

Taller 1 - Requerimientos funcionales

Ingeniería de Software I

Grupo 2

G9:

**Javier Giraldo
Álvaro Romero
Manuel Navas
Tomás Vallejo**

1. Selección del proyecto

Para el desarrollo del taller se eligió el chat obtenido por el grupo 2, de modo que se puede desarrollar con mayor orden y claridad la actividad.

2. Levantamiento de requerimientos

1. Objetivo del Proyecto Desarrollar una aplicación web para la gestión integral del negocio de ventas de productos y servicios de asesoría. La aplicación busca optimizar la organización de pedidos, inventarios, ventas, finanzas y la interacción con los clientes.

2. Descripción General del Negocio La usuaria gestiona un negocio de ventas de artesanías, productos de decoración y accesorios, junto con la posibilidad de ofrecer servicios de asesoría en decoración de espacios. Actualmente, utiliza WhatsApp, Instagram, Excel y una libreta para realizar sus operaciones diarias, pero desea una solución más eficiente y automatizada.

3. Requerimientos Funcionales

3.1. Gestión de Pedidos

- Creación, visualización y actualización de pedidos.
- Visualizar información del pedido: número de pedido, cliente, productos solicitados, estado (pendiente, pagado, enviado, completado, cancelado), fecha límite de entrega, monto total y medio de pago.
- Capacidad para modificar pedidos existentes.
- Generación de notificaciones automáticas para la administradora y los clientes.
 - Administradora: notificaciones de pedidos nuevos, cambios de estado y pedidos con productos agotados.
 - Cliente: confirmación de pedido con detalles y link de seguimiento.

3.2. Gestión de Inventario

- Registro y control de los productos disponibles.
- Información de productos: nombre, precio, cantidad en stock, descripción breve y fotos.
- Actualización automática del inventario al realizarse una venta.
- Notificaciones automáticas cuando el stock esté por agotarse.

3.3. Gestión de Ventas

- Registro automático de ventas con detalles de productos vendidos, montos y clientes.
- Integración con pasarelas de pago locales (Nequi, Daviplata) para facilitar pagos.
- Generación de comprobantes de pago para los clientes.

3.4. Gestión de Catálogo en Línea

- Visualización de productos en un catálogo interactivo.
- Posibilidad de que los clientes vean el estado de su pedido a través de un enlace.

3.5. Gestión de Clientes

- Registro automático de clientes al realizar un pedido.
- Información del cliente: nombre, número de contacto (WhatsApp o teléfono), dirección de envío e historial de pedidos.

3.6. Finanzas y Reportes

- Registro de ingresos y gastos.
- Informes automáticos de ventas, ingresos, gastos y ganancia neta.
- Resúmenes semanales o mensuales enviados por correo electrónico.

3.7. Notificaciones

- La administradora recibirá notificaciones de estado de pedidos y alertas de stock bajo mediante WhatsApp.
- Los clientes recibirán notificaciones de confirmación de pedidos y actualizaciones de estado mediante WhatsApp.

4. Requerimientos No Funcionales

- **Acceso Multidispositivo:** La aplicación debe ser accesible desde dispositivos móviles y computadoras.
- **Interfaz Intuitiva:** La interfaz debe ser sencilla y fácil de usar para usuarios sin experiencia técnica.
- **Disponibilidad:** La aplicación debe funcionar en línea, con capacidad de operar ciertas funciones offline en caso de pérdida de conexión.
- **Seguridad:** Los datos de clientes, pedidos y pagos deben estar protegidos bajo protocolos de seguridad.
- **Escalabilidad:** La aplicación debe permitir la adición de nuevas funciones (como la agenda de asesorías) en el futuro..

5. Priorización de Requerimientos

1. Gestión de pedidos.
2. Control de inventario.
3. Gestión de ventas y pagos.
4. Catálogo en línea.
5. Promoción de productos.
6. Agenda para asesorías (futuro).

6. Requisitos Técnicos

- **Tecnologías:** Aplicación web accesible desde navegadores móviles y de escritorio.
- **Integraciones:** Pasarelas de pago locales (Nequi, Daviplata) y servicios de notificación (WhatsApp y correo electrónico).

7. Presupuesto y Plazos

- **Presupuesto Inicial:** Aproximadamente 1 millón de pesos colombianos, con posibilidad de ajustes.
- **Plazo de Implementación:** Prototipo funcional en 1 mes con módulos principales (gestión de pedidos e inventario). Desarrollo continuo de funciones adicionales.

3. Análisis de requerimientos

Requerimientos Funcionales:

- Control de inventario.
- Portal de ventas.
- Registro de ventas.
- Chat con usuarios.
- Catálogo en línea.
- Horario de sesiones.
- Portal de ventas (con integración de pasarelas de pago locales).

Requerimientos No Funcionales:

- Acceso multidispositivo.
- Interfaz intuitiva.
- Disponibilidad en línea y operación offline.
- Seguridad en datos sensibles.
- Escalabilidad para futuras funcionalidades.

Método MoSCoW:

Debe tener:

- Control de inventarios.
- Portal de ventas.
- Registro de ventas.
- Seguridad en datos sensibles.

Debería tener:

- Chat con usuarios.
- Catálogo en línea.
- Acceso multidispositivo.
- Interfaz intuitiva.

Puede tener:

- Horario de sesiones.
- Escalabilidad

No va a tener:

- Disponibilidad offline.

4. Casos de uso

(TO DO: Una vez clasificadas las funcionalidades, realizar el diagrama y especificación de UN caso de uso del administrador y el usuario.)

1. Registro de producto en el inventario

Actor Administrador
Descripción El administrador puede agregar un nuevo producto al sistema, indicando nombre, cantidad, precio, y categoría.
Precondiciones El usuario debe estar autenticado o como invitado en el portal.
Flujos <ol style="list-style-type: none">1. El administrador selecciona la opción “Gestión de inventario”.2. Hacer click en “Agregar producto”.3. Ingresar los datos del producto:<ol style="list-style-type: none">a. Nombre.b. Categoría.c. Cantidad.d. Precio unitario.<ol style="list-style-type: none">i. Si el administrador no ingresa todos los datos requeridos, el sistema muestra un mensaje indicando los campos faltantes.4. Confirmar operación.<ol style="list-style-type: none">a. Si ocurre un error en la conexión al servidor, el sistema indica que la operación no se pudo completar e invita a intentar nuevamente.
Postcondiciones <ul style="list-style-type: none">• La venta queda registrada en el sistema.• El inventario se actualiza reduciendo la cantidad del producto comprado.

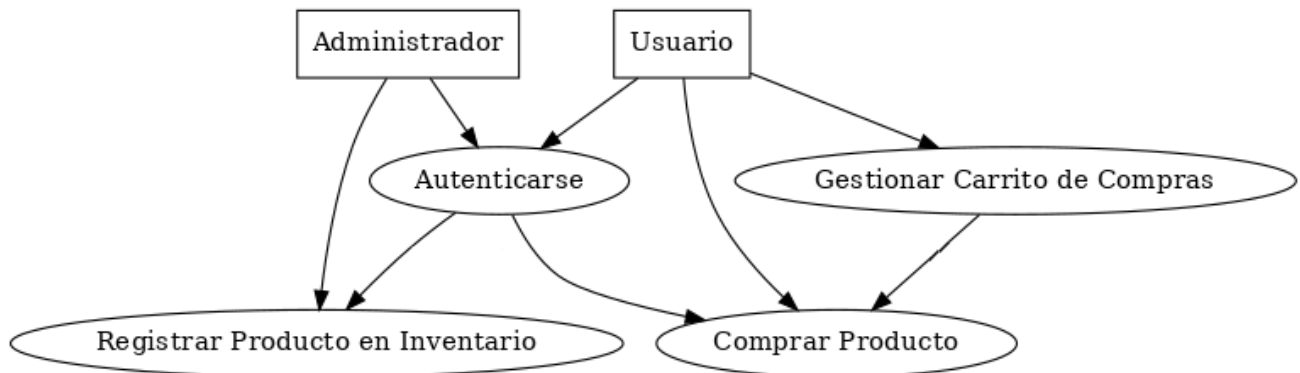
2. Comprar producto

Actor Usuario
Descripción Un usuario puede agregar productos al carrito, revisarlo y realizar la compra.
Precondiciones El usuario debe estar autenticado o como invitado en el portal.
Flujos <ol style="list-style-type: none">1. El usuario navega por el catálogo y selecciona un producto.2. Hacer click en “Agregar al carrito”.<ol style="list-style-type: none">a. Si la cantidad a agregar al carrito no se encuentra en inventario.

- b. Se indica al usuario por medio de un cuadro de diálogo.
3. Revisar el carrito de compras.
4. Confirma la compra y elige un método de pago.
5. Ingresar los datos de pago y confirma la transacción.
 - a. Si el método de pago falla, el sistema muestra un mensaje indicando el error y permite reintentar.
6. El sistema progresa el pago y genera una confirmación con un número de orden.

Postcondiciones

- La venta queda registrada en el sistema.
- El inventario se actualiza reduciendo la cantidad del producto comprado.

Diagrama:

5. Historia de usuario

Historia de Usuario #1

Descripción conceptual

Módulo	<i>Módulo de inventarios</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>El módulo debe permitir al administrador gestionar eficientemente los productos en el inventario, asegurando una actualización precisa de las cantidades, precios y categorías de los productos.</i>

Descripción técnica

Backend

Registrar producto de inventario

URL	Método	Código html
localhost:8080/admin/inventario/producto	POST	201 400
Descripción Al realizar una solicitud POST, el sistema debería recibir los datos de un producto y registrarlos en el inventario. Si el registro es exitoso, el sistema retorna un código 201 junto con los datos del producto registrado.		
Datos de entrada <pre>{ "nombre": "Producto A", "categoria": "Electrónica", "cantidad": 50, "precio": 15.500 }</pre>	Datos de salida 201 <pre>{ "status": "success", "data": { "id": 1, "nombre": "Producto A", "categoria": "Artesanía", "cantidad": 50, "precio": 15.500 } }</pre> 400 <pre>{ "status": "error", "message": "Faltan parámetros requeridos (parámetro faltante)" }</pre>	

Actualizar producto en inventario

URL	Método	Código html
localhost:8080/admin/invent		200

ario/producto/{id}	PATCH	422
<p align="center">Descripción</p> <p>Permite modificar los detalles de un producto en el inventario. Los datos actualizados deben ser enviados en el cuerpo de la solicitud. Si la operación es exitosa, el sistema retorna un código 200 con los datos actualizados. Si hay algún error en los datos de entrada, retorna un código 422.</p>		
<p align="center">Datos de entrada</p> <pre>{ "nombre": "Producto A - Actualizado", "categoria": "mueble", "cantidad": 45, "precio": 15.500 }</pre>	<p align="center">Datos de salida</p> <p>200</p> <pre>{ "status": "success", "data": { "id": 1, "nombre": "Producto A - Actualizado", "categoria": "mueble", "cantidad": 45, "precio": 15.500 } }</pre> <p>400</p> <pre>{ "status": "error", "message": "Faltan parámetros requeridos (parámetro faltante)" }</pre>	

Frontend

Interacción esperada

En esta historia de usuario, el Administrador interactúa con el sistema a través de una interfaz gráfica en el panel de administración, específicamente en el módulo de gestión de inventarios. La funcionalidad de la HU se centra en permitir al administrador agregar productos al inventario, consultar el estado de los productos y editar o eliminar productos existentes.

Interacciones clave

1. Formulario de registro de productos:
 - a. El administrador hace clic en un botón "Agregar Producto" en el menú de gestión de inventarios.
 - b. Se abre un formulario emergente (modal) donde podrá ingresar los datos del producto (nombre, cantidad, precio, categoría).
 - c. El formulario debe tener un botón de "Guardar" para enviar la información y agregar el producto, o "Cancelar" para cerrar el modal sin realizar cambios.
2. Visualización del inventario:
 - a. El administrador ve un listado de productos con los detalles más importantes: nombre, categoría, cantidad y precio.

- b. Cada producto tendrá dos botones al lado de su nombre: "Editar" y "Eliminar".
- c. Al hacer clic en "Editar", el administrador puede modificar los datos del producto en el mismo formulario que aparece al agregar un producto.
- d. Al hacer clic en "Eliminar", se muestra un modal de confirmación para asegurar que el administrador desea eliminar el producto.

Mockups

1. Vista general

[← Volver](#)

Productos

Agregar

ID	Nombre	Categoría	Cantidad	Precio	Eliminar	Editar

Cerrar sesión

2. Agregar producto

[← Volver](#)

Productos

Agregar

Nuevo producto

ID
Nombre:
Categoría:
Cantidad:
Precio:

GUARDAR
 CANCELAR

	Cantidad	Precio	Eliminar	Editar

Cerrar sesión

3. Editar producto

←

Volver

+

Agregar

⊗

Cerrar sesión

Productos

ID	Nombre	Categoría

Editar producto

ID

Nombre:

Categoría:

Cantidad:

Precio:

GUARDAR

⊗

CANCELAR

Editar

✍

✍

✍

✍

✍

✍

✍

✍

Los mockups fueron realizados con figma

Historia de Usuario #2

Descripción conceptual

Módulo	<i>Módulo del carrito</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>El módulo permite al usuario escoger ítems del catálogo y guardarlos en el carrito. Además se le permite al usuario modificar la cantidad de estos.</i>

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/users/catalogo	POST	200 422
Caso de uso técnico Al seleccionar un ítem del catálogo añadirlo al carrito por medio de su id y con la cantidad puesta por el usuario. Si se ejecuta la operación correctamente se retorna con el código 200. En caso de que falte alguno de los parámetros debe retornar el código 422.		
Datos de entrada 200: { "id": 1, "cantidad": 4 }	Datos de salida 200: { "status": "success", "data": { "id": 1, "nombre": "Producto A", "cantidad": 4, "precio": 8.500 } }	
422: { "id": 1, "csntidad": 4 }	422: { "status": "unprocessed", "message": "No se encontró el parámetro cantidad" }	

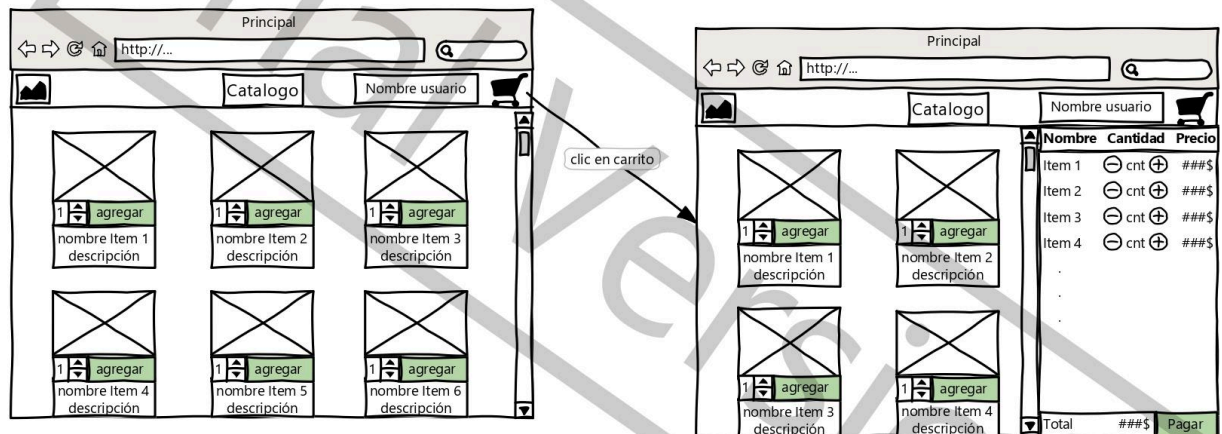
URL	Método	Código html
localhost:8080/users/catalogo/carrito	GET	200

Caso de uso técnico	
Al consultar, debe retornar el código 200 y un array con los productos en el carrito del usuario.	
Datos de entrada	Datos de salida
	<pre>{ "status": "success", "data": [{ "id": 1, "nombre": "Producto A", "cantidad": 1, "precio": 8.500 }, { "id": 2, "nombre": "Producto B", "cantidad": 3, "precio": 7.200 }, ...] }</pre>

URL	Método	Código html
localhost:8080/users/catalogo/carrito	PATCH	200 422
Caso de uso técnico		
Debe modificar las cantidades dentro del carrito con los valores modificados. Si se ejecuta la operación correctamente se retorna con el código 200. En caso de que falte alguno de los parámetros debe retornar el código 422.		
Datos de entrada	Datos de salida	
200: <pre>{ "id": 1, "cantidad": 2 }</pre>	200: <pre>{ "status": "success", "data": { "id": 1, "nombre": "Producto A", "cantidad": 2, "precio": 8.500 } }</pre>	
422: <pre>{ "id": 1, "csntidad": 4 }</pre>	422: <pre>{ "status": "unprocessed", "message": "No se encontró el parámetro"</pre>	

}	cantidad"
	}

Frontend



Historia de Usuario #3

Descripción conceptual

Módulo	<i>Módulo gestión de pedidos</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>El módulo debe permitir al administrador crear, visualizar, actualizar y gestionar el estado de los pedidos. Además, los clientes deben poder recibir notificaciones y consultar el estado de sus pedidos en tiempo real.</i>

Descripción técnica

Backend

Registrar un nuevo pedido

URL	Método	Código html
localhost:8080/admin/pedid		201

os	POST	400
Descripción El sistema recibe los datos del pedido y los registra en la base de datos. Si la operación es exitosa, retorna un código 201 con los detalles del pedido. Si faltan datos requeridos, retorna un código 400		
Datos de entrada <pre>{ "cliente": { "nombre": "Juan Pérez", "contacto": "3151234567" }, "productos": [{ "id": 1, "cantidad": 2 }, { "id": 3, "cantidad": 1 }], "medio_pago": "Nequi", "fecha_limite": "2024-12-31" }</pre>	Datos de salida 201 <pre>{ "status": "success", "data": { "id": 101, "cliente": "Juan Pérez", "productos": [{ "id": 1, "nombre": "Vela decorativa", "cantidad": 2, "precio_unitario": 5000 }, { "id": 3, "nombre": "Maceta pequeña", "cantidad": 1, "precio_unitario": 15000 }], "estado": "Pendiente", "monto_total": 25000, "medio_pago": "Nequi", "fecha_limite": "2024-12-31" } }</pre> 400 <pre>{ "status": "error", "message": "Faltan parámetros requeridos: cliente.nombre, productos" }</pre>	

Actualizar estado del pedido

URL localhost:8080/admin/pedidos/{id}	Método PATCH	Código html 200 422
Descripción Permite modificar el estado de un pedido específico. Retorna los datos actualizados del		

pedido si la operación es exitosa.

Datos de entrada	Datos de salida
<pre>{ "estado": "Enviado" }</pre>	<pre>200 { "status": "success", "data": { "id": 101, "estado": "Enviado" } } 400 { "status": "error", "message": "Pedido no encontrado" }</pre>

Frontend

Interacción esperada

- Creación de pedidos:
 1. El administrador selecciona "Gestión de Pedidos" en el menú principal.
 2. Hace clic en el botón "Nuevo Pedido"
 3. Se abre un formulario emergente para ingresar los datos del cliente, productos seleccionados y el medio de pago.
 4. Al guardar, el sistema genera el pedido y muestra un mensaje de confirmación.
- Visualización y actualización de pedidos:
 1. El administrador puede ver un listado de pedidos en una tabla con los detalles clave: número de pedido, cliente, estado, monto total, y fecha límite.
 2. Cada pedido tiene un botón de acción para actualizar su estado (e.g., "Pendiente" a "Enviado").
 3. Al actualizar el estado, el sistema envía automáticamente notificaciones al cliente.

Mockups

Tabla de pedidos con columnas para número de pedido, cliente, estado, monto total, fecha límite y acciones.

Tools and More		ST	Follow	Market Share	ST	Best Selling Products	ST	Product Details	ST
Product Name	ST	Follow	Market Share	ST	Best Selling Products	ST	Product Details	ST	
Apple iPhone 12 Pro	100	100	100	100	100	100	100	100	
Samsung Galaxy S21 Ultra	95	95	95	95	95	95	95	95	
Google Pixel 5	90	90	90	90	90	90	90	90	
OnePlus 9 Pro	85	85	85	85	85	85	85	85	
Xiaomi Mi 11 Ultra	80	80	80	80	80	80	80	80	
Huawei P50 Pocket	75	75	75	75	75	75	75	75	
Motorola Edge 2022	70	70	70	70	70	70	70	70	
Nokia 8.3 5G	65	65	65	65	65	65	65	65	
BlackBerry KEY2	60	60	60	60	60	60	60	60	
HTC U20 5G	55	55	55	55	55	55	55	55	
Asus ROG Phone 5	50	50	50	50	50	50	50	50	
Lenovo Legion 5	45	45	45	45	45	45	45	45	
Acer Predator Helios 300	40	40	40	40	40	40	40	40	
Dell Alienware 17	35	35	35	35	35	35	35	35	
Razer Blade 15	30	30	30	30	30	30	30	30	
MSI Stealth 15M	25	25	25	25	25	25	25	25	
ASUS ROG Zephyrus G14	20	20	20	20	20	20	20	20	
HP Spectre x360	15	15	15	15	15	15	15	15	
Lenovo Yoga 920	10	10	10	10	10	10	10	10	
Microsoft Surface Pro 8	5	5	5	5	5	5	5	5	

Historia de Usuario #4

Descripción conceptual

Módulo	Módulo gestión de pedidos
Descripción de la(s) funcionalidad(es) requerida(s):	El módulo debe permitir al administrador crear, visualizar, actualizar y gestionar el estado de los pedidos. Además, los clientes deben poder recibir notificaciones y consultar el estado de sus pedidos en tiempo real.

Descripción técnica

Backend

Cancelar pedido

URL	Método	Código html
localhost:8080/admin/pedidos	DELETE	201 400

Descripción	
El sistema recibe el id del pedido y lo borra de la base de datos. Si la operación es exitosa, retorna un código 201. Si no se encuentra el pedido con ese id, retorna un código 400	
Datos de entrada <pre>{ "id":101 }</pre>	Datos de salida 201 <pre>{ "status": "succes", "message": "el pedido fue cancelado" }</pre> 400 <pre>{ "status": "error", "message": "no se encontró el pedido" }</pre>

Front end

En la parte superior de cada pedido se encuentra un icono de color rojo el cual al ser oprimido sale un mensaje emergente el cual pregunta por la confirmación de la acción al aceptar de ejecuta la petición al backend.