



UNIVERSIDAD NACIONAL DE COLOMBIA



Taller Requerimientos

Presentado por:

Cristian Felipe Moreno Gómez

✉ crmorenogo@unal.edu.co

Diego Alejandro Rojas Reina

✉ drojasre@unal.edu.co

Edinson Sánchez Fuentes

✉ edsanchezf@unal.edu.co

Sebastián Olarte Ramírez

✉ molartera@unal.edu.co

Facultad de Ingeniería
Departamento de Sistemas e Industrial
Ingeniería de Software 1 (2016701) - Grupo 2
2024 – 2

Contenido

1	Conversación seleccionada	4
2	Levantamiento de requerimientos	4
2.1	Contexto del negocio	4
2.2	Problemas identificados	4
2.3	Solución	4
2.4	Levantamiento de Requerimientos Funcionales	5
2.4.1	Autenticación y autorización de usuarios	5
2.4.2	Gestión de pedidos:	5
2.4.3	Gestión de pagos: <i>(Esencial para los pedidos)</i>	5
2.4.4	Gestión de inventario:	5
2.4.5	Catálogo en línea:	6
2.4.6	Gestión de clientes:	6
2.4.7	Gestión financiera:(No es tan importante)	6
2.4.8	Notificaciones:	6
2.5	Levantamiento de Requerimientos No Funcionales	6
2.5.1	Facilidad de uso:	6
2.5.2	Accesibilidad:	7
2.5.3	Rendimiento:	7
2.5.4	Seguridad:	7
2.5.5	Estética y diseño visual:	7
3	Análisis de requerimientos	9
3.1	Análisis de Requerimientos Funcionales	9
3.1.1	Autenticación y autorización	9
3.1.2	Gestión de pedidos	10
3.1.3	Gestión de pagos	11
3.1.4	Gestión de inventario	12
3.1.5	Catálogo en línea	13
3.1.6	Gestión de clientes	13
3.1.7	Gestión financiera	14
3.1.8	Notificaciones	15
3.2	Análisis de Requerimientos No Funcionales	15
3.2.1	Facilidad de uso	15
3.2.2	Accesibilidad	16
3.2.3	Rendimiento	16

3.2.4	Seguridad	17
3.2.5	Estética y diseño visual	18
4	Especificación y diagrama de un caso de uso del administrador y el usuario	21
5	Historias de Usuarios	26
5.1	HU Login	26
5.2	HU Registro	28
5.3	HU Gestión de pedidos	30
5.3.1	Rol Cliente	30
5.3.2	Rol Administrador	33
5.4	Gestión de Inventario	36
6	<i>Referencias</i>	39

1 Conversación seleccionada

Teniendo en cuenta que es recomendable seleccionar la conversación que aporta mayor cantidad de información, en este taller sobre requerimientos se selecciona la conversación del “GP 2”.

2 Levantamiento de requerimientos

Durante la entrevista, la cliente manifestó necesidades relacionadas con la gestión de su negocio de ventas de artesanías y accesorios decorativos. Actualmente, maneja procesos de manera manual, lo que le genera problemas de organización y eficiencia. Entonces, ella busca una solución que centralice y automatice los procesos clave, permitiendo una gestión más eficiente y profesional, pero sin dejar de lado la accesibilidad y facilidad.

2.1 Contexto del negocio

La cliente tiene un negocio pequeño que combina productos hechos a mano (artesanías) con productos comprados para revender. También considera ofrecer asesorías de decoración, aunque aún no es prioritario por el momento. Actualmente, utiliza herramientas básicas como WhatsApp y Excel para gestionar inventarios, pedidos y ventas.

2.2 Problemas identificados

- Desorganización en la gestión de pedidos y pagos.
- Dificultades para mantener actualizado el inventario.
- Falta de visibilidad para los clientes (no existe un catálogo centralizado).
- Procesos manuales que consumen tiempo y son propensos a errores.
- Necesidad de herramientas para promocionar productos y captar nuevos clientes.

2.3 Solución

Tomando como base la conversación con la cliente, ella se decanta por la propuesta de una **página web** que proporcione todo lo que se requiere para su lógica de negocio, así pues, se propone una app web que sea intuitiva y fácil de manejar.

2.4 Levantamiento de Requerimientos Funcionales

2.4.1 Autenticación y autorización de usuarios

El sistema debe identificar y diferenciar entre los roles de administrador y cliente mediante un registro/ login seguro para personalizar las funcionalidades disponibles.

- Registro de usuarios con roles asignados (Administrador o Cliente).
- Inicio de sesión mediante correo electrónico y contraseña.
- Renderización de la interfaz y funcionalidades según el rol (Administrador o Cliente).
- Opción de registro rápido para clientes con Google o redes sociales (opcional).

2.4.2 Gestión de pedidos:

Este permite al administrador registrar, actualizar y gestionar pedidos de manera eficiente, manteniendo un seguimiento claro del estado de cada pedido y notificando tanto al administrador como al cliente sobre su progreso.

- Registro de pedidos con información detallada (cliente, productos, estado del pedido).
- Actualización automática del estado del pedido (Pendiente, Pagado, Enviado, Completado).
- Notificaciones automáticas al cliente sobre el estado de su pedido, preferiblemente vía WhatsApp.

2.4.3 Gestión de pagos: *(Esencial para los pedidos)*

Se debe integrar métodos de pago confiables y populares en la región, permitiendo que los clientes realicen pagos directamente desde la plataforma.

- Integración con pasarelas de pago locales como Nequi y Daviplata.
- Confirmación automática de pagos recibidos y vinculación con el estado del pedido.
- Registro de transacciones de pago en el historial del pedido.
- Notificaciones automáticas al administrador y al cliente al completar un pago.

2.4.4 Gestión de inventario:

El inventario debe actualizarse automáticamente según los pedidos realizados y debe proporcionar al administrador herramientas para gestionar los productos disponibles y evitar confusiones por falta de stock.

- Registro y seguimiento de stock por producto.

- Actualización automática del inventario tras completar un pedido.

2.4.5 Catálogo en línea:

La página web debe incluir un catálogo en línea accesible desde cualquier dispositivo, donde los clientes puedan visualizar productos, precios y disponibilidad.

- Visualización de productos con fotos, descripciones, precios y disponibilidad.
- Opciones para personalizar productos (como los llaveros).
- Filtro de productos por categorías.

2.4.6 Gestión de clientes:

El sistema debe almacenar y gestionar la información de los clientes para facilitar el seguimiento de sus pedidos y mantener un historial de compras.

- Almacenamiento de información de clientes (nombre, contacto, historial de pedidos).
- Registro automático de nuevos clientes al realizar un pedido.
- Posibilidad de editar datos de clientes.

2.4.7 Gestión financiera:(No es tan importante)

La solución debe ofrecer herramientas para registrar y analizar las finanzas del negocio, incluyendo ingresos por ventas y gastos relacionados.

- Registro de ingresos y gastos relacionados con el negocio.
- Resumen de ventas y ganancias netas en un formato visual fácil de entender.

2.4.8 Notificaciones:

El sistema debe enviar notificaciones automáticas relevantes tanto al administrador como a los clientes para mejorar la comunicación y eficiencia.

- Alertas al administrador sobre productos con bajo stock vía WhatsApp.
- Confirmación de pedidos y actualizaciones enviadas a clientes vía WhatsApp.

2.5 Levantamiento de Requerimientos No Funcionales

2.5.1 Facilidad de uso:

La interfaz debe ser intuitiva y amigable, diseñada para usuarios con poca experiencia técnica.

- Navegación clara y organizada con menús y botones bien etiquetados.
- Proceso simple para realizar pedidos, gestionar productos e inventarios.

- Capacitación mínima necesaria para el administrador al usar el sistema por primera vez.

2.5.2 Accesibilidad:

El sistema debe ser accesible desde dispositivos móviles y computadoras, asegurando compatibilidad y funcionalidad en múltiples plataformas.

- Compatible con navegadores modernos en móvil y computadora.
- Optimización para una experiencia fluida en pantallas pequeñas y grandes.
- Funcionalidad básica disponible offline (como consulta de inventarios).

2.5.3 Rendimiento:

La página debe estar optimizada para una carga rápida y un manejo eficiente de datos, especialmente en dispositivos con recursos limitados.

- Tiempo de carga de páginas menor a 3 segundos en condiciones de conexión estándar.
- Capacidad para manejar al menos 100 usuarios concurrentes sin afectar el rendimiento.

2.5.4 Seguridad:

Es necesario garantizar la protección de los datos de los usuarios y restringir el acceso a funciones sensibles según el rol asignado.

- Cifrado de contraseñas y datos sensibles (como transacciones de pago).
- Autenticación segura mediante HTTPS para todas las comunicaciones.
- Protección contra accesos no autorizados y ataques comunes como inyecciones SQL o XSS.

2.5.5 Estética y diseño visual:

El diseño debe reflejar el estilo cálido y profesional del negocio, creando una experiencia atractiva para los clientes.

- Uso de una paleta de colores cálidos y naturales (beige, terracota, verde oliva).
- Diseño limpio, minimalista y visualmente atractivo.
- Imágenes grandes y textos claros para destacar productos y promociones.

3 Análisis de requerimientos

3.1 Análisis de Requerimientos Funcionales

3.1.1 Autenticación y autorización

Autenticación y autorización				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Registro de usuarios con roles	Must Have	Es fundamental para diferenciar entre administradores y clientes, permitiendo personalizar las funcionalidades del sistema. La implementación es sencilla y estándar, por lo que no requiere mucho tiempo ni tiene alta complejidad.	2	Baja
Inicio de sesión con correo y contraseña	Must Have	Necesario para garantizar un acceso seguro al sistema. Utiliza prácticas estándar, por lo que su implementación es rápida y de baja complejidad.	2	Baja
Renderización según el rol	Must Have	Permite mostrar características específicas según el rol asignado, asegurando que los administradores y clientes tengan	3	Media

		experiencias diferenciadas. La lógica es sencilla, por lo que su implementación es directa.		
Registro rápido con Google (opcional)	Could Have	Agrega conveniencia para los clientes al facilitar el registro, pero no es esencial para el funcionamiento inicial del sistema. Su complejidad es baja al usar librerías de terceros.	2	Baja

3.1.2 Gestión de pedidos

Gestión de pedidos				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Registro de pedidos	Must Have	Es crucial para gestionar y rastrear pedidos. La lógica es sencilla y requiere formularios estándar, lo que facilita su implementación en poco tiempo.	2	Baja
Actualización automática del estado del pedido	Must have	Mejora el seguimiento del pedido, actualizando su progreso automáticamente. La complejidad	3	Media

		es media debido a la lógica necesaria para manejar múltiples estados y actualizaciones.		
Notificaciones vía WhatsApp	Should Have	Aumenta la comunicación con el cliente al informar sobre el progreso del pedido. Requiere integración con servicios de mensajería, lo que añade complejidad.	3	Media

3.1.3 Gestión de pagos

Gestión de pagos				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Integración con Nequi y Daviplata	Must Have	Fundamental para permitir pagos directos en la plataforma. La complejidad es alta debido a la necesidad de integrar APIs externas y asegurar transacciones seguras.	5	Alta
Confirmación automática de pagos	Should Have	Facilita el proceso de actualización del pedido tras recibir un pago. Es compleja debido a la lógica necesaria para verificar pagos y actualizar estados.	3	Alta

Registro de transacciones	Should Have	Permite un registro claro de pagos realizados. La implementación es directa y no requiere lógica avanzada.	2	Baja
Notificaciones de pago	Should Have	Mejora la comunicación al notificar sobre pagos exitosos. La implementación es sencilla, pero requiere integración con servicios de notificación	2	Baja

3.1.4 Gestión de inventario

Gestión de inventario				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Registro y seguimiento de stock	Must Have	Es esencial para administrar el inventario del negocio. La lógica es básica y no requiere mucho tiempo para implementarse.	2	Baja
Actualización automática tras pedido	Must Have	Evita problemas de stock al reducir automáticamente el inventario después de un pedido. Su complejidad es media debido a la lógica de actualización.	3	Media

3.1.5 Catálogo en línea

Catálogo en línea				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Visualización de productos	Must Have	Fundamental para que los clientes puedan ver los productos. Es una funcionalidad estándar y rápida de implementar.	2	Baja
Personalización de productos	Must Have	Ofrece valor agregado al permitir personalización. La lógica es sencilla y rápida de implementar.	2	Baja
Filtros por categorías	Must Have	Facilita la búsqueda de productos para los clientes. La lógica de filtrado es estándar y fácil de aplicar.	2	Baja

3.1.6 Gestión de clientes

Gestión de clientes				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Almacenamiento de información	Must Have	Esencial para hacer un seguimiento de los clientes y sus pedidos. La implementación	2	Baja

		es directa con formularios estándar.		
Edición de datos de clientes	Must Have	Automatiza el registro de clientes al generar un pedido, simplificando la gestión. Su complejidad es baja.	2	Baja
Registro automático de clientes	Should Have	Permite actualizar información de clientes cuando sea necesario. La lógica es sencilla y estándar.	2	Baja

3.1.7 Gestión financiera

Gestión financiera				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Registro de ingresos y gastos	Should Have	Facilita llevar el control financiero del negocio. Es una funcionalidad deseable, pero no esencial. Su implementación es rápida y sencilla.	2	Baja
Resumen visual de ventas y ganancias	Should Have	Proporciona una visión clara de las finanzas del negocio. Es útil, pero no crítica, y su complejidad es baja.	2	Baja

3.1.8 Notificaciones

Notificaciones				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Alertas de bajo stock vía WhatsApp	Must Have	Asegura que el inventario no se agote sin aviso. La complejidad es media por la integración con sistemas de notificación.	3	Media
Confirmaciones y actualizaciones a clientes	Should Have	Mejora la experiencia del cliente al mantenerlo informado. Su complejidad es media debido a las integraciones con servicios de mensajería.	3	Media

3.2 Análisis de Requerimientos No Funcionales**3.2.1 Facilidad de uso**

Facilidad de uso				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Navegación clara y organizada	Must Have	Esencial para que los usuarios puedan usar el sistema sin complicaciones. La implementación es sencilla.	1	Baja
Proceso simple de gestión	Should Have	Mejora la eficiencia del sistema. Su	1	Baja

		implementación es rápida y estándar.		
Capacitación mínima	Should Have	Garantiza una adopción rápida del sistema. La implementación se basa en una buena documentación.	1	Baja

3.2.2 Accesibilidad

Accesibilidad				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Compatibilidad con navegadores modernos	Must Have	Necesario para que el sistema funcione en múltiples dispositivos. Es estándar.	0	Baja
Optimización para diferentes pantallas	Must Have	Facilita una buena experiencia en cualquier dispositivo. Requiere diseño responsivo.	1	Media
Funcionalidad básica offline	Could Have	Proporciona acceso limitado sin conexión. Es opcional y su complejidad es media.	3	Media

3.2.3 Rendimiento

Rendimiento

Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Tiempo de carga menor a 3 segundos	Should Have	Mejora la experiencia del usuario. Requiere optimización avanzada.	3	Alta
Manejo de 100 usuarios concurrentes	Should Have	Garantiza estabilidad bajo carga moderada. Es compleja por las pruebas de carga.	3	Alta

3.2.4 Seguridad

Seguridad				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Cifrado de contraseñas y datos sensibles	Must Have	Fundamental para proteger información. Su complejidad es alta por prácticas avanzadas de seguridad.	3	Alta
Protección contra ataques comunes	Must Have	Esencial para prevenir vulnerabilidades. La complejidad es alta por las validaciones necesarias.	3	Alta
Autenticación mediante HTTPS	Should Have	Asegura la comunicación. Es de complejidad media por configuraciones específicas.	3	Media

3.2.5 Estética y diseño visual

Estética y diseño visual				
Funcionalidad	Prioridad (MoSCoW)	Descripción	Estimación (Días)	Complejidad
Paleta de colores cálidos	Should Have	Mejora la identidad visual. Su implementación es rápida.	1	Baja
Diseño limpio y minimalista	Should Have	Mejora la usabilidad. Es estándar y rápida de implementar.	1	Baja
Imágenes y textos destacados	Could Have	Agrega atractivo visual. Es opcional y de baja complejidad.	1	Baja

3.3 Resumen de Estimaciones

3.3.1 Resumen de Estimación de Requerimientos Funcionales

Prioridad	Número de Funcionalidades	Tiempo estimado (días)
Must Have	14	35
Should Have	8	19
Could Have	1	2

3.3.2 Resumen de Estimación de Requerimientos No Funcionales

Prioridad	Número de Funcionalidades	Tiempo estimado (días)
Must Have	5	8
Should Have	7	13
Could Have	2	4

3.3.3 Resumen de Estimación Total de Requerimientos Funcionales y No Funcionales

Prioridad	Número de Funcionalidades	Tiempo estimado (días)
Must Have	19	43

Should Have	15	32
Could Have	3	6

3.3.4 Planificación con recursos

Prioridad	Tiempo estimado (días)	Número de Desarrolladores	Tiempo estimado (días) / Número de Desarrolladores	Días hábiles
Must Have	43	4	43/4	11
Should Have	32	4	32/4	8
Could Have	6	4	6/4	2

4 Especificación y diagrama de un caso de uso del administrador y el usuario

Gestión de Pedidos por parte del Administrador
Actor: Administrador
Breve descripción: Permite al administrador, registrar, actualizar y gestionar pedidos en el sistema.
Descripción Paso a Paso Precondiciones: <ol style="list-style-type: none"> 1. El administrador está autenticado en el sistema y tiene permisos para gestionar pedidos. 2. Existen productos en el inventario. Flujo Principal (Escenario ideal): <ol style="list-style-type: none"> 1. El administrador accede al módulo de gestión de pedidos desde el menú principal. 2. Selecciona "Registrar nuevo pedido". 3. Ingresa la información del cliente (nombre, contacto) y los productos solicitados (nombre, cantidad). 4. El sistema valida el stock disponible y permite registrar el pedido si hay suficiente inventario. 5. El sistema actualiza automáticamente el estado del pedido a "Pendiente". 6. El cliente recibe una notificación automática vía WhatsApp confirmando el registro del pedido. 7. Conforme el pedido avanza, el administrador actualiza su estado: <ul style="list-style-type: none"> o "Pagado" tras confirmar el pago. o "Enviado" una vez el pedido ha sido despachado. o "Completado" al finalizar la entrega. 8. En cada cambio de estado, el cliente recibe una notificación. 9. El inventario se ajusta automáticamente al marcar el pedido como "Completado". Flujos alternativos: <ol style="list-style-type: none"> 1. Stock insuficiente: <ul style="list-style-type: none"> o El administrador intenta registrar un pedido, pero uno o más productos no tienen suficiente inventario. o El sistema muestra un mensaje: "Stock insuficiente para el producto [nombre del producto]. Ajuste las cantidades o actualice el inventario." o El administrador revisa y ajusta el pedido o actualiza el inventario antes de continuar. 2. Datos incompletos del cliente: <ul style="list-style-type: none"> o El administrador intenta registrar un pedido, pero no se ingresaron todos los datos requeridos del cliente. o El sistema muestra un mensaje: "Información del cliente incompleta. Por favor, complete todos los campos obligatorios."

- El administrador corrige la información y continúa.
- 3. **Error en la conexión con el servidor:**
 - El administrador envía los datos del pedido, pero ocurre un problema técnico que impide el registro en la base de datos.
 - El sistema muestra un mensaje: "No se pudo registrar el pedido debido a un problema técnico. Intente nuevamente más tarde."
 - El administrador verifica la conexión o contacta al soporte técnico.

Postcondiciones:

1. El pedido queda registrado y vinculado al cliente con su historial de cambios.
2. El inventario se ajusta automáticamente al completar el pedido.
3. Los clientes reciben notificaciones sobre el progreso del pedido.

Diagrama de Caso de Uso: Gestión de Pedidos por parte del Administrador

Compra y Pago por parte del Cliente**Actor:** Cliente**Breve descripción:**

Permite a un cliente realizar una compra seleccionando productos desde el catálogo en línea y completando el pago mediante pasarelas locales.

Descripción Paso a Paso**Precondiciones:**

1. El cliente debe estar registrado y autenticado en el sistema.
2. Deben existir productos disponibles en el inventario.

Flujo Principal (Escenario ideal):

1. El cliente accede al catálogo en línea desde su navegador.
2. Navega por las categorías de productos y selecciona los que desea comprar.
3. Agrega los productos al carrito, personalizándolos si aplica.
4. Procede a realizar el pedido, ingresando su información personal o iniciando sesión.
5. Selecciona un método de pago compatible (Nequi o Daviplata).
6. Realiza el pago desde la plataforma mediante la pasarela seleccionada.
7. El sistema confirma automáticamente el pago y actualiza el estado del pedido a "Pagado".
8. El cliente recibe una notificación confirmando el pago y el inicio de la preparación del pedido.
9. Una vez que el pedido es enviado, el cliente recibe otra notificación con los detalles del envío.

Flujo alternativo:

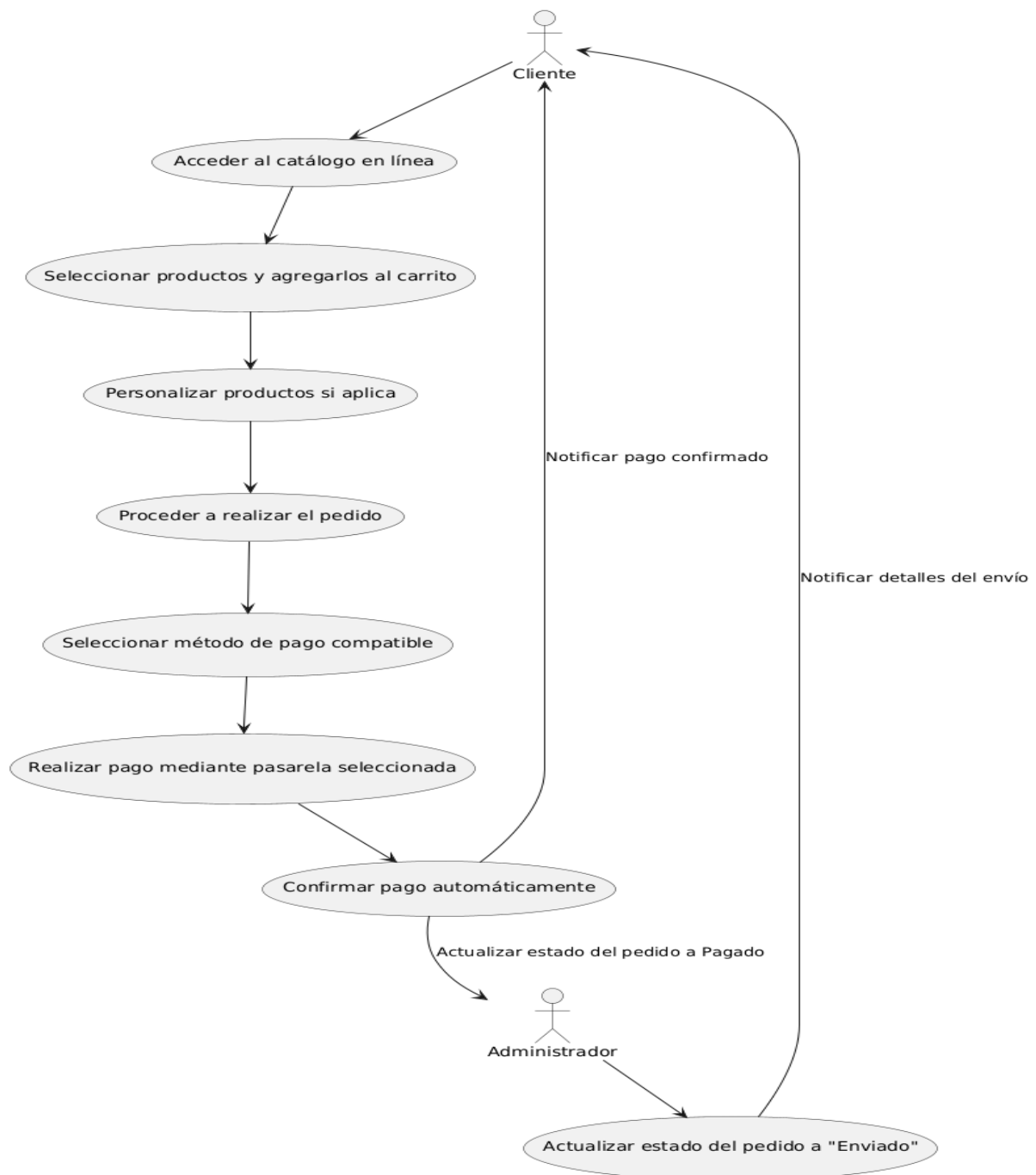
1. **Fallo en el pago:**
 - El cliente intenta realizar el pago, pero la pasarela rechaza la transacción (por saldo insuficiente, error en la tarjeta, etc.).
 - El sistema muestra un mensaje: "Pago rechazado. Por favor, verifique su método de pago e intente nuevamente."
 - El cliente corrige el problema o selecciona otro método de pago.
2. **Producto agotado durante la compra:**
 - Al confirmar el pedido, el sistema detecta que uno de los productos seleccionados ya no está disponible.
 - Muestra un mensaje: "El producto [nombre del producto] no está disponible actualmente. Por favor, actualice su carrito."
 - El cliente ajusta su selección y vuelve a intentar.
3. **Problemas de conexión al sistema:**
 - El cliente intenta confirmar el pedido, pero ocurre un error en la conexión con el servidor.
 - El sistema muestra un mensaje: "No se pudo completar la compra debido a un problema técnico. Intente nuevamente más tarde."

- o El cliente espera o contacta al soporte técnico para resolver el problema

Postcondiciones:

1. El pedido queda registrado con todos los detalles, incluido el pago y estado actual.
2. El cliente puede consultar su historial de pedidos en su perfil.
3. El inventario se ajusta automáticamente según los productos solicitados.

Diagrama de Caso de Uso: Compra y Pago por parte del Cliente



5 Historias de Usuarios

5.1 HU Login

Anexo de Documentos Relacionados:

- En este apartado se agregan toda la documentación que sea relevante para entender o sustentar la historia de usuario

Descripción conceptual

Módulo	Autenticación y autorización de usuarios
Descripción de la(s) funcionalidad(es) requerida(s):	El sistema debe identificar y diferenciar entre los roles de administrador y cliente mediante un registro/ login seguro para personalizar las funcionalidades disponibles.

Descripción técnica

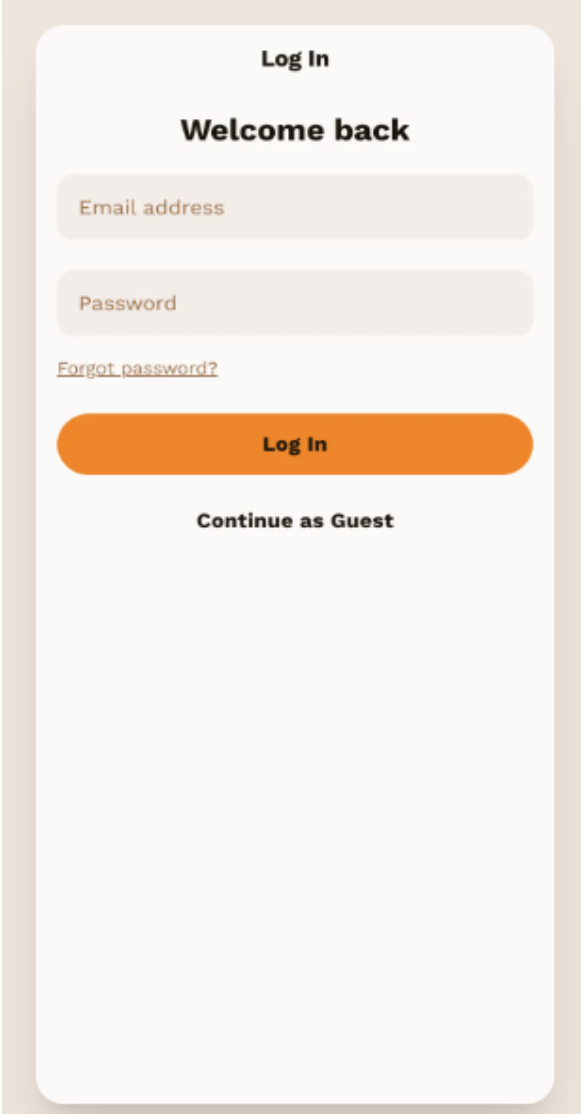
En este apartado existen dos partes, la del backend y la del frontend

Backend:

URL	Método	Código html
localhost:8080/login	GET	200 401
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios si retorna 401 es porque las credenciales están mal		
Datos de entrada { "data": [{ "usuario": "Juan Pérez" "password": "*****"] }] }	Datos de salida { "status": "success", "data": [{ "id": 1, "nombre": "Juan Pérez" "imagen": "imagen.html" }] }	
Datos de entrada { "data": [{ "usuario": "Juan Pérez" "password": "*****"] }] }	Datos de salida { "status": "fail", "data": [{ "message": "credenciales incorrectas", }] }	

Frontend

En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción

A mobile app login screen mockup. At the top, it says "Log In" and "Welcome back". Below that are two input fields: "Email address" and "Password". A link "Forgot password?" is positioned below the password field. A large orange button labeled "Log In" is centered below the inputs. At the bottom, there is a link "Continue as Guest".

Log In

Welcome back

Email address

Password

[Forgot password?](#)

Log In

[Continue as Guest](#)

Interacción esperada:

el usuario se meterá a esa vista y diligenciará sus credenciales

Mockups/Prototipos:

(Insertar imágenes o esquemas de las pantallas esperadas)

Flujo visual y eventos:

(Especificar el comportamiento esperado, por ejemplo, qué sucede al hacer clic en un botón, mostrar un spinner mientras carga, etc.)

5.2 HU Registro

Anexo de Documentos Relacionados:

- *En este apartado se agregan toda la documentación que sea relevante para entender o sustentar la historia de usuario*

Descripción conceptual

Módulo	<i>Registro de usuarios</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>El sistema debe permitir al usuario crear un perfil</i>

Descripción técnica

En este apartado existen dos partes, la del backend y la del frontend

Backend:

URL	Método	Código html
localhost:8080/register	GET	200 400
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios si retorna 400 es no se puede crear el usuario debido a una Bad Request		
Datos de entrada { "data": [{ "usuario": "Juan Pérez" "password": "*****" "imagen": "imagen.jpg" }] }	Datos de salida { "status": "success", "data": [{ "id": 1, "nombre": "Juan Pérez" "imagen": "imagen.jpg" }] }	
Datos de entrada {	Datos de salida {	

<pre>"data": [{ "usuario": "J" "password": "*****" }] }</pre>	<pre>"status": "fail", "data": [{ "message": "Bad Request", }] }</pre>
---	--

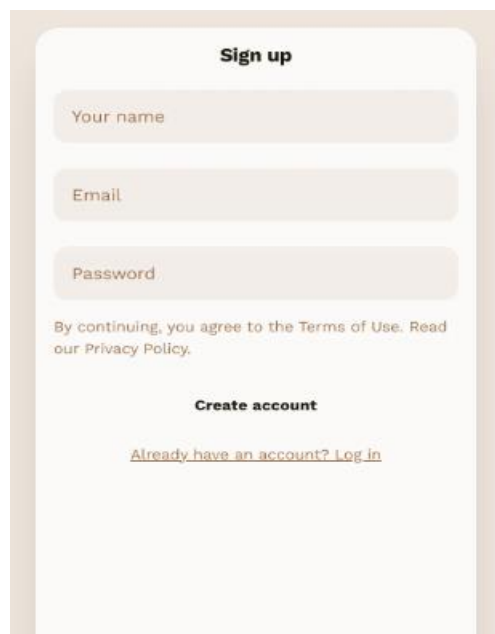
Frontend

En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción

Interacción esperada:

(Describir cómo el usuario interactúa con la funcionalidad, ejemplo: un botón, un formulario, un modal, etc.)

Mockups/Prototipos:



Flujo visual y eventos:

el usuario creara su perfil llenado el email, password y el nombre del usuario

5.3 HU Gestión de pedidos

5.3.1 Rol Cliente

Anexo de Documentos Relacionados:

- En este apartado se agregan toda la documentación que sea relevante para entender o sustentar la historia de usuario

Descripción conceptual

Módulo	<i>Agregar pedido (cliente)</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Este módulo permite a los clientes pasar del carrito de compras al checkout y terminar la transacción</i>

Descripción técnica

En este apartado existen dos partes, la del backend y la del frontend

Backend:

URL	Método	Código html
localhost:8080/checkout?id Products=1,2 ...	GET	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios si retorna 403 es porque no está autorizado		
Datos de entrada <pre>{ "data": [{ "token": "asF6sSFd4" }] }</pre>	Datos de salida <pre>{ "status": "success", "data": [{ "products": [{ "idProducto": "asdfasdf" "nombre": "producto1" "precio": 50000 "descripcion": "este es nuestro producto" }], }] }</pre>	
Datos de entrada	Datos de salida	
{	{	

<pre>"data": [{ "token": "" }] }</pre>	<pre>"status": "fail", "data": [{ "message": "Unauthorized", }] }</pre>
--	---

URL	Método	Código html
localhost:8080/checkout?idP roducts=1,2 ...	POST	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios si retorna 400 es no se puede crear el usuario debido a una Bad Request		
Datos de entrada <pre>{ "data": [{ "token": "asF6sSFd4" }] }</pre>	Datos de salida <pre>{ "status": "success", "data": [{ "products": [{ "idOrden": "sdfsd" "status": "en camino" "productos": "{}" "precio": 50000 }], }] }</pre>	
Datos de entrada <pre>{ "data": [{ "token": "" }] }</pre>	Datos de salida <pre>{ "status": "fail", "data": [{ "message": "Unauthorized", }] }</pre>	

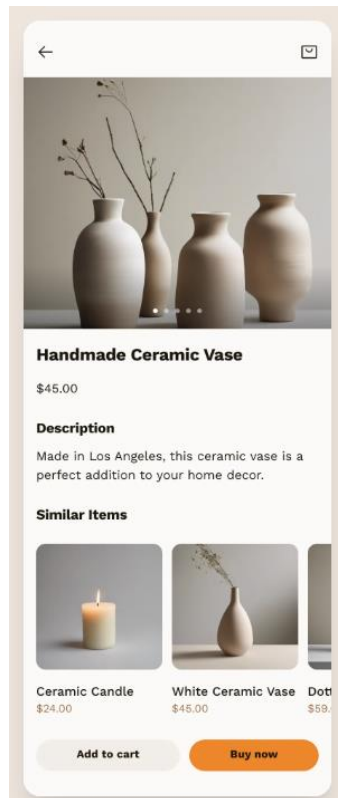
Frontend

En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción

Interacción esperada:

(Describir cómo el usuario interactúa con la funcionalidad, ejemplo: un botón, un formulario, un modal, etc.)

Mockups/Prototipos:



Flujo visual y eventos:

se espera que el usuario pueda navegar en la vista general de productos y la vista detallada de cada uno

5.3.2 Rol Administrador

Anexo de Documentos Relacionados:

- En este apartado se agregan toda la documentación que sea relevante para entender o sustentar la historia de usuario

Descripción conceptual

Módulo	<i>Cambiar estado de pedido pedido (cliente)</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Este módulo permite a los Administradores gestionar el estado del pedido, cambiar de recibido a en camino y a entregado</i>

Descripción técnica

En este apartado existen dos partes, la del backend y la del frontend

Backend:

URL	Método	Código html
localhost:8080/dashboard/pedidos	GET	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los pedidos si retorna 403 es porque no está autorizado		
Datos de entrada <pre>{ "data": [{ "token": "asF6sSFd4" }] }</pre>	Datos de salida <pre>{ "status": "success", "data": [{ "products": [{ "idProducto": "asdfasdf" "nombre": "producto1" "precio": 50000 "descripcion": "este es nuestro producto" }], }] }</pre>	
Datos de entrada {	Datos de salida {	

<pre>"data": [{ "token": "" }] }</pre>	<pre>"status": "fail", "data": [{ "message": "Unauthorized", }] }</pre>
--	---

URL	Método	Código html
localhost:8080/dashboard/pedidos	PATCH	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y un mensaje con la actualización efectuada si retorna 400 es no se puede crear el usuario debido a una Bad Request		
Datos de entrada { "data": [{ "token": "asF6sSFd4" "idOrder": "123421422" "status": "Entregado" }] }	Datos de salida { "status": "success", "data": [{ "mensaje": "Registro actualizado" }] }	
Datos de entrada { "data": [{ "token": "" }] }	Datos de salida { "status": "fail", "data": [{ "message": "Unautorized", }] }	

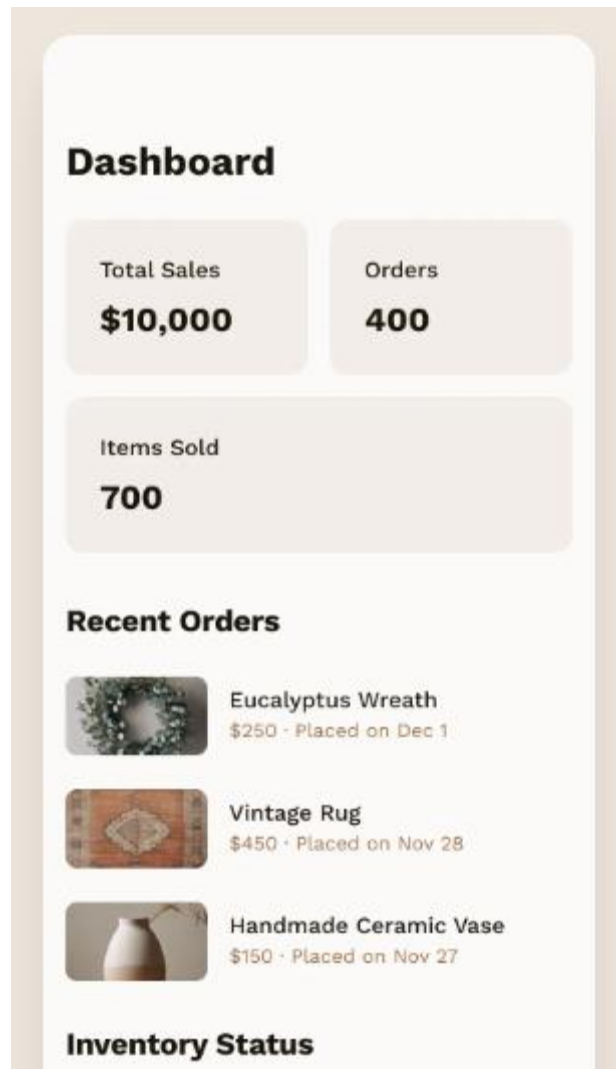
Frontend

En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción

Interacción esperada:

(Describir cómo el usuario interactúa con la funcionalidad, ejemplo: un botón, un formulario, un modal, etc.)

Mockups/Prototipos:



Flujo visual y eventos:

se espera tener los datos de las ventas más relevantes, las órdenes, los estados de los pedidos y la edición de ellos en la misma vista

5.4 Gestión de Inventario

Anexo de Documentos Relacionados:

- En este apartado se agregan toda la documentación que sea relevante para entender o sustentar la historia de usuario

Descripción conceptual

Módulo	<i>Cambiar estado de pedido pedido (cliente)</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Este módulo permite a los Administradores gestionar el inventario</i>

- **Descripción técnica**

En este apartado existen dos partes, la del backend y la del frontend

Backend:

URL	Método	Código html
localhost:8080/dashboard/inventario	GET	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los pedidos si retorna 403 es porque no está autorizado		
Datos de entrada <pre>{ "data": [{ "token": "asF6sSFd4" }] }</pre>	Datos de salida <pre>{ "status": "success", "data": [{ "products": [{ "idProducto": "asdfasdf" "nombre": "producto1" "precio": 50000 "descripcion": "este es nuestro producto" }], }] }</pre>	
Datos de entrada <pre>{ "data": [{ "token": "" }] }</pre>	Datos de salida <pre>{ "status": "fail", "data": [{ "message": "Unauthorized", }] }</pre>	

<pre>]] } </pre>	<pre>]] } </pre>
-------------------------	-------------------------

URL	Método	Código html
localhost:8080/dashboard/pedidos	PATCH	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y un mensaje con la actualización efectuada si retorna 400 es no se puede crear el usuario debido a una Bad Request		
Datos de entrada <pre> { "data": [{ "token": "asF6sSFd4" "idOrder": "123421422" "status": "Entregado" }] } </pre>	Datos de salida <pre> { "status": "success", "data": [{ "mensaje": "Registro actualizado" }] } </pre>	
Datos de entrada <pre> { "data": [{ "token": "" }] } </pre>	Datos de salida <pre> { "status": "fail", "data": [{ "message": "Unautorized", }] } </pre>	

URL	Método	Código html
localhost:8080/dashboard/pedidos	DELETE	200 403
Caso de uso técnico Al consultar, debe retornar código 200 y un mensaje con la actualización efectuada si retorna 403 no esta autorizado para modificar el inventario		
Datos de entrada <pre> { "data": [{ "token": "asF6sSFd4" "idProducto": "123421422" }] } </pre>	Datos de salida <pre> { "status": "success", "data": [{ "mensaje": "Producto Borrado" }] } </pre>	

Datos de entrada	Datos de salida
<pre>{ "data": [{ "token": "" }] }</pre>	<pre>{ "status": "fail", "data": [{ "message": "Unauthorized", }] }</pre>

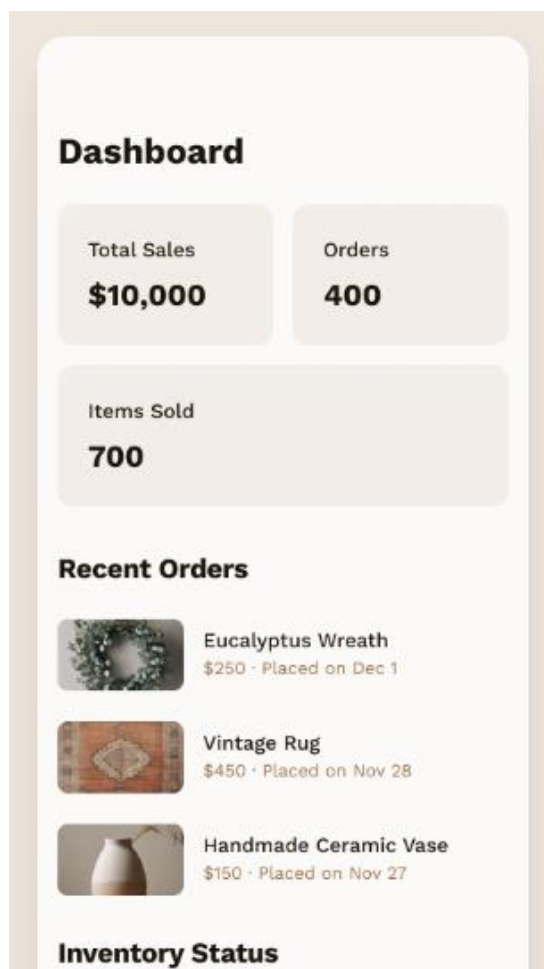
Frontend

En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción

Interacción esperada:

(Describir cómo el usuario interactúa con la funcionalidad, ejemplo: un botón, un formulario, un modal, etc.)

Mockups/Prototipos:



Flujo visual y eventos:

se espera tener los datos de las ventas más relevantes, las órdenes, los estados de los pedidos y la edición de ellos en la misma vista

6 Referencias

[1] OpenAI. (2024). *ChatGPT (versión GPT-4)*. OpenAI. Disponible en <https://openai.com/chatgpt>

[2] PlantUML. (n.d.). *Documentación oficial de PlantUML*. Recuperado el 19 de diciembre de 2024, de <https://plantuml.com/es/>

[3]Galileo ai. disponible en <https://www.usegalileo.ai/>