

Taller 01: Requerimientos

Sebastian Andrade Cedano
Cristian Steven Motta Ojeda
Juan Esteban Santacruz Corredor

Ingeniería de software I

Universidad Nacional de Colombia

Bogotá D.C.
2024

Índice

Índice.....	2
1. Conversación seleccionada.....	3
2. Levantamiento de requerimientos.....	3
3. Análisis de requerimientos.....	7
4. Caso de uso.....	11
5. Historias de usuario.....	14
5.1. Historia de usuario #1.....	14
5.2. Historia de usuario #2.....	17
5.3. Historia de usuario #3.....	21

1. Conversación seleccionada

Teniendo en cuenta que es recomendable seleccionar la conversación que aporta mayor cantidad de información, para el presente trabajo, se utilizará la conversación del “GP 2”.

2. Levantamiento de requerimientos

Durante una entrevista realizada, la cliente expuso diversas necesidades relacionadas con la gestión de su negocio de ventas de artesanías y accesorios decorativos. Actualmente, sus procesos son manuales, lo que ocasiona problemas de organización y eficiencia. Por lo tanto, busca una solución que centralice y automatice las actividades clave, garantizando una gestión más profesional y eficiente, sin perder la accesibilidad ni la facilidad de uso.

Contexto y Problemas del Negocio

La cliente administra un pequeño negocio que combina la venta de productos hechos a mano (artesanías) con la reventa de artículos decorativos adquiridos a terceros. Aunque ha considerado ofrecer asesorías de decoración, esta iniciativa no es una prioridad por el momento. Actualmente, utiliza herramientas como WhatsApp y Excel para manejar inventarios, pedidos y ventas.

Los principales problemas identificados son:

- Desorganización en la gestión de pedidos y pagos.
- Dificultad para mantener actualizado el inventario.
- Ausencia de un catálogo centralizado para los clientes.
- Procesos manuales que consumen tiempo y generan errores.
- Falta de herramientas para promocionar productos y atraer nuevos clientes.

Solución propuesta

Se propone desarrollar una aplicación web intuitiva y fácil de usar que centralice las operaciones esenciales del negocio, satisfaciendo las necesidades identificadas durante la entrevista. Esta solución incluirá funcionalidades para la gestión de usuarios, pedidos, pagos, inventario, catálogo, clientes y finanzas, además de garantizar una experiencia accesible y segura.

Requerimientos del sistema

Funcionales:

1. Autenticación y Roles de Usuario
 - a. Registro de usuarios con roles definidos (Administrador o Cliente).
 - b. Inicio de sesión seguro mediante correo electrónico y contraseña.
 - c. Interfaces personalizadas según el rol del usuario.
 - d. Registro rápido para clientes mediante Google o redes sociales (opcional).
2. Gestión de Pedidos y Pagos

- a. Registro y seguimiento de pedidos con información detallada (cliente, productos, estado).
 - b. Actualización automática del estado del pedido (Pendiente, Pagado, Enviado, Completado).
 - c. Notificaciones automáticas sobre el progreso del pedido, preferiblemente mediante WhatsApp.
 - d. Integración con pasarelas de pago locales como Nequi y Daviplata.
 - e. Confirmación automática de pagos y registro de transacciones.
3. Gestión de Inventario y Catálogo
- a. Actualización automática del inventario tras la finalización de un pedido.
 - b. Registro y seguimiento del stock por producto.
 - c. Catálogo en línea con visualización de productos, descripciones, precios y disponibilidad.
 - d. Opciones para personalizar productos.
 - e. Filtros de búsqueda por categorías.
4. Gestión de Clientes y Finanzas
- a. Almacenamiento de datos de clientes (nombre, contacto, historial de pedidos).
 - b. Registro automático de nuevos clientes al realizar un pedido.
 - c. Edición de información de clientes.
 - d. Registro de ingresos y egresos relacionados con el negocio (opcional).
 - e. Resumen visual de ventas y ganancias netas.
5. Notificaciones Automáticas
- a. Alertas al administrador sobre productos con bajo stock, preferiblemente vía WhatsApp.
 - b. Confirmaciones de pedidos y actualizaciones para los clientes vía WhatsApp.

No funcionales:

1. Facilidad de Uso
- a. Navegación clara con menús y botones bien definidos.
 - b. Proceso sencillo para realizar pedidos y gestionar inventarios.
 - c. Capacitación mínima requerida para el administrador.
2. Accesibilidad y Rendimiento
- a. Compatibilidad con navegadores modernos en dispositivos móviles y computadoras.
 - b. Optimización para pantallas de distintos tamaños y funcionalidades offline básicas.
 - c. Tiempo de carga inferior a 3 segundos en condiciones estándar.
 - d. Capacidad para manejar al menos 100 usuarios simultáneos.

3. Seguridad y Estética

- a. Cifrado de contraseñas y protección de datos sensibles.
- b. Comunicaciones seguras mediante HTTPS.
- c. Diseño visual elegante y profesional con una paleta de colores oscuros (azules y blancos, con naranjas).
- d. Imágenes grandes y texto claro para destacar productos y promociones

3. Análisis de requerimientos

Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Información Adicional
Autenticación y Roles de Usuario	Must Have	Registro e inicio de sesión seguro con roles diferenciados.	8	Tecnologías: React, Node.js, MongoDB.
Gestión de Pedidos	Must Have	Registro, actualización y notificación del estado de los pedidos.	13	Tecnologías: React, Node.js, WhatsApp API.
Gestión de Pagos	Must Have	Integración con pasarelas de pago y registro automático de transacciones.	13	Tecnologías: Node.js, Nequi/Daviplata API.
Gestión de Inventario	Must Have	Registro y actualización automática del stock según pedidos realizados.	5	Tecnologías: MongoDB, Node.js.
Catálogo en Línea	Must Have	Visualización de productos con descripciones, fotos y filtros de categorías.	8	Tecnologías: React, CSS, Firebase Storage.

Gestión de Clientes	Should Have	Almacenamiento de datos de clientes y seguimiento de historial de compras.	8	Tecnologías: MongoDB, Node.js.
Gestión Financiera	Could Have	Registro de ingresos y egresos con resúmenes visuales.	3	Tecnologías: D3.js, MongoDB.
Notificaciones Automáticas	Must Have	Envío de alertas y actualizaciones a clientes y administradores.	5	Tecnologías: WhatsApp API, Node.js.
Facilidad de Uso	Must Have	Diseño de interfaz intuitiva y capacitación para el administrador.	2	Tecnologías: Figma, React.
Accesibilidad	Must Have	Optimización para dispositivos móviles y navegadores modernos.	3	Tecnologías: React, CSS.

Seguridad	Must Have	Implementación de cifrado, HTTPS y protección contra ataques comunes.	5	Tecnologías: Node.js, SSL, OWASP best practices.
Estética y Diseño Visual	Should Have	Diseño atractivo con paleta de colores naturales y visualización clara de productos.	5	Tecnologías: Figma, CSS.

4. Caso de uso

Gestión de pedidos por parte del Administrador

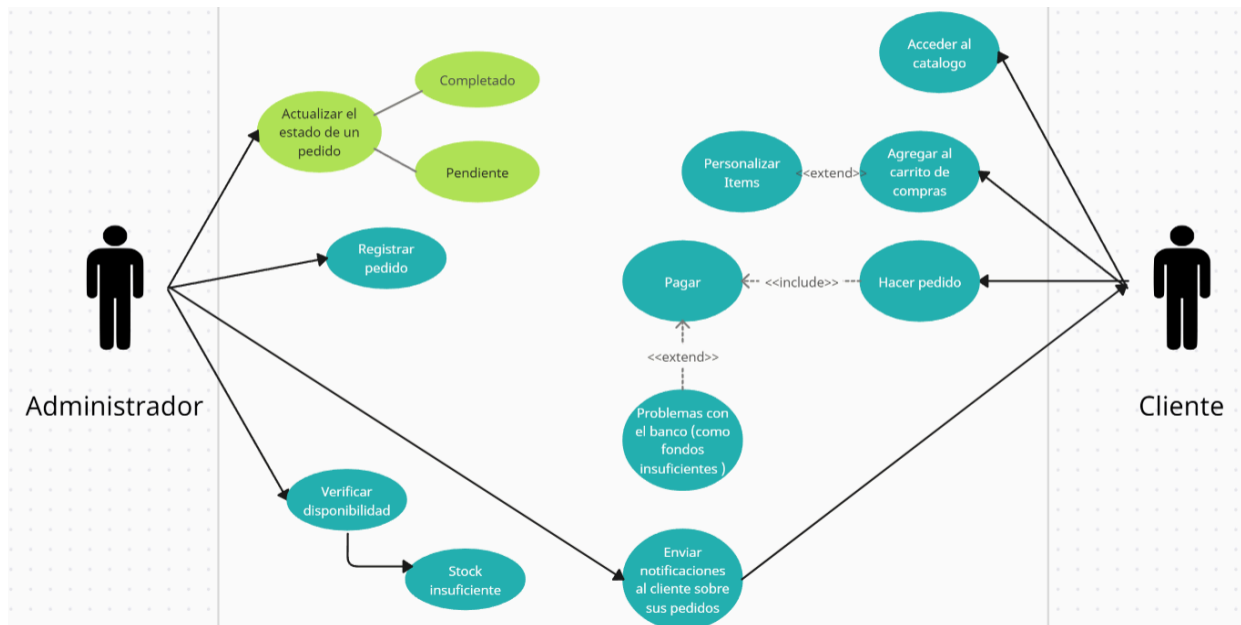
- *Actor Principal:* Administrador.
- *Precondiciones:*
 - El administrador debe haber iniciado sesión en el sistema con las credenciales adecuadas y los productos necesarios para el pedido deben estar disponibles en el inventario.
- *Flujo Principal:*
 - El administrador registra un nuevo pedido con los datos del cliente (nombre y contacto) y los detalles de los productos solicitados (nombre y cantidad).
 - Se verifica la disponibilidad del inventario. (Si el stock es suficiente, permite continuar con el registro del pedido).
 - El cliente recibe una notificación confirmando el registro del pedido.
 - A medida que el pedido progresa, el administrador puede actualizar su estado a Completado y luego se ajustará el inventario automáticamente.
- *Flujos Alternativos:*
 - Stock insuficiente: El administrador puede modificar el pedido o actualizar el inventario para proceder.
 - Error en la conexión con el servidor: Si ocurre un problema técnico al registrar el pedido, el sistema muestra: "No se pudo registrar el pedido debido a un problema técnico. Intente nuevamente más tarde." Adicionalmente el administrador verifica la conexión o contacta al soporte técnico para resolver el problema.
- *Postcondiciones:*
 - El pedido queda registrado correctamente en el sistema, asociado al cliente y con un historial detallado de los cambios realizados. El inventario del sistema se ajusta automáticamente al finalizar el pedido, el cliente recibe notificaciones sobre el estado y progreso de su pedido.

Gestión de pedidos por parte del cliente

- *Actor Principal:* Cliente.
- *Precondiciones:*
 - El cliente debe haber iniciado sesión en el sistema y los productos necesarios para el pedido deben estar disponibles en el inventario.
- *Flujo Principal:*
 - El cliente accede al catálogo en línea desde su navegador. Selecciona los productos deseados y los agrega al carrito de compras, personalizándolos si corresponde.
 - Procede a confirmar el pedido ingresando su información personal o iniciando sesión en el sistema.
 - Selecciona el método de pago preferido (Nequi o Daviplata) y hace el pago.

- Una vez enviado el pedido, el cliente recibe una notificación con los detalles del envío (incluyendo el estado "Enviado").
- *Flujos Alternativos:*
 - Fallo en el pago: El cliente corrige el problema o selecciona otro método de pago.
 - Problemas de conexión al sistema: El cliente puede esperar a que el problema sea resuelto o contactar al soporte técnico para asistencia.
- *Postcondiciones:*
 - El pedido queda registrado con todos los detalles del cliente, productos seleccionados y estado actual. El inventario del sistema se ajusta automáticamente según los productos solicitados.

Diagrama de caso de uso



5. Historias de usuario

5.1. Historia de usuario #1

Descripción conceptual

Módulo	Módulo de Login
Descripción de la(s) funcionalidad(es) requerida(s):	Permitir a los usuarios autenticarse en el sistema acorde a su rol

Descripción técnica

Backend

El backend para el sistema de inicio de sesión debe manejar de manera segura la autenticación de usuarios haciendo el manejo correcto de las credenciales de acceso. Al recibir las credenciales, debe validarse contra la base de datos y gestionar intentos fallidos. Todas las operaciones deben seguir principios de seguridad para prevenir ciberataques y la violación del sistema de acceso.

URL	Método	Código html
localhost:8080/login	POST	200 403
Caso de uso técnico Al consultar, debe validar las credenciales enviadas y en caso de ser correctas permitir al usuario iniciar sesión en el sistema		
Datos de entrada	Datos de salida	
200: { "Id": 123456, "Password": "password" }		
403: { "Id": 123456, "Password": "password" }	{ "message": "Wrong password" }	

Frontend

En la aplicación web, al ingresar a la página de inicio, el usuario encuentra una pantalla sencilla y limpia con un formulario de inicio de sesión al centro. Este formulario incluye campos para

ingresar su respectivo usuario y contraseña, acompañados de un botón que permite realizar el envío del formulario.

Si el usuario comete un error, como dejar campos vacíos o ingresar credenciales incorrectas, se muestra un mensaje amigable justo debajo del formulario, indicando el problema y ofreciendo instrucciones para solucionarlo.

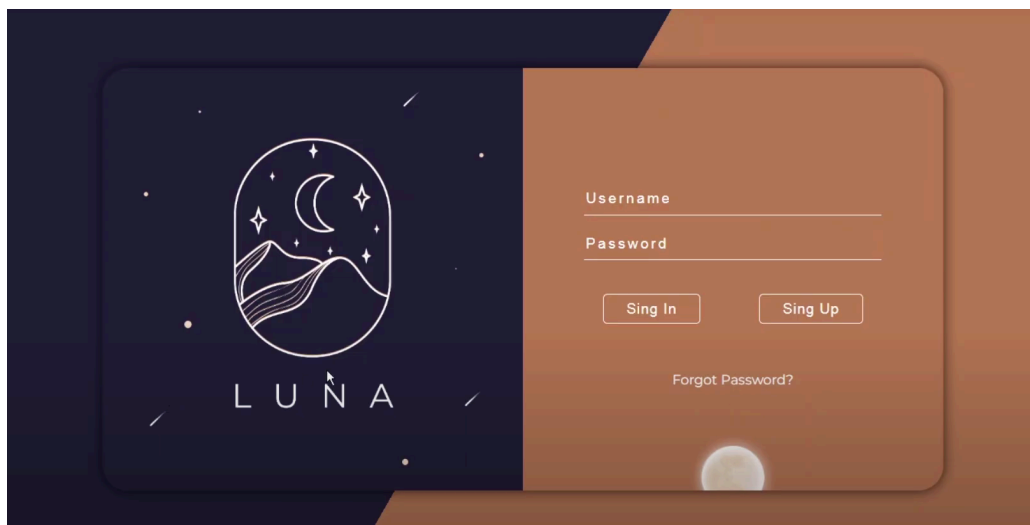
En caso de olvidar la contraseña, un enlace visible al pie del formulario lleva al usuario a una nueva pantalla donde puede ingresar su correo electrónico para recibir un enlace de restablecimiento.

Finalmente, tras un inicio de sesión exitoso, el sistema redirige automáticamente al usuario a un panel principal donde puede realizar las acciones acordes a su rol.

Interacción esperada:

Se espera que el usuario llene los campos requeridos, y acto seguido haga envío del formulario haciendo clic en el botón pensado para esto.

Mockups/Prototipos:



Flujo visual y eventos:

Una vez presionado el botón de envío del formulario, el frontend enviará los datos al backend, y mostrará una animación de carga en lo que se recibe la respuesta del servidor

5.2. Historia de usuario #2

Descripción conceptual

Módulo	Módulo de Compras para usuarios
Descripción de la(s) funcionalidad(es) requerida(s):	Permitir a los usuarios ver el catálogo de productos disponibles

Descripción técnica

Backend

El backend debe gestionar el carrito de compras de forma persistente, asociando cada carrito con el usuario autenticado. Se debe contar con un modelo de datos que almacene los productos agregados, sus cantidades y precios.

URL	Método	Código html
localhost:8080/cart/items	POST	201
Caso de uso técnico Debe permitir añadir un producto al carrito de compras, retorna el carrito actualizado		
Datos de entrada	Datos de salida	
201: { "Product_id": 12123, "Quantity": 2 }	{ "Products" : [{ "product_id": 1212, "quantity": 2 }, { "product_id": 5312, "quantity": 3 },] }	

URL	Método	Código html
localhost:8080/cart/items	DELETE	200
Caso de uso técnico		

Debe permitir eliminar un producto al carrito de compras, retorna el carrito actualizado	
Datos de entrada	Datos de salida
200: <pre>{ "Product_id": 12123, "Quantity": 2 }</pre>	<pre>{ "Products" : [{ "product_id": 5312, "quantity": 3 },] }</pre>

URL	Método	Código html
localhost:8080/cart/items	GET	200
Caso de uso técnico Debe permitir consultar los productos en el carrito		
Datos de entrada	Datos de salida	
200:	<pre>{ "Products" : [{ "product_id": 1212, "quantity": 2 }, { "product_id": 5312, "quantity": 3 },] }</pre>	

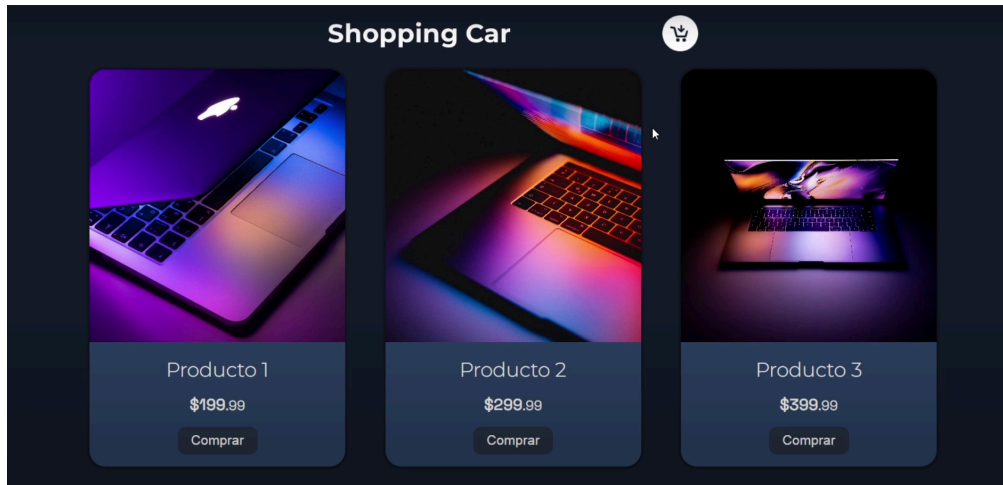
Frontend

El frontend debe ofrecer una interfaz interactiva para gestionar el carrito de compras, permitiendo agregar y eliminar productos de manera dinámica sin recargar la página. El resumen del carrito mostrará los productos y sus cantidades, considerando impuestos y descuentos si aplica. Las acciones de agregar, eliminar y actualizar productos deben realizar peticiones al backend para mantener el estado sincronizado. La interfaz debe ser clara y ofrecer un flujo de navegación sencillo entre la tienda y el carrito.

Interacción esperada:

Se espera que el usuario administre su carrito de compras a su gusto, agregando y eliminando los productos que desee.

Mockups/Prototipos:



Flujo visual y eventos:

Una vez accedido el catálogo de productos, el usuario podrá hacer clic en él y añadirlo a su carrito de compras, luego podrá consultarlo y comprarlo si lo desea.

5.3. Historia de usuario #3

Descripción conceptual

Módulo	Módulo de inventario
Descripción de la(s) funcionalidad(es) requerida(s):	Permitir el registro y la actualización automática del stock en función de los pedidos realizados.

Descripción técnica

Backend

El sistema debe gestionar el inventario de manera automática, actualizando las cantidades disponibles en tiempo real cada vez que se procesa un pedido.

URL	Método	Código html
localhost:8080/inventory/products	POST	201
Caso de uso técnico Debe permitir añadir un producto al inventario, retorna un mensaje con el estado de la operación y el id del producto añadido.		
Datos de entrada	Datos de salida	
201: { "Product_id": 12312, "Name": "vaso de vidrio", "Category": "Artesanías", "Stock": 50, "Price": 10000 }	{ "Message": "Producto registrado con éxito.", "Product_id": 101 }	

URL	Método	Código html
localhost:8080/inventory/update	PUT	200
Caso de uso técnico Reducir automáticamente la cantidad disponible en el inventario cuando se confirme un pedido, retorna un mensaje con el estado de la operación y el stock actualizado.		

Datos de entrada	Datos de salida
200: <pre>{ "Product_id": 101, "Quantity_sold": 5 }</pre>	<pre>{ "Message": "Inventario actualizado. Stock restante: 45.", "Stock": 45 }</pre>

URL	Método	Código html
localhost:8080/inventory/products	GET	200
Caso de uso técnico Debe permitir consultar el inventario a los administradores, retorna el inventario de productos con todos los atributos respectivos.		
Datos de entrada	Datos de salida	
200:	<pre>{ "Inventory": [{ "Product_id": 12312, "Name": "vaso de vidrio", "Category": "Artesanías", "Stock": 50, "Price": 10000 }, { "Product_id": 1232, "Name": "plato", "Category": "Artesanías", "Stock": 520, "Price": 10200 }] }</pre>	

Frontend

El frontend debe ofrecer una interfaz interactiva para que los administradores gestionen el inventario de productos de manera eficiente. Esta interfaz permitirá agregar, editar y eliminar productos, así como actualizar automáticamente las cantidades disponibles tras la confirmación de un pedido, sin necesidad de recargar la página. Las acciones realizadas deben sincronizarse con el backend en tiempo real para mantener la consistencia de los datos. La interfaz debe ser intuitiva, ofreciendo filtros por categorías y opciones de búsqueda rápida para facilitar la administración.

Interacción esperada:

Se espera que el administrador gestione el inventario de forma dinámica, agregando, editando o eliminando productos, y que el stock se actualice automáticamente tras los pedidos. Además, podrá buscar y filtrar productos rápidamente.

Mockups/Prototipos:

The mockup displays a web interface for product inventory management. On the left, a sidebar contains navigation links: 'Resumen', 'Categorías', 'Productos', and 'Anterior'. The main content area is divided into two sections. The top section, titled 'Nuevo producto', contains a form with the following fields: 'Nombre', 'ID', 'Categoría', 'Precio', 'Cantidad', 'Valor', 'Fecha', and 'Venta'. Below the form are two buttons: 'Descartar' and 'Agregar'. The bottom section displays a table of products with columns for 'Disponibilidad' and 'In-stock/Out of stock'. The table lists several products, including 'Maggi', 'Bru', 'Red Bull', 'Bourn Vita', 'Horlicks', 'Harpic', 'Ariel', 'Scotch Brite', and 'Coca cola'. The 'Disponibilidad' column shows the status of each product, with 'In-stock' in green and 'Out of stock' in red. A 'Siguiente' button is located at the bottom right of the table.

Disponibilidad
In-stock
Out of stock
In-stock
Out of stock
In-stock
In-stock
Out of stock
In-stock
Low stock

Flujo visual y eventos:

Una vez accedido al módulo de inventario, el administrador podrá agregar nuevos productos mediante un formulario emergente, editar información existente haciendo clic en el botón de edición dentro de la tabla de productos o eliminar artículos seleccionándolos y confirmando la acción. Los cambios se reflejarán en tiempo real. Además, el administrador podrá buscar productos específicos mediante un campo de búsqueda o aplicar filtros por categoría para facilitar la navegación. Los productos con bajo stock aparecerán resaltados en rojo para una rápida identificación.