuthContext.Provider>
  );

  await waitFor(() => {
    expect(screen.getByText(/cargando comentarios/i)).toBeInTheDocument();
  });

  await waitFor(() => {
    expect(screen.getByText(/comentarios:/i)).toBeInTheDocument();
    expect(screen.getByPlaceholderText(/escribe tu comentario.../i)).toBeInTheDocument();
  });
  fireEvent.click(screen.getAllByTestId('star-icon')[3]);
  fireEvent.change(screen.getByPlaceholderText(/escribe tu comentario.../i), { target: { va
  fireEvent.click(screen.getByText(/publicar comentario/i));
  await waitFor(() => {
    expect(screen.getByText(/cargando comentarios/i)).toBeInTheDocument();
  });
  expect(axios.post).toHaveBeenCalledWith(expect.stringContaining('/comments'), expect.anytl

```

Figura LIII Código test - comentarios usuarios.

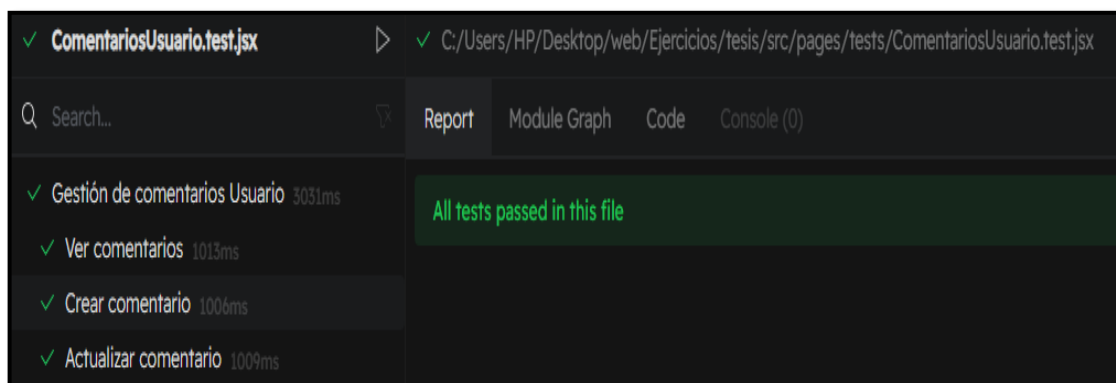


Figura LIV Prueba unitaria - comentarios usuarios.

Pruebas de compatibilidad

Esta evaluación se centra en examinar las diversas funcionalidades en los navegadores más comúnmente empleados, con el objetivo de detectar posibles complicaciones al mostrar el contenido al usuario. De esta manera, se exhiben los resultados que se han obtenido en los diferentes navegadores desde la **Figura LV** hasta la **Figura LXXV**.

Brave

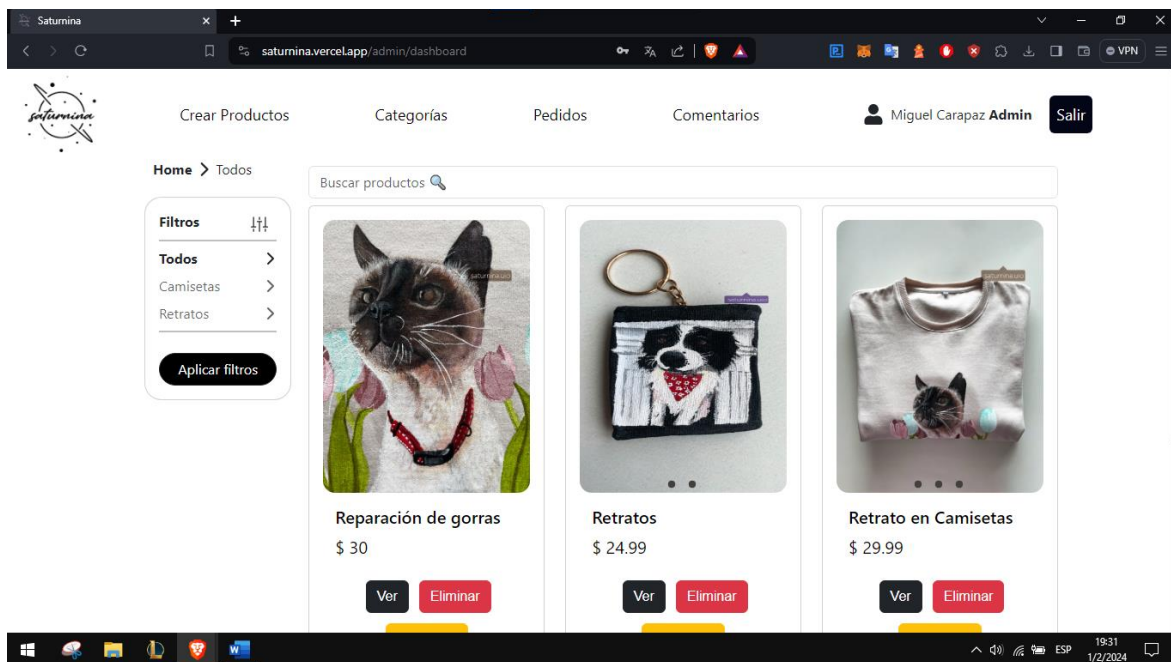


Figura LV Brave – Visualizar Productos.

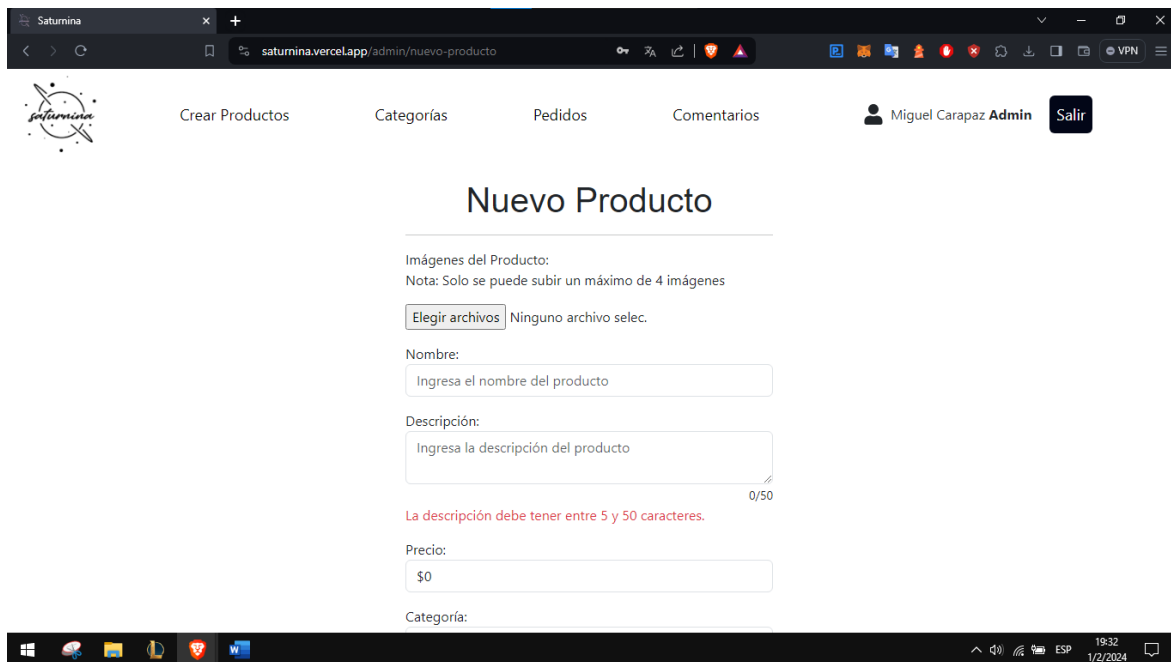


Figura LVI Brave – Gestionar Productos.

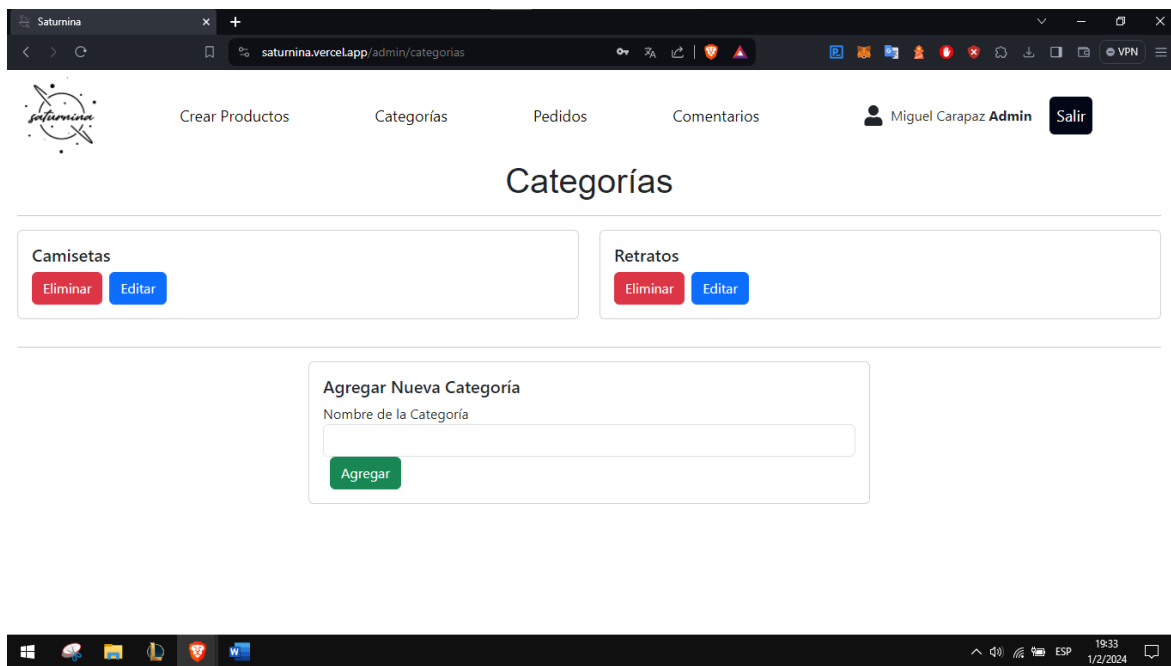


Figura LVII Brave – Gestionar Categorías.

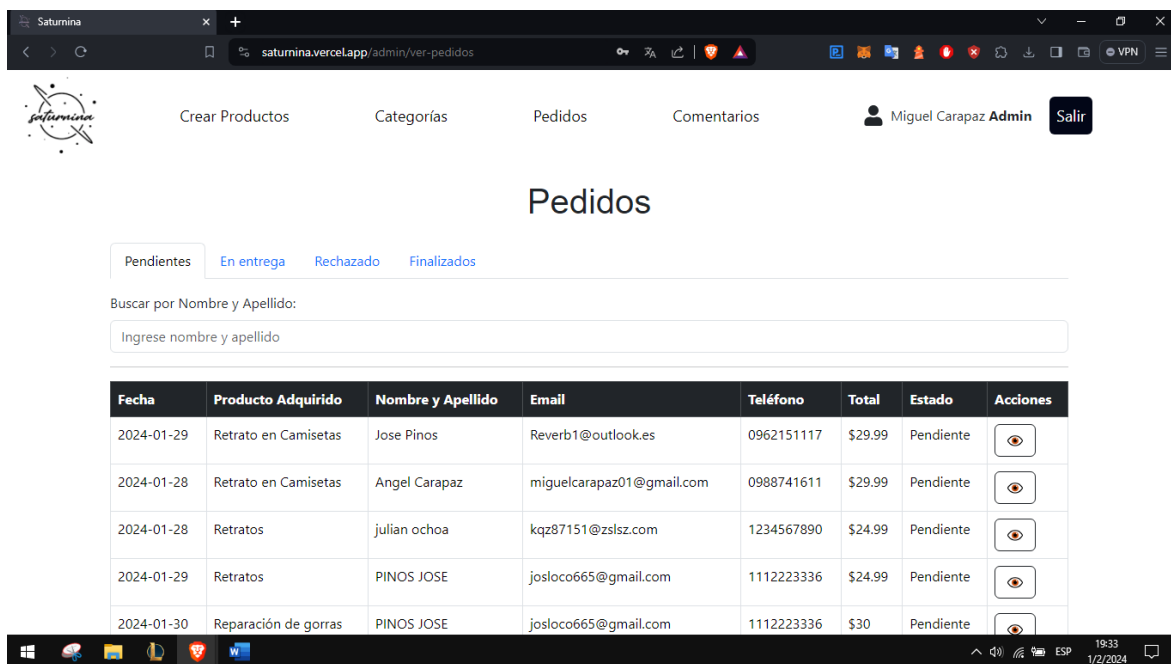


Figura LVIII Brave – Gestionar Pedidos.



Figura LIX Brave – Comentarios y/o Sugerencias.

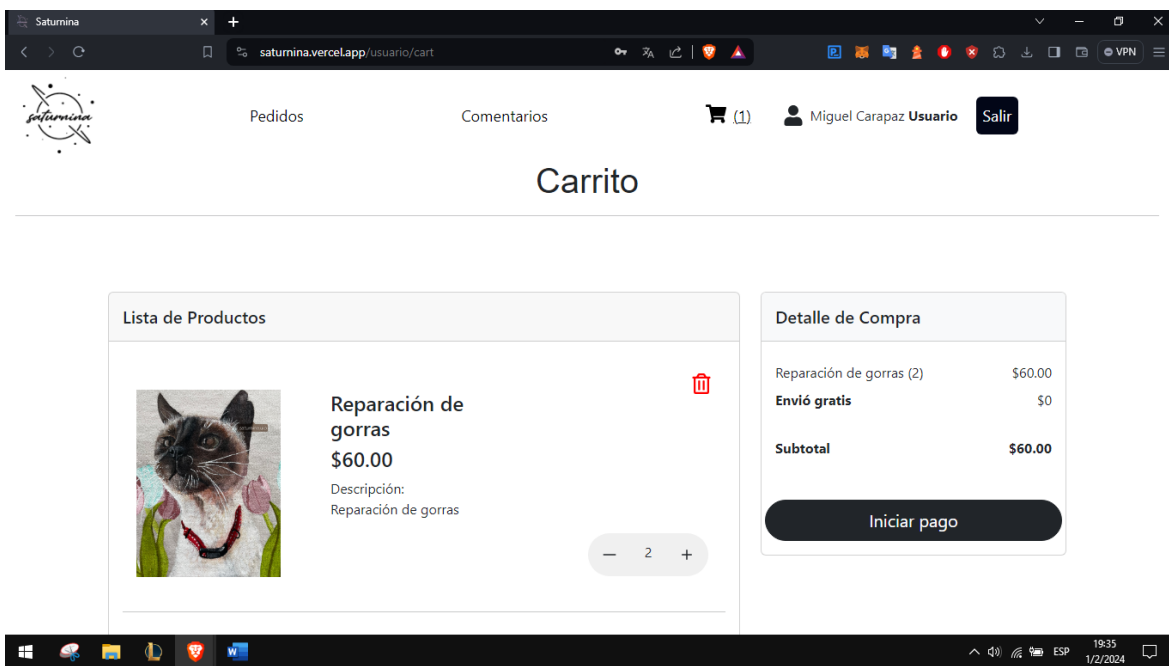


Figura LX Brave – Gestionar Carrito de Compras.

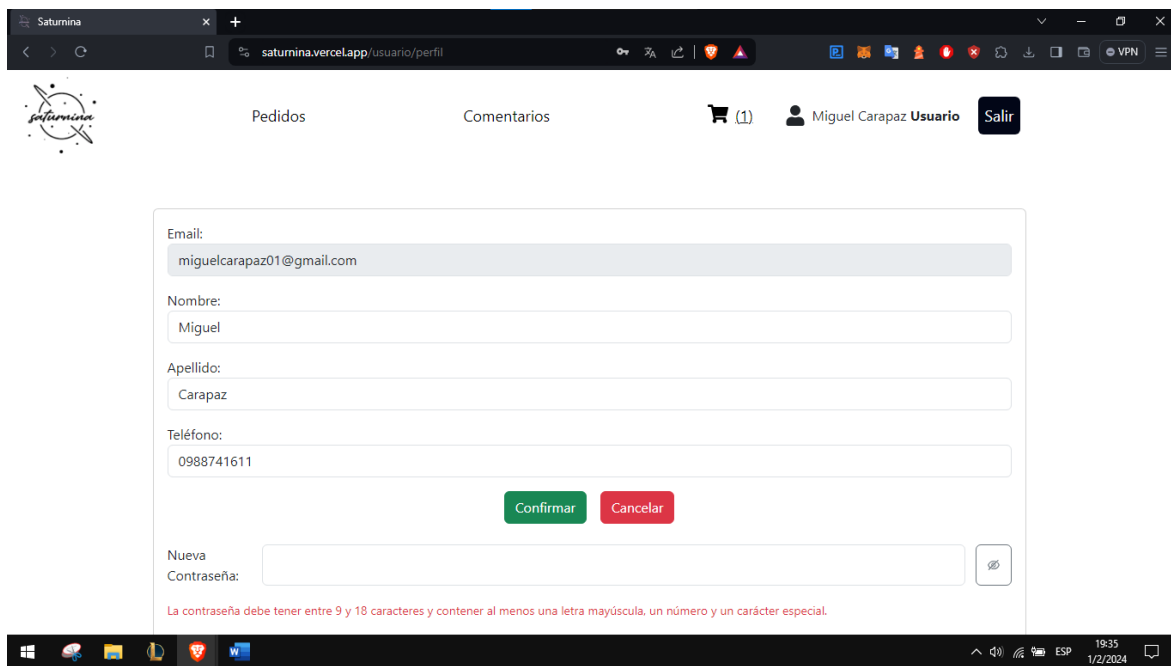


Figura LXI Brave – Perfil de Usuario.

Google Chrome

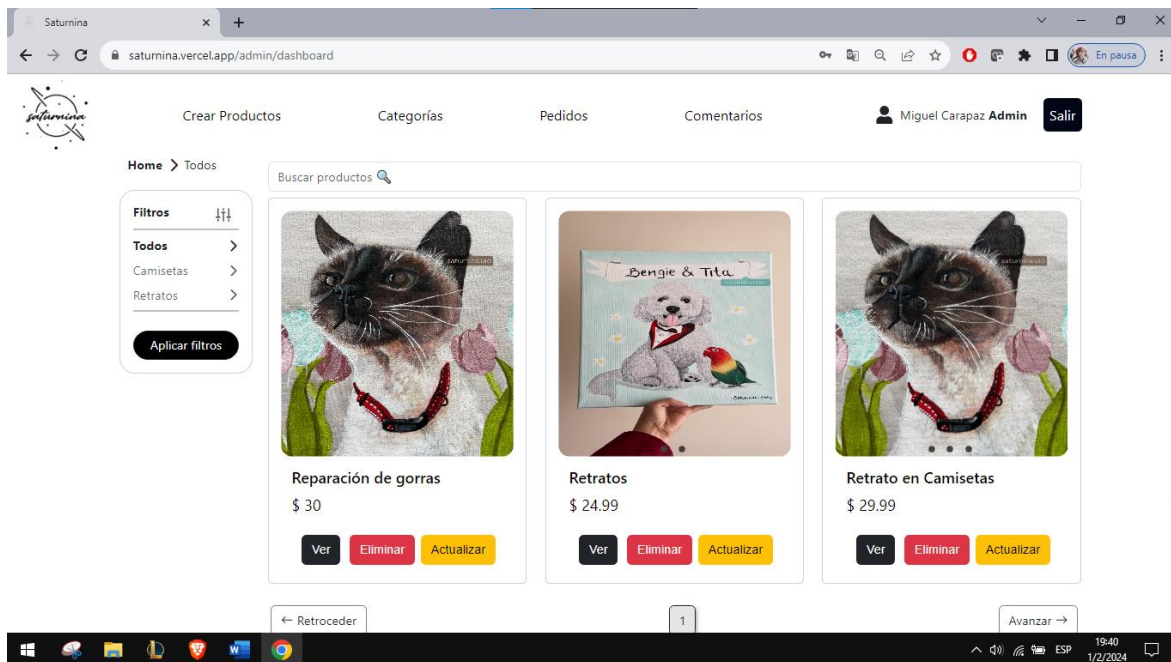


Figura LXII Google Chrome – Visualizar Productos.

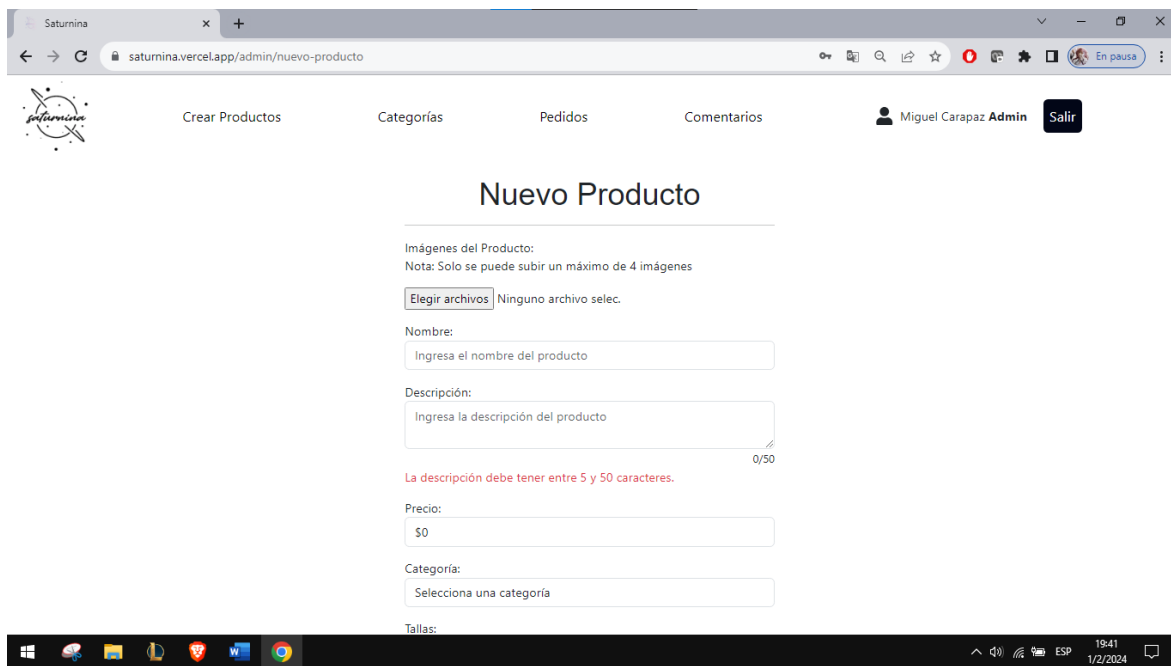


Figura LXIII Google Chrome – Gestionar Productos.

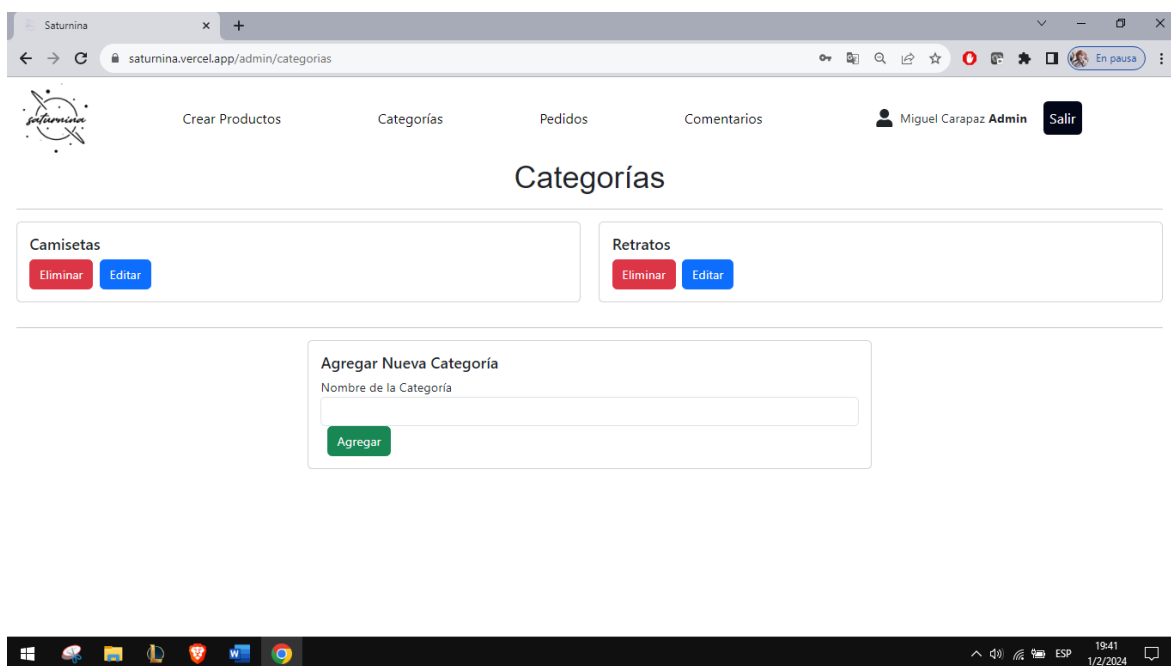


Figura LXIV Google Chrome – Gestionar Categorías.

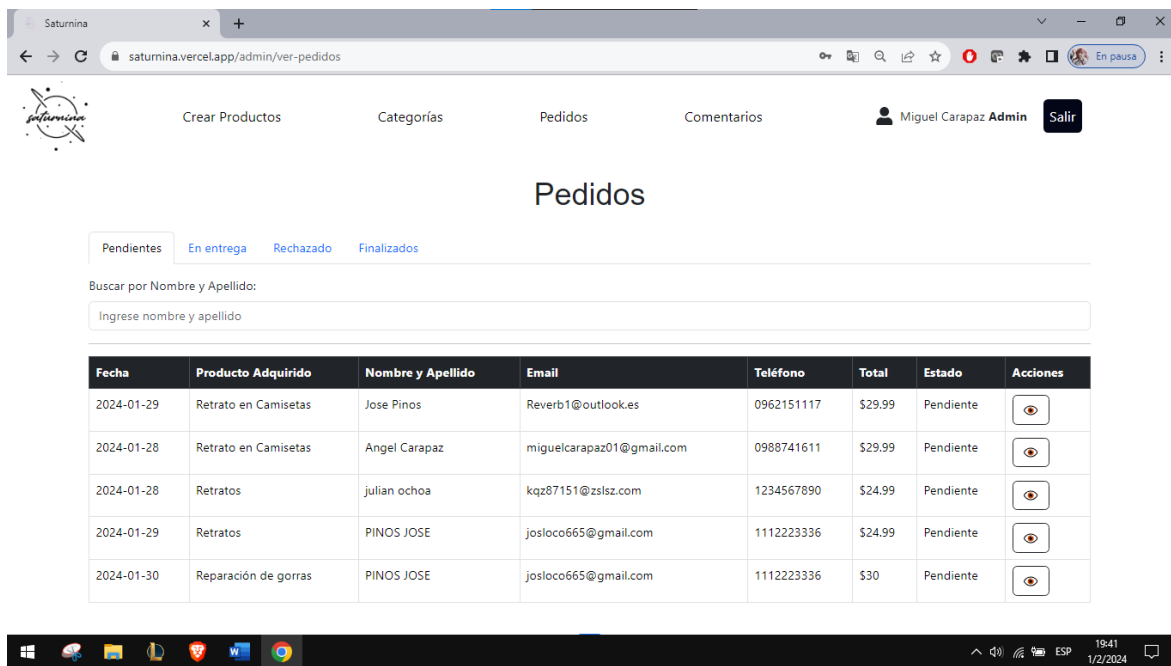


Figura LXV Google Chrome – Gestionar Pedidos.

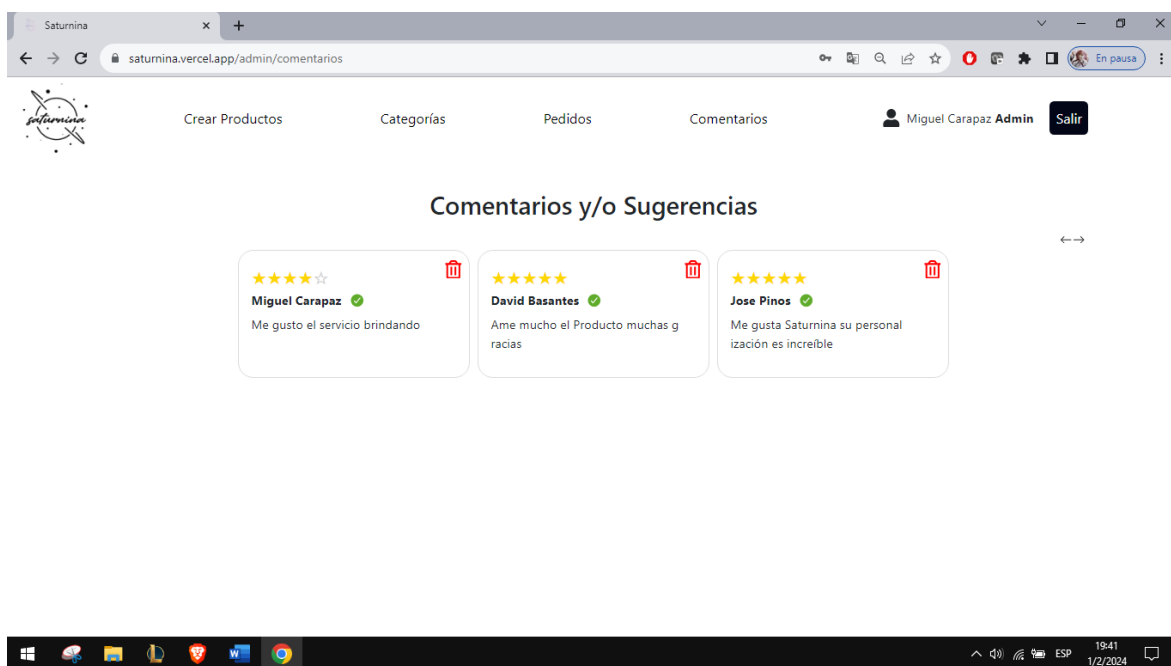


Figura LXVI Google Chrome – Comentarios y/o Sugerencias.

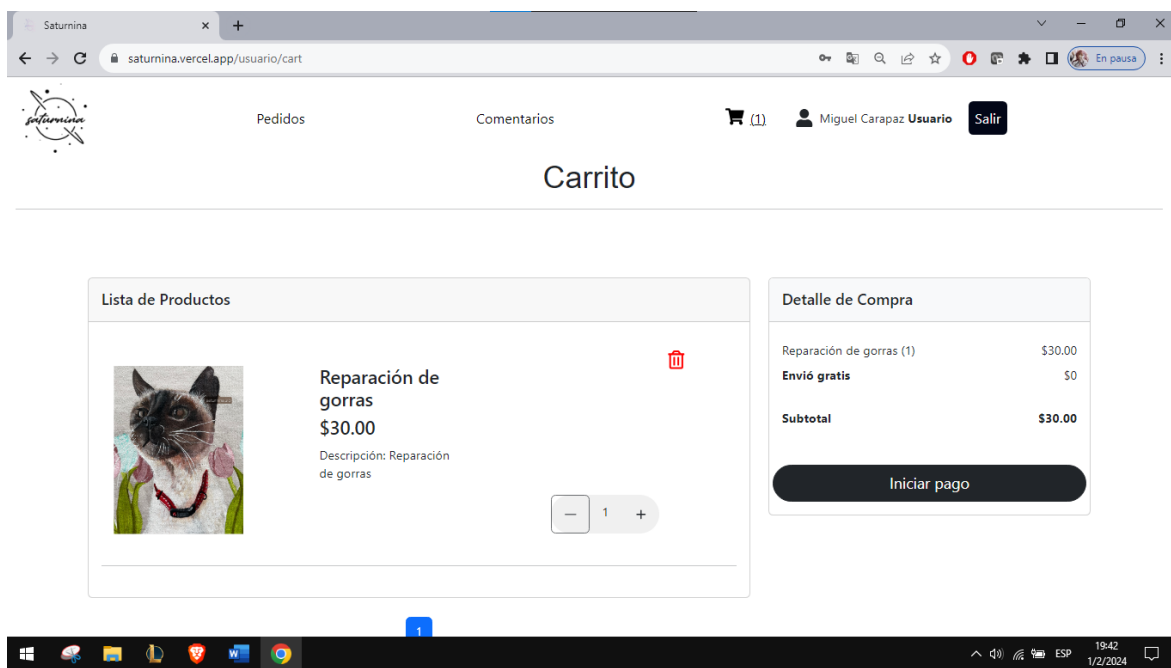


Figura LXVII Google Chrome – Gestionar Carrito de Compras.

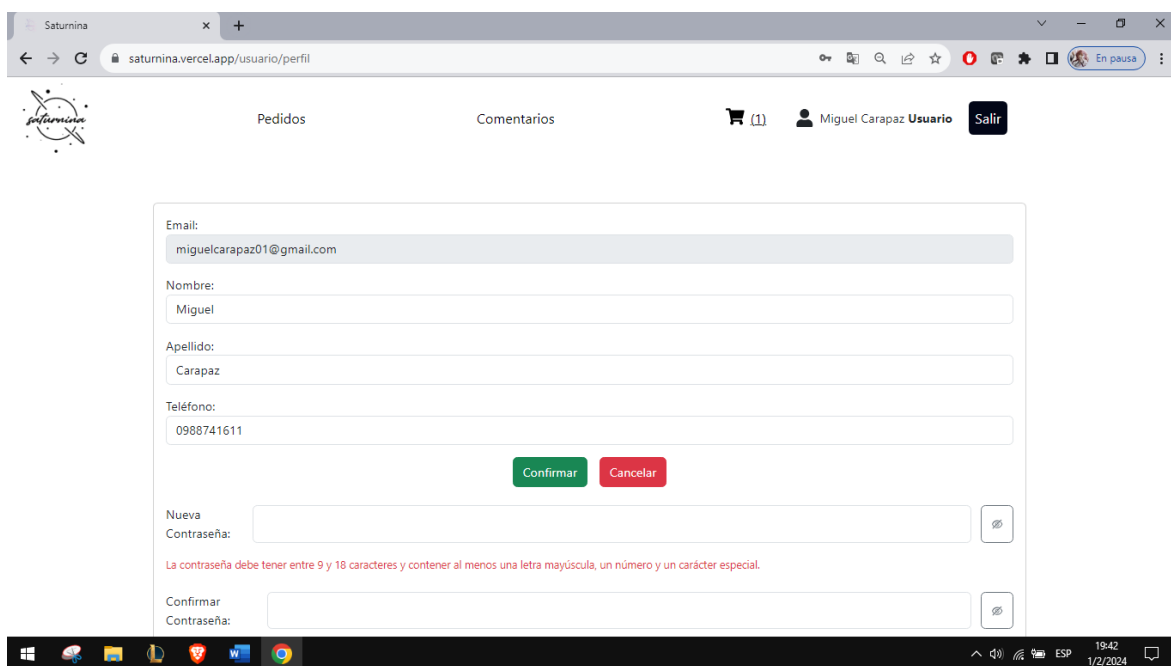


Figura LXVIII Google Chrome – Perfil de Usuario.

Microsoft Edge

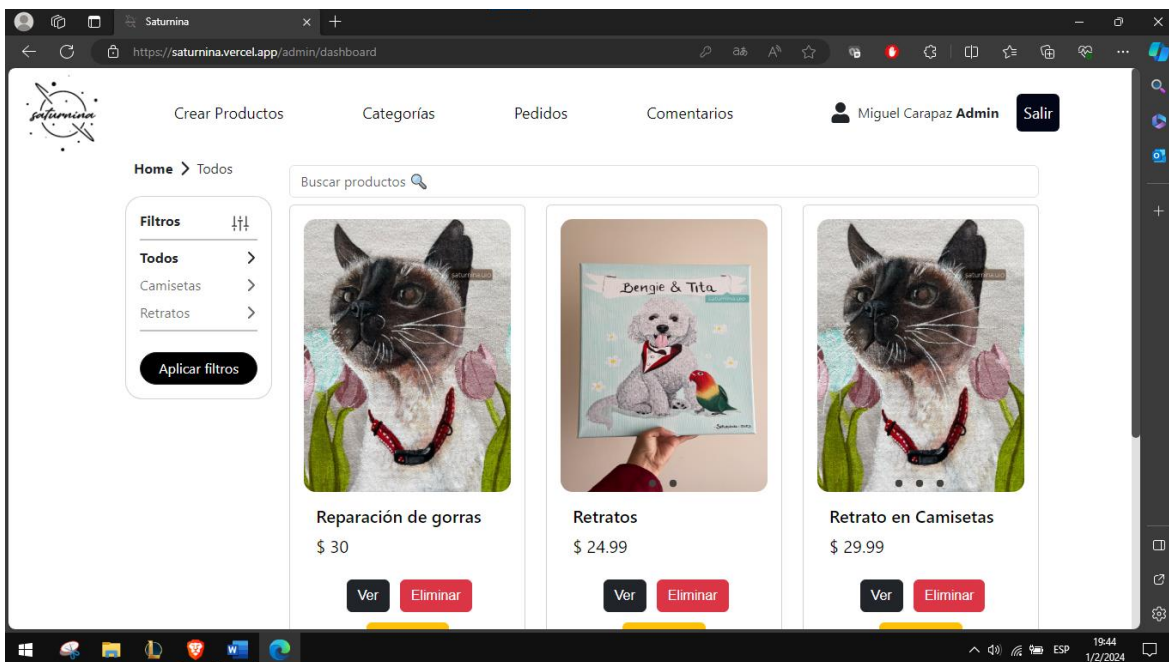


Figura LXIX Microsoft Edge – Visualizar Productos.

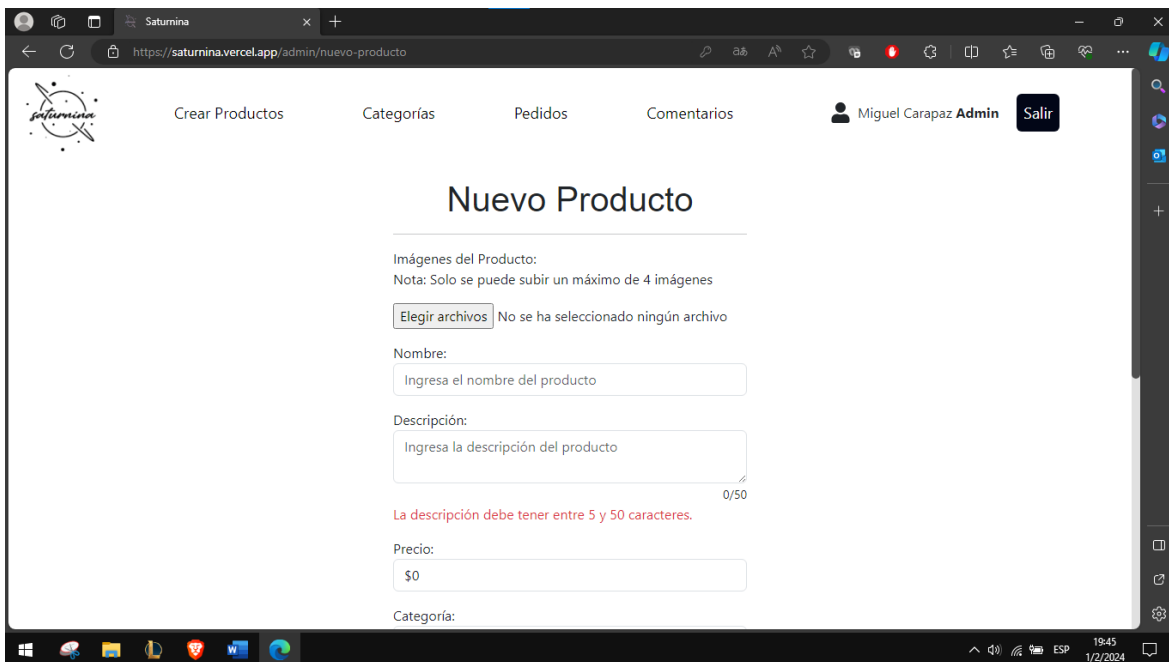


Figura LXX Microsoft Edge – Gestionar Productos.

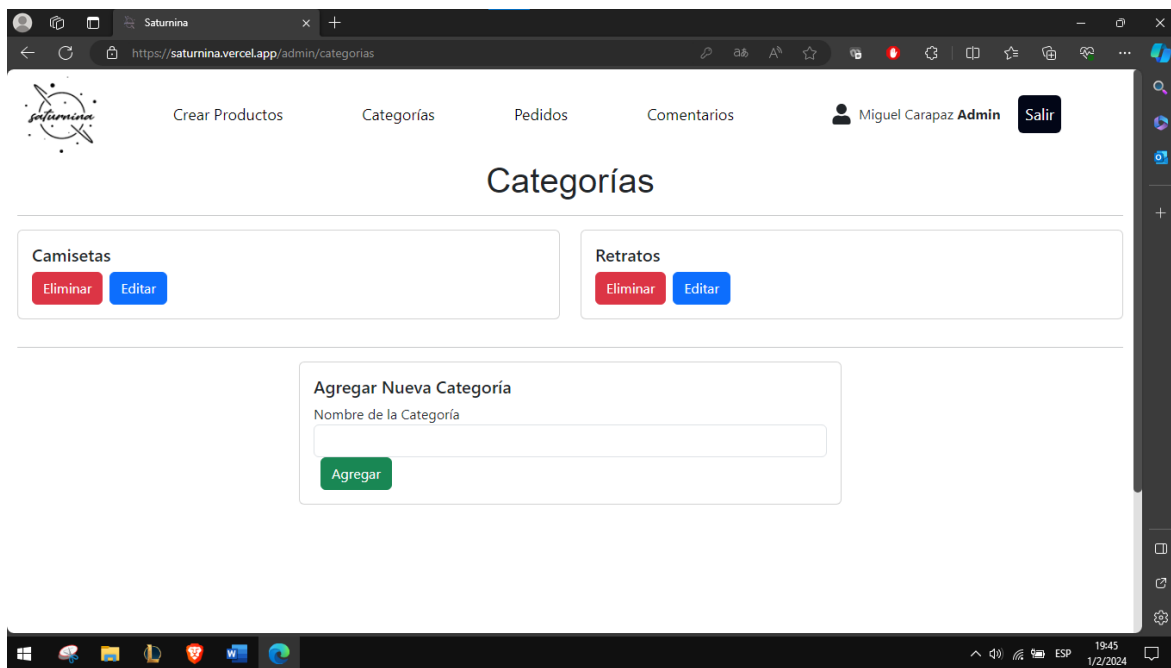


Figura LXXI Microsoft Edge – Gestionar Categorías.

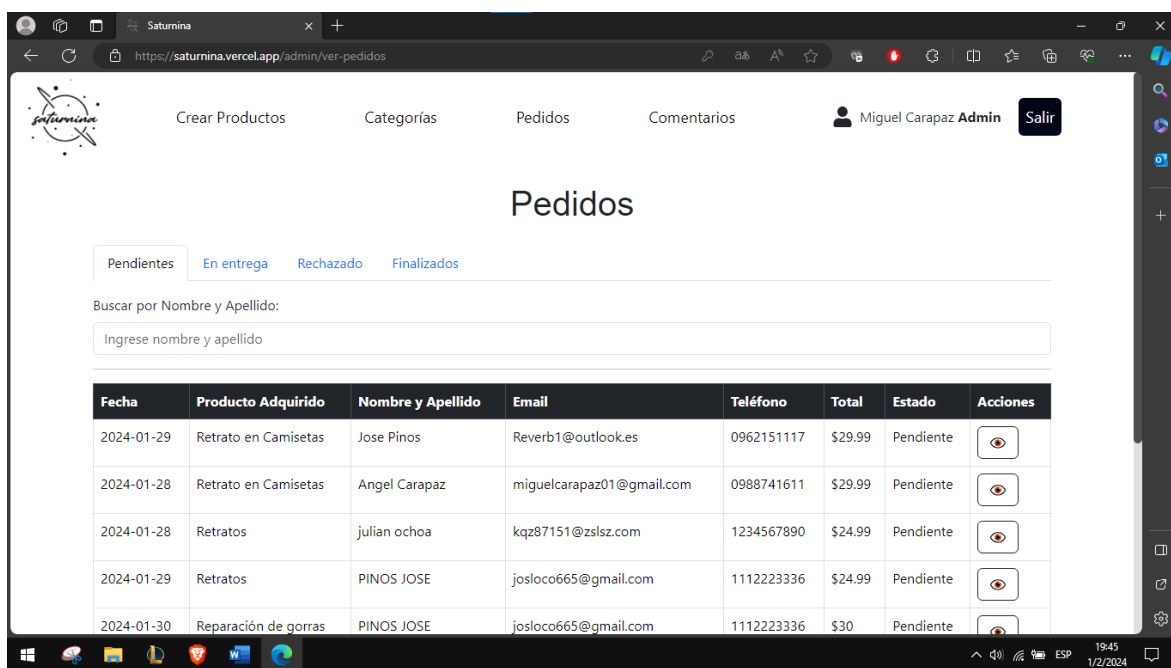


Figura LXXII Microsoft Edge – Gestionar Pedidos.



Figura LXXIII Microsoft Edge – Comentarios y/o Sugerencias.

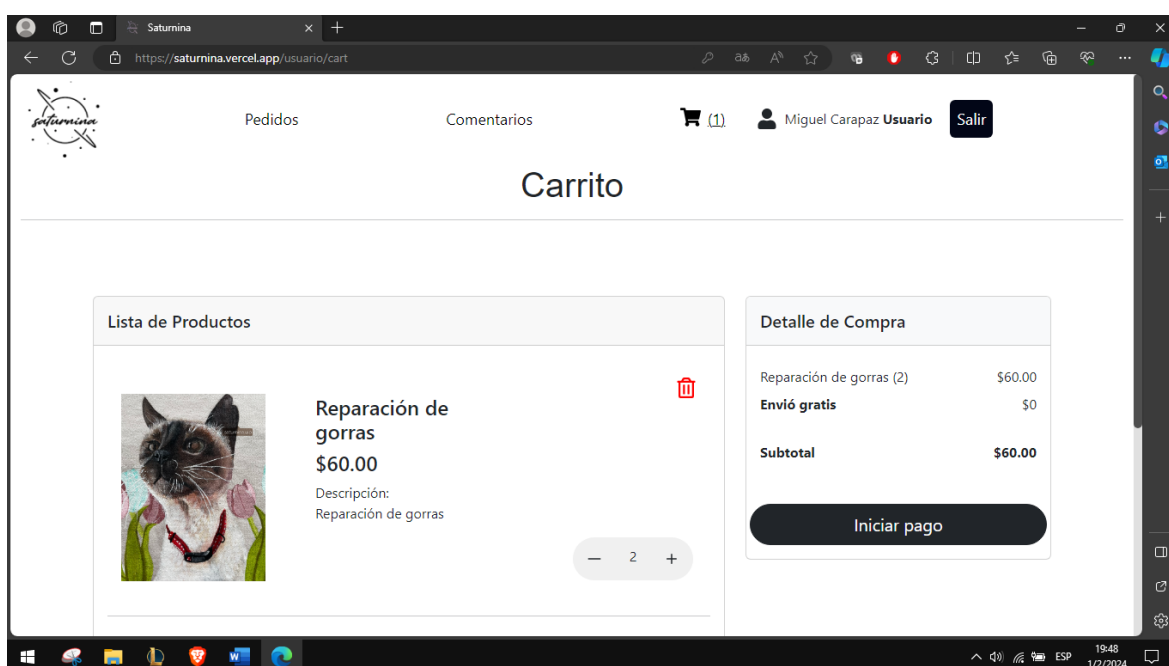


Figura LXXIV Microsoft Edge – Gestionar Carrito de Compras.

Email: miguelcarapaz01@gmail.com
 Nombre: Miguel
 Apellido: Carapaz
 Teléfono: 0988741611
 Nueva Contraseña:

[Confirmar](#) [Cancelar](#)

La contraseña debe tener entre 9 y 18 caracteres y contener al menos una letra mayúscula, un número y un carácter especial.

Figura LXXV Microsoft Edge – Perfil de Usuario.

Pruebas de aceptación

A continuación, se enlistan las pruebas de aceptación, que comprenden desde la **Tabla XV** hasta la **Tabla XXV**. Estas evaluaciones resultan cruciales para la descripción y confirmación del procedimiento asociado a las diversas responsabilidades asignadas a los usuarios del frontend.

Tabla XV Prueba de aceptación – registrarse.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA001	Identificador de historia de Usuario: HU001
Nombre: Registrarse	
Descripción: El usuario administrador y cliente en el frontend debe consumir varios endpoints para: <ul style="list-style-type: none"> • Registrarse 	
Pasos de ejecución: Para registrarse todos los usuarios deberán: <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • En lado derecho superior dar clic en “Registrarse”. • Llenar el formulario respectivo para el registro. 	

<ul style="list-style-type: none"> • Dar clic en el botón “Registrarme”. • Confirmar el correo enviado a tu correo.
Resultado deseado: El componente frontend admite el registro de todos los usuarios.
Evaluación de la prueba: El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.

Tabla XVI Prueba de aceptación – modificar perfil.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA003	Identificador de historia de Usuario: HU003
Nombre: Modificar Perfil	
Descripción: El usuario administrador y cliente en el frontend debe consumir varios endpoints para: <ul style="list-style-type: none"> • Visualizar y editar su perfil 	
Pasos de ejecución: Para visualizar y editar el perfil todos los usuarios deberán: <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • En el lado superior derecho dar clic en perfil. • Visualizar sus datos. • Dar clic en el botón “Actualizar perfil”. • Modificar los datos que quiera de su perfil. • Dar clic en el botón “Aceptar”. 	
Resultado deseado: El componente frontend admite la visualización y modificación del perfil de todos los usuarios.	
Evaluación de la prueba: El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.	

Tabla XVII Prueba de aceptación – gestionar categorías.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA004	Identificador de historia de Usuario: HU004
Nombre: Gestionar Categorías	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Registrar categorías • Modificar categorías • Eliminar categorías • Visualizar categorías 	
<p>Pasos de ejecución:</p> <p>Para registrar categorías el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • En el lado superior izquierdo dar clic en crear nuevo producto. • Dentro del formulario para crear nuevo producto dar clic en el botón “Categorías”. • Llenar el campo con el nombre de la categoría. • Dar clic en el botón crear nueva categoría. <p>Para modificar categorías el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En el lado superior izquierdo dar clic en crear nuevo producto. • Dentro del formulario para crear nuevo producto dar clic en el botón “Categorías”. • Dar clic en el botón actualizar de la categoría que se quiere modificar. • Llenar el campo con el nombre de la categoría. • Dar clic en el botón actualizar. <p>Para eliminar categorías el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En el lado superior izquierdo dar clic en crear nuevo producto. • Dentro del formulario para crear nuevo producto dar clic en el botón “Categorías”. • Dar clic en el botón eliminar de la categoría que se quiere modificar. <p>Para visualizar categorías el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En el lado superior izquierdo dar clic en crear nuevo producto. • Dentro del formulario para crear nuevo producto dar clic en el botón “Categorías”. 	
Resultado deseado:	

El componente frontend admite la visualización, creación, modificación y eliminación de las categorías del perfil administrador.

Evaluación de la prueba:

El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.

Tabla XVIII Prueba de aceptación – gestionar productos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA005	Identificador de historia de Usuario: HU005
Nombre: Gestionar Productos	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Registrar productos • Modificar productos • Eliminar productos • Visualizar productos 	
<p>Pasos de ejecución:</p> <p>Para registrar productos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • En el lado superior izquierdo dar clic en crear nuevo producto. • Llenar el formulario con los datos del producto. • Dar clic en el botón crear nuevo producto. <p>Para modificar productos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Dar clic en el botón de actualizar del producto que desea modificar. • Llenar el formulario con los datos del producto. • Dar clic en el botón actualizar. <p>Para eliminar productos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Dar clic en el botón eliminar del producto que se desee eliminar. <p>Para visualizar los productos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Visualizar los productos. 	

<p>Resultado deseado:</p> <p>El componente frontend admite la visualización, creación, modificación y eliminación de los productos del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.</p>

Tabla XIX Prueba de aceptación – gestionar pedidos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA006	Identificador de historia de Usuario: HU006
Nombre: Gestionar Pedidos	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar pedidos • Listar pedidos • Cambiar el estado del pedido 	
<p>Pasos de ejecución:</p> <p>Para visualizar pedidos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • Visualizar los pedidos. <p>Para listar los pedidos el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • En la parte superior podrá encontrar tabs con los estados de los pedidos para listar. <p>Para cambiar el estado del pedido el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • Dar clic en el botón con un ojo “detalles del pedido”. • Dar clic en el botón de actualizar estado. • Seleccionar el estado del pedido que quiera cambiar. • Escribir el motivo del estado. 	

<ul style="list-style-type: none"> • Dar clic en actualizar.
<p>Resultado deseado:</p> <p>El componente frontend admite la visualización, listar y modificación de los estados del pedido del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.</p>

Tabla XX Prueba de aceptación – gestionar comentarios.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA007	Identificador de historia de Usuario: HU007
Nombre: Gestionar comentarios	
<p>Descripción: El usuario administrador en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar comentarios y/o sugerencias • Eliminar comentarios y/o sugerencias 	
<p>Pasos de ejecución:</p> <p>Para visualizar comentarios y/o sugerencias del producto el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Dar clic en el botón “ver” del producto que quiera visualizar los comentarios y/o sugerencias. <p>Para eliminar comentarios y/o sugerencias del producto el usuario administrador deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Dar clic en el botón de actualizar del producto que desea modificar. • Dar clic en el botón “ver” del producto que quiera visualizar los comentarios y/o sugerencias. • Dar clic en el botón con una “X” del comentario que desee eliminar. 	
<p>Resultado deseado:</p> <p>El componente frontend admite la visualización y eliminación de los comentarios y/o sugerencias del perfil administrador.</p>	
<p>Evaluación de la prueba:</p> <p>El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.</p>	

Tabla XXI Prueba de aceptación – visualizar categorías.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA008	Identificador de historia de Usuario: HU008
Nombre: Visualizar categorías	
Descripción: El usuario cliente en el frontend debe consumir varios endpoints para: <ul style="list-style-type: none"> • Visualizar las categorías 	
Pasos de ejecución: Para visualizar las categorías el usuario cliente deberá: <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Visualizar las categorías y filtrar los productos por categorías haciendo clic en alguna de las categorías. 	
Resultado deseado: El componente frontend admite la visualización de las categorías del perfil cliente.	
Evaluación de la prueba: El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.	

Tabla XXII Prueba de aceptación – visualizar productos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA009	Identificador de historia de Usuario: HU009
Nombre: Visualizar productos	
Descripción: El usuario cliente en el frontend debe consumir varios endpoints para: <ul style="list-style-type: none"> • Visualizar los productos 	
Pasos de ejecución: Para visualizar los productos el usuario cliente deberá: <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Visualizar los productos disponibles. 	

Resultado deseado:

El componente frontend admite la visualización de los productos del perfil cliente.

Evaluación de la prueba:

El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.

Tabla XXIII Prueba de aceptación – gestionar carrito de compras.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA010	Identificador de historia de Usuario: HU010
Nombre: Gestionar carrito de compras	
<p>Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Agregar productos al carrito de compras. • Visualizar productos agregados al carrito de compra. • Eliminar productos del carrito de compras. • Incrementar o disminuir la cantidad del producto. • Visualizar el precio de cada producto y el valor total a pagar. • Realizar la compra mediante transferencia o deposito. 	
<p>Pasos de ejecución:</p> <p>Para agregar productos al carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Seleccionar la talla, color del producto en caso de que el producto no cuente con talla o color no se selecciona. • Dar clic en el botón de agregar al carrito y se visualizara que el contador del carrito en la parte superior derecha incremento en 1 en caso de intentar agregar de nuevo el mismo producto se incrementara la cantidad del producto. <p>Para visualizar los productos del carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En la parte superior derecha dar clic en el botón con un icono de carrito. • Si el carrito cuenta con productos podrá observar los productos agregados al carrito caso contrario si no se ha agregado productos se mostrará el carrito vacío. <p>Para eliminar productos del carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. 	

<ul style="list-style-type: none"> • En la parte superior derecha dar clic en el botón con un icono de carrito. • Dar clic en el botón de eliminar del producto que quiera eliminar del carrito. <p>Para incrementar o disminuir la cantidad del producto en el carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En la parte superior derecha dar clic en el botón con un icono de carrito. • Dar clic en el botón “+” para incrementar la cantidad del producto o dar clic en el botón “-” para disminuir la cantidad del producto. <p>Para visualizar el precio de cada producto y el valor total a pagar en el carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En la parte superior derecha dar clic en el botón con un icono de carrito. • Visualizar en la parte derecha un detalle de la compra con el nombre de los productos, su precio individual, la cantidad y el valor total a pagar. <p>Para realizar la compra en el carrito de compras el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En la parte superior derecha dar clic en el botón con un icono de carrito. • En la parte derecha del detalle de compra dar clic en el botón “Realizar pedido”. • Llenar los datos del formulario incluyendo la imagen de la transferencia o deposito. • Dar clic en el botón “Realizar pedido”.
<p>Resultado deseado:</p> <p>El componente frontend admite la agregación, visualización, eliminación, incremento o decremento, visualización del detalle de compra y realización de la compra del carrito de compras del perfil cliente.</p>
<p>Evaluación de la prueba:</p> <p>El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.</p>

Tabla XXIV Prueba de aceptación – visualizar pedidos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA011	Identificador de historia de Usuario: HU011
Nombre: Visualizar pedidos	

<p>Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar pedidos • Listar pedidos • Actualizar datos básicos del pedido
<p>Pasos de ejecución:</p> <p>Para visualizar pedidos el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • Visualizar los pedidos. <p>Para listar los pedidos el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • En la parte superior podrá encontrar tabs con los estados de los pedidos para listar. <p>Para actualizar datos básicos del pedido el usuario cliente deberá:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • Ingresar en su perfil. • Dar clic en el botón de pedidos. • Dar clic en el botón con un ojo “detalles del pedido”. • Dar clic en el botón de actualizar (este botón solo estará disponible si el estado del pedido se encuentra en revisión caso contrario no se encontrar disponible). • Llenar los datos del formulario. • Dar clic en actualizar.
<p>Resultado deseado:</p> <p>El componente frontend admite la visualización, listar y modificación de datos básicos del pedido del perfil cliente.</p>
<p>Evaluación de la prueba:</p> <p>El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.</p>

Tabla XXV Prueba de aceptación – gestionar calificación del producto.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA012	Identificador de historia de Usuario: HU012
Nombre: Gestionar calificación del producto	

Descripción: El usuario cliente en el frontend debe consumir varios endpoints para:

- Visualizar comentarios y/o sugerencias
- Crear comentarios y/o sugerencias
- Actualizar comentarios y/o sugerencias

Pasos de ejecución:

Para visualizar comentarios y/o sugerencias del producto el usuario cliente deberá:

- Ingresar a la URL del frontend en un navegador.
- Haber iniciado sesión en el componente frontend.
- Dar clic en el botón “ver” del producto que quiera visualizar los comentarios y/o sugerencias.

Para crear comentarios y/o sugerencias del producto el usuario cliente deberá:

- Haber iniciado sesión en el componente frontend.
- Dar clic en el botón “ver” del producto que quiera visualizar los comentarios y/o sugerencias.
- Asegurarse de haber realizado una compra del producto y que este se encuentre finalizada para poder comentar.
- Seleccionar la calificación del producto y llenar el campo con el comentario que desee hacer al producto.
- Dar clic en el botón “Crear comentario”.

Para actualizar comentarios y/o sugerencias del producto el usuario cliente deberá:

- Haber iniciado sesión en el componente frontend.
- Dar clic en el botón “ver” del producto que quiera visualizar los comentarios y/o sugerencias.
- Seleccionar la calificación del producto y llenar el campo con el comentario que desee hacer al producto.
- Dar clic en el botón “Actualizar comentario”.

Resultado deseado:

El componente frontend admite la visualización, creación y actualización de los comentarios y/o sugerencias del perfil cliente.

Evaluación de la prueba:

El resultado cumple con las expectativas, y el cliente expresa su plena satisfacción al 100%.

Certificado que confirma el cumplimiento de todos los requisitos y funcionalidades

SATURNINA

Quito, 26 de enero de 2024

CERTIFICADO

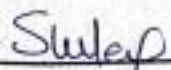
A quien interese

Yo, **Shirley Stefania Rivadeneira Armas** cédula de ciudadanía **1718299603** como dueña de "Saturnina", certifico:

Que el Sr, **Miguel Angel Carapaz Mendoza** con cédula de ciudadanía **1726527755**, estudiante de la Carrera de Tecnología Superior en Desarrollo de Software, procedió a desarrollar e implementar un sistema web para el negocio mencionado anteriormente, mismo que cumple con todos los requerimientos y funcionalidades definidas en las reuniones mantenidas.

Es todo cuanto puedo certificar en honor a la verdad, el interesado puede hacer uso de este como bien requiera.

Atentamente,



Shirley Stefania Rivadeneira Armas

ANEXO III

A continuación, para visualizar el Manual de Usuario del frontend se debe dirigir a la siguiente URL:

<https://youtu.be/TfCpxfe0XMk?si=szProd-SgAjKVUus>

En donde se explica de forma clara y sencilla las diversas funcionalidades del frontend, así como cada uno de los perfiles que forman parte de este componente.

ANEXO IV

A continuación, se presenta las credenciales de acceso del frontend, además del repositorio de GitHub, donde se encuentra todo el código y los pasos a seguir para su instalación en el apartado del README.

Enlace de acceso

Para acceder al frontend en su versión de producción, se usa la URL:

<https://saturnina.vercel.app/>

Credenciales del perfil administrador:

- Correo del administrador:
- Contraseña:

Credenciales para el perfil cliente:

- Correo de cliente:
- Contraseña:

Repositorio del frontend

El proyecto está alojado en un repositorio de GitHub, el cual se puede acceder a través de la siguiente URL:

<https://github.com/MiguelCarapaz/SaturninaFrontend.git>