



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

SISTEMA DE CONTROL DE VENTAS E INVENTARIO PARA LA EMPRESA

CAMI BORDADOS

Proyecto de titulación previo a la obtención del título de: Tecnología Superior En

Desarrollo De Software

Autor: Esteban David Arteaga Pazmiño, Elkin Guamán y Cristopher Cifuentes

Tutor: Carlos Miguel Cárdenas Riofrio

Quito, Ecuador

2025

Dedicatoria

Dedicamos este trabajo a nuestras familias, quienes nos han brindado apoyo incondicional y fortaleza durante todo el proceso de formación académica. A quienes han creído en nuestras capacidades, impulsándonos a continuar incluso en los momentos más desafiantes. También lo dedicamos a todas las personas que valoran el esfuerzo, la dedicación y la perseverancia, y que inspiran a alcanzar las metas con compromiso y responsabilidad.

Tabla de contenidos

Dedicatoria	2
Lista de figuras	4
Agradecimientos	8
Introducción	9
Capítulo I.....	10
Levantamiento de Requisitos y Diseño del Sistema	10
Capítulo II	16
Construcción del Sistema	16
Capítulo III.....	21
Pruebas y Estabilización.....	21
Conclusiones	23
Recomendaciones.....	23
Referencias bibliográficas.....	25
Anexos	26

Lista de figuras

1. Diagrama entidad-relación de la base de datos
2. Ventana de login
3. Ventana de inicio de administrador
4. Ventana de inicio de vendedor
5. Ventana de ventas
6. Ventana de pagos
7. Ventana de inventario (solo para el administrador)
8. Ventana de clientes
9. Ventana de registros (solo para el administrador)
10. Prueba de inicio de sesión con datos incorrectos
11. Prueba de registro de cliente repetido
12. Prueba de alerta por bajo stock

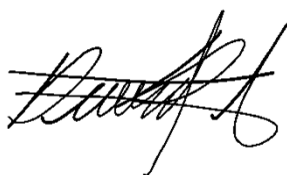
DECLARACIÓN y AUTORIZACIÓN

Yo, Esteban David Arteaga Pazmiño con C.I. 2300815392 autor(a) del trabajo de titulación intitulado: “Sistema de control de ventas e inventario para la empresa Cami Bordados”, previa a la obtención del título de Tecnólogo en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 12 de agosto de 2025



Esteban David Arteaga Pazmiño

C.I. 2300815392

DECLARACIÓN y AUTORIZACIÓN

Yo, Elkin Devison Guamán Cedeño con C.I. 1751126697 autor(a) del trabajo de titulación intitulado: “Sistema de control de ventas e inventario para la empresa Cami Bordados”, previa a la obtención del título de Tecnólogo en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 12 de agosto de 2025



Elkin Devison Guamán Cedeño

C.I. 1751126697

DECLARACIÓN y AUTORIZACIÓN

Yo, Cristopher Fabian Cifuentes Nolivos con C.I. 1751249515 autor(a) del trabajo de titulación intitulado: “Sistema de control de ventas e inventario para la empresa Cami Bordados”, previa a la obtención del título de Tecnólogo en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 12 de agosto de 2025



Cristopher Fabian Cifuentes Nolivos

C.I. 1751249515

Agradecimientos

Nosotros, Elkin Guamán, Esteban Arteaga y Cristopher Cifuentes, expresamos nuestro más profundo agradecimiento a la Pontificia Universidad Católica del Ecuador – PUCE TEC, por proporcionarnos las bases académicas y técnicas necesarias para desarrollar este proyecto.

A nuestro tutor, Carlos Miguel Cárdenas Riofrio, por su orientación constante, sus valiosos aportes y por guiarnos con paciencia y profesionalismo en cada etapa del trabajo.

A la propietaria de Cami Bordados, por su colaboración y apertura durante el levantamiento de requisitos, aportando información clave para que el sistema respondiera a las necesidades reales del negocio.

Asimismo, agradecemos a nuestros compañeros y docentes, quienes compartieron sus conocimientos y experiencias, y a nuestras familias, que con su apoyo moral y motivación hicieron posible la culminación de este proyecto.

Introducción

El presente proyecto tiene como objetivo el diseño e implementación de un sistema web de control de ventas e inventario para la empresa Cami Bordados, una microempresa dedicada a la elaboración de bordados personalizados. La necesidad de esta solución surgió a partir de las limitaciones de los procesos manuales utilizados, los cuales generaban demoras en el registro de operaciones, dificultad en el seguimiento de pagos y poca precisión en el control de insumos.

El sistema propuesto fue desarrollado bajo una arquitectura modular utilizando React.js para el frontend, Node.js con Express para el backend, Prisma como ORM y SQLite como gestor de base de datos. Esta combinación de tecnologías modernas garantiza un rendimiento óptimo, facilidad de mantenimiento y escalabilidad a mediano plazo.

La metodología de trabajo adoptada fue Kanban, lo que permitió una gestión visual y flexible de las tareas, favoreciendo la colaboración del equipo y la retroalimentación constante con la propietaria de la empresa. El resultado es una plataforma intuitiva, accesible desde cualquier dispositivo con conexión a internet, que integra funciones como gestión de clientes, registro de ventas y pagos, control de inventario, administración de usuarios, y generación de reportes y auditorías.

Este documento describe de manera detallada el proceso de desarrollo del sistema, desde el levantamiento de requisitos y diseño, hasta la construcción, pruebas y recomendaciones para su uso y mejora continua. Asimismo, se exponen las conclusiones obtenidas tras la implementación, destacando los beneficios alcanzados y las proyecciones a futuro para la evolución de la herramienta.

Capítulo I

Levantamiento de Requisitos y Diseño del Sistema

1.1 Análisis de Requisitos

Los siguientes requisitos de desarrollo se definieron como producto de la identificación de necesidades clave expresadas por la parte receptora del sistema. Se tomó como objetivo principal el cumplimiento de estas necesidades de forma más completa y eficiente posible, buscando maximizar la satisfacción del cliente con el producto final.

1.1.1 Requisitos Funcionales

Estos objetivos se rigen por las necesidades de la propietaria de Cami Bordados, quien será la principal ocupante de la plataforma. Las siguientes funcionalidades deberán ser desarrolladas para el primer despliegue en producción de la página:

- **Gestión de clientes**
 - Permitir registrar, consultar, editar y eliminar información de los clientes.
 - Cada cliente será identificado por nombre, teléfono y correo electrónico.
- **Gestión de ventas**
 - Registrar ventas concretadas por el negocio. Cada venta contendrá la información pertinente a esta y estará asociada a un cliente.
 - Registrar pagos realizados por los clientes, los cuales pueden cubrir una o varias ventas.
 - El sistema debe reflejar el estado de las ventas agrupadas por cliente (pagado/pendiente).
- **Control de inventario**

- Permitir el registro de insumos de manera flexible según sus características.
- Permitir el registro de cambios en la cantidad de insumos disponibles, e incluir un sistema configurable de alertas de stock bajo para cada registro.
- **Gestión y autenticación de usuarios**
 - Dividir usuarios por roles predefinidos los cuales definirán el acceso permitido a las funcionalidades del sistema, y permitir también la creación de nuevos roles por parte del administrador.
 - El inicio de sesión se maneja por usuario y contraseña
 - El administrador puede registrar nuevos vendedores por usuario y correo. Al vendedor le debe llegar un correo con una contraseña temporal, la cual deberá ser cambiada luego del primer inicio de sesión.
 - Permitir la eliminación de vendedores por parte del administrador.
- **Reportes y auditoría**
 - Generar automáticamente reportes de cambios en las tablas de la base de datos del sistema, registrando el cambio, así como el usuario quien lo realizó. Estos reportes serán accesibles solo para el administrador del sistema.

1.1.2 Requisitos No Funcionales

Los siguientes puntos no fueron completamente requeridos por el cliente, sino que fueron identificados por el equipo de desarrollo según las necesidades expuestas en los requisitos funcionales. Estos requisitos tienen como finalidad mejorar la experiencia de

usuario, así como establecer las expectativas de rendimiento del sistema, teniendo en cuenta el uso previsto a corto y medio plazo.

- **Búsqueda y organización de registros**
 - Incluir un sistema de búsqueda robusto para cada sección del sistema que trabaje con una lista de registros en la base de datos, así como un sistema de organización que permita mostrar los registros agrupados por distintos campos.
- **Escalabilidad**
 - Soportar una concurrencia de 20 usuarios simultáneos, para garantizar la escalabilidad del sistema a medio plazo.
- **Seguridad**
 - Almacenar los datos sensibles, tanto de los clientes como de los usuarios, de manera encriptada.
- **Disponibilidad**
 - El sistema estará disponible 24/7 desde cualquier dispositivo, salvo en casos específicos de mantenimiento.
- **Facilidad de uso**
 - Diseñar la interfaz de usuario para ser intuitiva y fácil de usar para cualquier persona no técnica que haya leído el manual de usuario.

1.1.3 Usuarios

Se identificaron dos roles de usuario iniciales necesarios para el funcionamiento de la plataforma web. Estos roles están sujetos a cambios luego del despliegue del sistema según las necesidades del administrador.

- **Administrador**

- Tiene acceso a todas las funcionalidades definidas del sistema, incluida la creación, modificación y eliminación de roles y usuarios, así como la generación de nuevas cuentas de administrador.

- La cuenta de administrador inicial, así como su rol asociado no podrá ser borrada ni modificada en ninguna circunstancia.

- **Vendedor**

- Puede acceder solamente a los módulos de ventas, pagos e inventario, y dentro de estos está limitado a las funciones de crear, exceptuando el módulo de inventario donde podrán modificar la cantidad disponible de insumos.

1.1.3 Limitaciones

- La base de datos utilizada (SQLite) está optimizada para entornos con volúmenes moderados de datos y usuarios; para una alta concurrencia podría requerirse migrar a un gestor más robusto.

- No se implementó un sistema de notificaciones en tiempo real; las alertas se generan solo al momento de acceder a la sección correspondiente.

- El sistema está diseñado para uso web, no incluye aplicación móvil nativa.

- La gestión de inventario está orientada a insumos físicos y no contempla productos digitales o virtuales.

1.2 Diseño del sistema

El enfoque de diseño se centra en la usabilidad y en la resolución de los principales problemas del sistema anterior que este reemplaza.

1.2.1 Interfaz de usuario

Los pilares establecidos para el diseño de la interfaz de usuario fueron la usabilidad y eficiencia en el flujo de trabajo del usuario. Teniendo en cuenta esto y los requisitos funcionales y no funcionales previamente establecidos, se consideraron los siguientes puntos clave para su diseño:

- **Uso de elementos visuales:** Se utilizan principios de diseño que buscan maximizar la cantidad de información transmitida visualmente, como la implementación de un diseño claro en toda la página y el uso de íconos y colores en lugar de texto donde sea posible.
- **Diseño consistente:** Para la estructura visual general de la página se tomó como base una plantilla compatible con React, con el objetivo de dinamizar el proceso de diseño y asegurar la consistencia entre los diferentes módulos de la página.
- **Diseño intuitivo:** El uso de una nomenclatura clara y el principio de consistencia previamente establecido dan como resultado un diseño fácil de usar y entender para el usuario final. Los elementos visuales se distribuyeron de manera lógica para maximizar el entendimiento sin necesidad de familiarizarse previamente con la plataforma.

1.2.2 Arquitectura del sistema

Los principales aspectos que se tomaron en cuenta para el diseño de la arquitectura del sistema fueron escalabilidad, eficiencia y rendimiento. Por lo tanto, se eligió un sistema de diseño modular, separando la aplicación en los siguientes elementos:

- **Frontend:** Todo lo relacionado a los aspectos con los que el cliente interactúa directamente. Se eligió el marco de trabajo React por la extensa cantidad de herramientas disponibles para facilitar su uso, así como su potencia para desarrollar una interfaz visual tanto funcional como llamativa.

- **Backend:** Los servicios en segundo plano; donde se maneja la lógica del funcionamiento de la página. El usuario no puede interactuar con este módulo directamente. El marco de trabajo utilizado para su desarrollo fue Express.js, por su compatibilidad técnica con el proyecto, su rendimiento y capacidad para ser escalable.
- **Base de datos:** Donde se almacenan los datos utilizados por la plataforma e ingresados por los usuarios. El motor de base de datos escogido fue SQLite, por su ligereza y rendimiento, especialmente para el caso de uso de este proyecto.

1.2.3 Características clave

Las siguientes funciones son consideradas esenciales para el correcto cumplimiento de las necesidades de los usuarios del sistema:

- **Autenticación inicial por correo:** Permite al administrador tener control sobre los usuarios con acceso al sistema manteniendo la autonomía de estos luego del primer paso de autenticación.
- **Pagos de múltiples ventas:** Permite cubrir varias ventas con un pago único, respondiendo a las necesidades reales del negocio y adaptándose al modo de uso del sistema anterior.
- **Control de insumos independiente de ventas:** Especialmente útil al ser para el caso de uso de servicios producción manual, ya que no venden directamente los insumos disponibles, sino los bordados específicos solicitados.
- **Estructura modular:** Al separar el sistema en los módulos descritos en el punto de arquitectura del sistema, se mantienen los principios de escalabilidad y facilidad tanto de desarrollo y de mantenimiento, permitiendo expandir y mantener el código de un módulo independientemente del resto.

Capítulo II

Construcción del Sistema

En este capítulo se explica cómo se fue construyendo el sistema “Cami Bordados”, desde las herramientas que usamos hasta cómo se armó cada parte del proyecto.

2.1 Herramientas que utilizamos

Para este proyecto decidimos trabajar con tecnologías actuales y bastante comunes en desarrollo web, ya que muchas ya las habíamos visto durante la carrera y nos resultaban familiares:

- **Frontend:** React.js con Vite, porque carga rápido y es fácil de configurar.
- **Backend:** Node.js usando Express.js como framework para las rutas y la lógica del servidor.
- **Base de datos:** SQLite, por ser liviano y no requerir instalaciones complicadas.
- **ORM:** Prisma, para trabajar más fácil con la base de datos desde Node.
- **Control de versiones:** Git, con repositorio local durante todo el desarrollo.
- **Gestión de dependencias:** npm.

Estas herramientas nos ayudaron a mantener el proyecto organizado y fácil de modificar.

2.2 Organización del proyecto

Dividimos el proyecto en tres carpetas principales para que todo esté bien separado y más fácil de ubicar:

- **Frontend:** Dentro de Frontend/src está el código del cliente. Ahí van los componentes de React, la conexión con el backend y los estilos.
- **Backend:** En esta parte están los controladores (controllers/), la configuración de Express (index.js) y los modelos de Prisma en prisma/.
- **Base de datos:** Guardamos aquí el archivo .db de SQLite y un script SQL con la estructura.

Este orden nos permitió trabajar de forma modular, sin mezclar responsabilidades.

2.3 Interfaz del sistema

Trabajamos la parte visual usando React con Vite, lo que facilitó el trabajo por su velocidad y simplicidad. Nos enfocamos en crear componentes reutilizables, lo que ahorró bastante tiempo. Algunos de los módulos más importantes que desarrollamos fueron:

- **Cientes:** Para registrar, editar o eliminar datos de los clientes.
- **Ventas y pagos:** Permite registrar ventas asociadas a cada cliente y controlar si ya se pagaron o no.
- **Inventario:** Módulo para registrar insumos y llevar el control de stock.
- **Usuarios:** Pantallas de inicio de sesión y cambio de contraseña.

También pusimos formularios con validaciones básicas y mensajes que confirman si algo salió bien o si hubo un error.

2.4 Lógica del sistema

El backend se encargó de toda la lógica del sistema, conectando el frontend con la base de datos. Creamos rutas para:

- **/clientes:** Crear, editar, obtener o eliminar clientes.
- **/ventas:** Registrar ventas y consultarlas por cliente.
- **/pagos:** Asociar pagos a una o varias ventas.
- **/insumos:** Control del inventario.
- **/usuarios:** Registro y autenticación de usuarios.

Toda esta lógica se configuró desde el archivo `index.js`, donde se levantó el servidor Express y se llamaron los controladores que están en la carpeta `controllers/`.

2.5 Base de datos

Optamos por SQLite porque es muy fácil de usar en proyectos pequeños como este, y no necesita instalación de servidores. Creamos los modelos con Prisma, lo que facilitó el acceso a los datos desde Node.js sin tener que escribir muchas consultas SQL manualmente.

Las tablas principales son:

- Usuarios
- Clientes
- Ventas
- Pagos
- Insumos
- Auditorias

También configuramos relaciones entre ventas y clientes, pagos y ventas, y un historial de acciones para tener un registro de los cambios.

2.6 Diagramas del sistema

- Diagrama conceptual de la arquitectura del sistema

El sistema está compuesto por tres capas principales: el frontend desarrollado en React.js, el backend implementado con Node.js y Express, y la base de datos SQLite gestionada mediante Prisma. La comunicación entre estas capas se realiza a través de solicitudes HTTP y respuestas en formato JSON.

- Esquema de la interfaz principal de usuario

La interfaz se compone de un menú lateral para la navegación entre módulos, una barra superior con opciones rápidas y un área principal donde se despliega el contenido correspondiente a cada funcionalidad.

- Flujo de autenticación y gestión de roles

El proceso de inicio de sesión comienza con la validación de credenciales, seguida de la asignación de permisos según el rol del usuario (administrador o vendedor), controlando el acceso a los diferentes módulos del sistema.

- Modelo entidad-relación de la base de datos

La base de datos está estructurada con las tablas Usuarios, Clientes, Ventas, Pagos, Insumos y Auditorías, con relaciones definidas para garantizar integridad referencial y consistencia de datos.

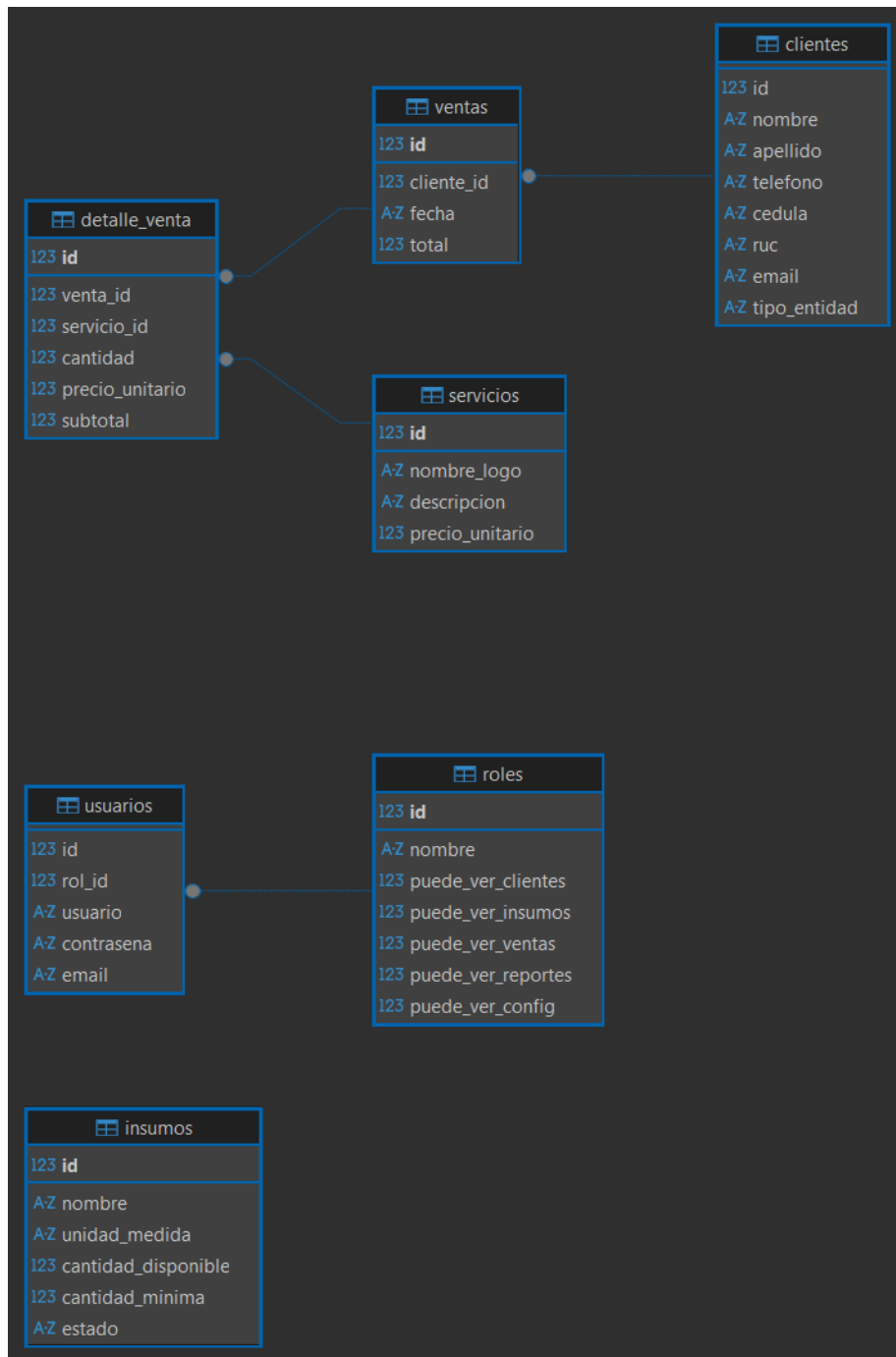


Fig. 1 - Diagrama entidad-relación de la base de datos

Capítulo III

Pruebas y Estabilización

Aquí contamos qué pruebas hicimos para asegurarnos de que todo funcionara bien antes de terminar el proyecto.

3.1 Tipos de pruebas aplicadas

A lo largo del desarrollo, fuimos haciendo distintas pruebas según la parte del sistema que estábamos trabajando:

- **Unitarias:** Probamos funciones específicas, como los cálculos de pagos.
- **Funcionales:** Verificamos que cada módulo hiciera lo que debía según los objetivos del proyecto.
- **De interfaz:** Nos fijamos en cómo se comportaban las pantallas y si la navegación era intuitiva.
- **De integración:** Confirmamos que el frontend y backend se comunicaran correctamente.
- **De usuario:** Simulamos escenarios reales, como si fuéramos los clientes o administradores del sistema.

3.2 Casos de prueba que destacaron

Algunos de los escenarios más importantes que probamos fueron:

- **Inicio de sesión con datos incorrectos:** Comprobamos que el sistema muestre un mensaje de error y no permita el acceso.
- **Registro de un cliente repetido:** Si ya existe un correo, aparece una advertencia.

- **Pagos para varias ventas:** Validamos que todas las ventas afectadas cambien su estado correctamente.
- **Bajo stock:** Al registrar cantidades menores al mínimo definido, el sistema muestra una alerta.

3.3 Resultados y ajustes finales

En general, las pruebas salieron bien. Hubo algunos errores menores al principio, como:

- Detalles visuales en pantallas pequeñas.
- Validaciones que faltaban en ciertos formularios.

Esos problemas se fueron corrigiendo en cada iteración. Al final, el sistema quedó funcionando correctamente y listo para entregarse.

Conclusiones

1. El desarrollo del sistema de control de ventas e inventario para Cami Bordados permitió centralizar y optimizar los procesos de registro, seguimiento y control de las operaciones comerciales, reduciendo significativamente el tiempo de gestión y minimizando errores humanos.
2. La implementación de tecnologías modernas como React.js, Node.js con Express, Prisma y SQLite brindó un sistema ágil, seguro y adaptable a futuras necesidades del negocio.
3. Las pruebas funcionales y de integración confirmaron que la plataforma cumple con los requisitos planteados por la empresa, ofreciendo una interfaz amigable y accesible desde cualquier dispositivo con conexión a internet.
4. El uso de un modelo modular facilita el mantenimiento, la escalabilidad y la incorporación de nuevas funcionalidades, lo que garantiza la sostenibilidad del sistema a largo plazo.
5. La interacción con el cliente final durante todas las fases del proyecto fue clave para asegurar que la solución se adaptara a los procesos reales, mejorando la experiencia de uso y la aceptación del sistema.

Recomendaciones

1. Realizar copias de seguridad periódicas de la base de datos para prevenir la pérdida de información por fallas técnicas.
2. Mantener el sistema actualizado, aplicando parches de seguridad y mejoras que vayan surgiendo en las tecnologías utilizadas.
3. Implementar, en futuras versiones, un módulo de reportes gráficos y analíticos que facilite la toma de decisiones estratégicas.

4. Capacitar al personal nuevo antes de otorgarles acceso, asegurando que conozcan las funcionalidades y buenas prácticas del sistema.
5. Considerar la migración a una base de datos más robusta en caso de que el volumen de datos y usuarios simultáneos aumente considerablemente.

Referencias bibliográficas

- React. (2024). React – A JavaScript library for building user interfaces. Meta Platforms, Inc. <https://react.dev/>
- Node.js. (2024). Node.js Documentation. OpenJS Foundation. <https://nodejs.org/>
- Express.js. (2024). Express – Fast, unopinionated, minimalist web framework for Node.js. <https://expressjs.com/>
- Prisma. (2024). Prisma Documentation. Prisma Data, Inc. <https://www.prisma.io/docs>
- SQLite. (2024). SQLite Documentation. <https://sqlite.org/docs.html>
- Pressman, R. S., & Maxim, B. R. (2021). Ingeniería de software: un enfoque práctico (9.^a ed.). McGraw-Hill.
- Sommerville, I. (2020). Ingeniería del software (10.^a ed.). Pearson Educación.

Anexos

- Manual de usuario
- Ventana de login

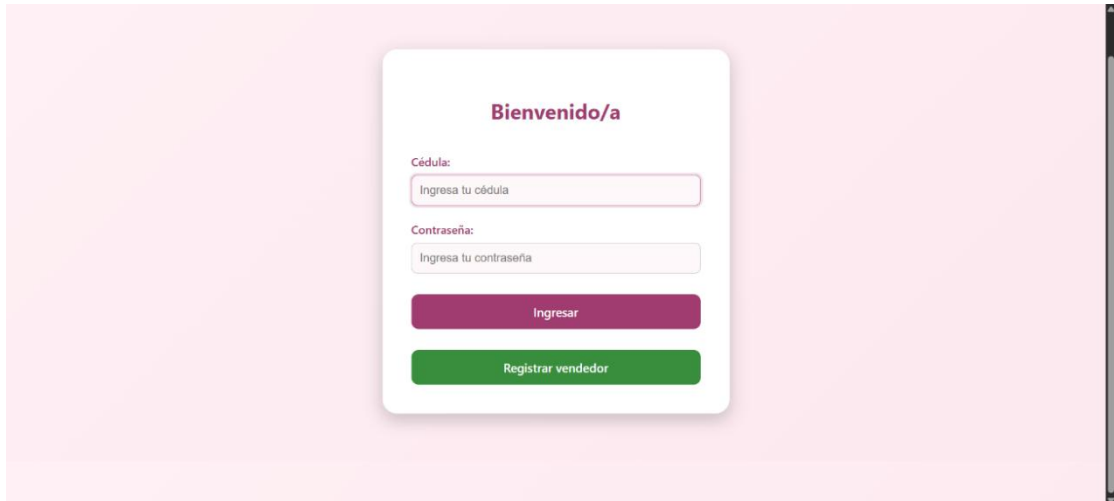


Fig. 2 - Ventana de login

- Ventana de inicio de administrador

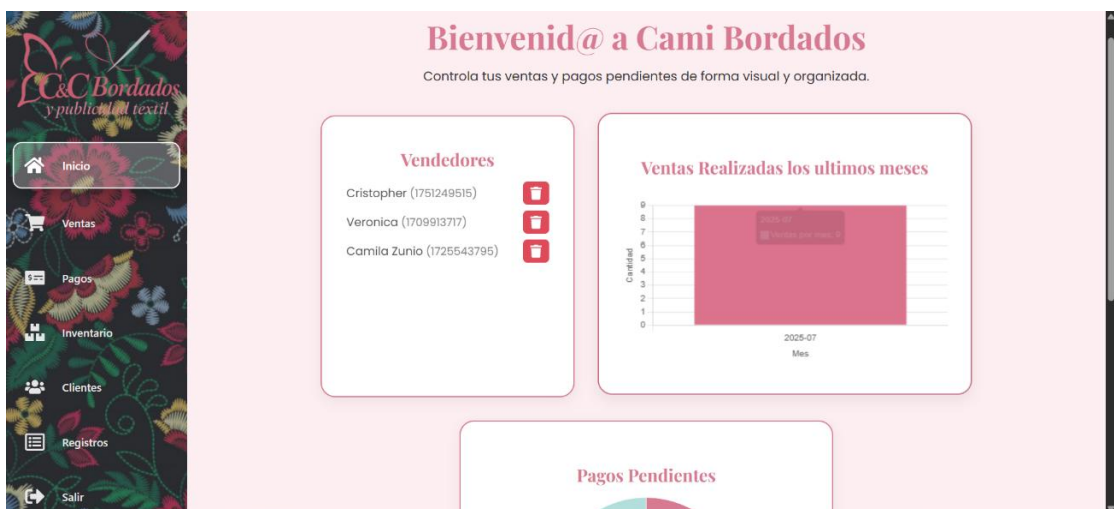


Fig. 3 - Ventana de inicio de administrador

- Ventana de inicio de vendedor

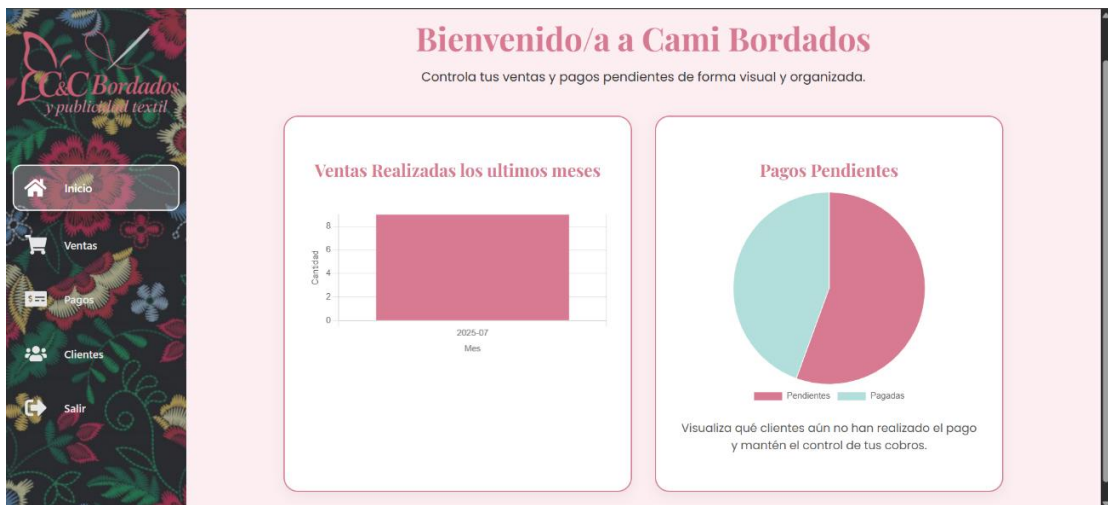


Fig. 4 - Ventana de inicio de vendedor

- Ventana de ventas

Registrar Venta

Cliente:
Selecciona o escribe el nombre del cliente...

Nombre del logo:

Cantidad:
1

Precio unitario:
\$

Precio total:
\$ 0

Fecha:

Fig. 5 - Ventana de ventas

- Ventana de pagos



Fig. 6 - Ventana de pagos

- Ventana de inventario (solo para el administrador)

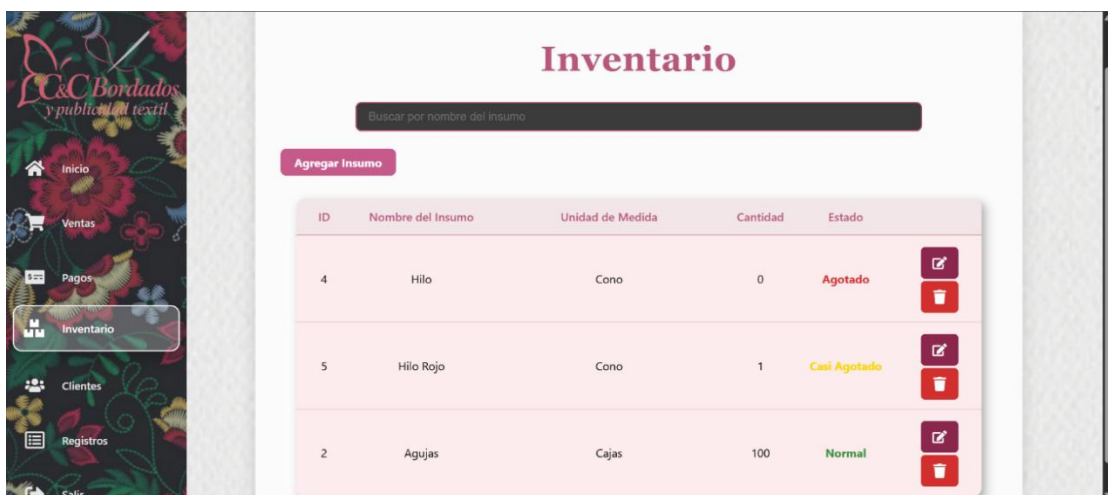


Fig. 7 - Ventana de inventario (solo para el administrador)

- Ventana de clientes

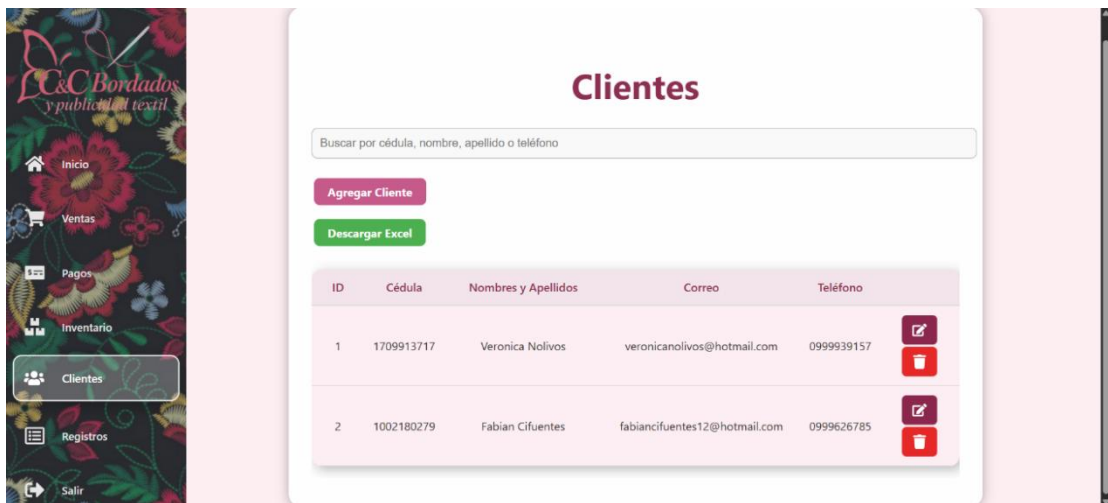


Fig. 8 - Ventana de clientes

- Ventana de registros (solo para el administrador)



Fig. 9 - Ventana de registros (solo para el administrador)

- Prueba de inicio de sesión con datos incorrectos

Bienvenido/a

Usuario o contraseña incorrectos.

Cédula:

1751249515

Contraseña:

.....

Ingresar

Registrar vendedor

Fig. 10 - Prueba de inicio de sesión con datos incorrectos

- Prueba de registro de cliente repetido

El correo ya está registrado.

Aceptar

Fig. 11 - Prueba de registro de cliente repetido

- Prueba de alerta por bajo stock







ID	Nombre del Insumo	Unidad de Medida	Cantidad	Estado	
4	Hilo	Cono	0	Agotado	 
5	Hilo Rojo	Cono	1	Casi Agotado	 
2	Agujas	Cajas	100	Normal	 

Fig. 12 - Prueba de alerta por bajo stock