

Taller requerimientos

Realizado por:

Andres Felipe Ramirez Montana

Isabella Garces Acosta

Johan David Lozano Leiva

Maria Luisa Bautista Arango

Ingeniería de sistemas y computación



Universidad Nacional de Colombia

Sede Bogotá

Ingeniería de software I

Profesor:

Oscar Eduardo Alvarez

1. Elegir (escriban cual eligieron de manera clara...) una de las dos conversaciones obtenidas en ambos cursos para trabajar todo el taller, están disponibles aquí:
Mi recomendación el GP2 logró hacer que explicara de manera MUY detallada, es decir tienen mucha más facilidad al trabajar con esta conversación.

POD: revisen cuanto ofrecía la cliente para el desarrollo 😊

Elegimos la conversación del grupo 2.

2. Redactar el levantamiento de requerimientos, recuerden que, es una narración que cuenta de manera objetiva y organizada la información relevante obtenida en el proceso de levantamiento (en este caso, entrevista con la IA), estas van a describir funcionalidades pero desde una manera conceptual.

Debe terminar siendo algo parecido a una lista de funcionalidades y especificaciones técnicas, después de haber sido contextualizado, sin embargo, cada una debe estar asociada a un proceso dentro del software.

La cliente es una emprendedora con un pequeño negocio enfocado en la venta de artesanías, accesorios para decoración y con la posibilidad de añadir servicios de asesoría en diseño de espacios. Está presentando problemas en la organización de sus procesos, principalmente en la gestión de pedidos, inventarios y comunicación con sus clientes, esto ya que maneja todo por whatsapp y excel, y no cuenta con un inventario ni catálogo claros. El presupuesto inicial de la cliente es de un millón de pesos y tiene previsto que demore un mes el proyecto.

Por lo anterior, el objetivo principal es crear una página web que permita al cliente gestionar su negocio de forma más organizada. En cuanto a los **requerimientos funcionales**, esta página debe tener:

Un sistema de **gestión de pedidos**, el cual los organice de forma clara, incluyendo los datos del cliente, los productos que ha solicitado, el estado del pedido y la fecha de entrega, teniendo un ID para cada pedido.

Control de inventario, el cual muestre cual es el inventario disponible y que este se actualice automáticamente según las ventas, que pueda incluir alertas de bajo stock y que permita al administrador modificar el inventario ya sea agregando, editando o eliminando productos.

Catálogo en línea que permita a los clientes acceder y explorar los productos con sus precios, imágenes y disponibilidad. Que divida los productos por categorías y que muestre solo los productos en stock.

Gestión financiera, se debe tener un registro de las transacciones realizadas y poder mantener el control sobre los gastos e ingresos. Se podría integrar una pasarela de pagos con pago por medio de nequi o daviplata, por ejemplo.

Notificaciones para el cliente sobre la confirmación de un pedido o novedades sobre los pedidos realizados.

Opciones de personalización: por último, debe tener un espacio donde el cliente pueda añadir sus preferencias con respecto a los productos que desea comprar, especialmente para los diseños de las artesanías, por ejemplo un formulario con las opciones del artículo que se vincule con el comprador.

Interfaz del administrador que le permita acceder a un resumen de la gestión de sus productos, pedidos y clientes.

En cuanto a **requerimientos no funcionales**:

Usabilidad: La página debe ser fácil de usar para usuarios con poca experiencia en tecnología.

Accesibilidad: Que sea accesible por medio de un navegador en dispositivos móviles y en computador.

Escalable: Que tenga la posibilidad de añadir funciones nuevas como lo que tenga que ver con los servicios de asesoría.

Rendimiento: Que la página cargue rápidamente.

Seguridad: Principalmente con las transacciones financieras y los datos personales.

Además se tienen unas sugerencias para la interfaz gráfica:

Se busca que el diseño refleje la calidez y el aspecto artesanal del negocio, con una paleta de colores naturales como el beige, el terracota o el verde oliva, así como una tipografía amigable que refleje las mismas características.

3. Una vez redactado el levantamiento de requerimientos, realizar el análisis de requerimientos, aquí ya se empieza el proceso de traducción a un listado de funcionalidades, clasificarlas, ordenarlas y detallarlas con **el método MoSCoW**, hagan un estimado de tiempo y complejidad asumiendo y proponiendo recursos disponibles (cantidad de desarrolladores, tiempo, costo, tecnologías, todo aquello que pueda impactar la triada de desarrollo en ingeniería de software) (Tiempo, costo, alcance)

Clasificación MoSCoW.

Must:

- **Gestión de pedidos.**
- **Control de inventarios.**
- **Catálogo en línea.**
- **Seguridad.**

Should:

- **Gestión Financiera.**
- **Usabilidad.**

Could:

- **Rendimiento**
- **Notificaciones.**
- **Accesibilidad.**

Won't:

- **Escalabilidad.**
- **Opciones de personalización.**

Estimación de tiempo:

Para la estimación de tiempo se van a utilizar las siguientes tecnologías:

- Python para el lado del servidor utilizando sus frameworks como Django.
- Astro para la interfaz dinámica para la experiencia de usuario.
- PostgreSQL para la base de datos del sistema.

Estas tecnologías en general son fáciles de manejar y la curva de aprendizaje no es demasiado compleja.

Para este caso se va a utilizar bases de datos relaciones haciendo una proyección al futuro. Por ejemplo, se puede crear entidades como cliente, productos, inventario. Además, estas entidades tendrán relaciones fuertes cuando el sistema esté en funcionamiento. Otro punto a tomar en cuenta es para la generación de reportes, por ejemplo en la gestión de pedidos se puede generar reportes como: "Ganancias de todos los pedidos que se hicieron en el último mes". Esto se puede generar con la ayuda del motor de búsqueda de una base de datos relacional.

Una vez ya seleccionadas las tecnologías a usar el equipo de para hacer el desarrollo de software se va a componer de 6 personas donde 5 son las personas están enfocadas en los conocimientos técnicos y el otro será el project management.

Ahora bien, teniendo en cuenta el equipo de trabajo podemos estimar los costos para la traída del software. Teniendo en cuenta la ubicación de la empresa contratada y el contexto económico, ya que para desplegar la aplicación se requiere de otras tecnologías encargadas de hacer el despliegue y los servicios, es importante contar con todas las herramientas necesarias para el funcionamiento y el rendimiento ya que la aplicación pueda contar con varios usuarios y roles.

En resumen los costos se rigen bajo el uso de tecnologías, salarios, licencias de software, infraestructura (como servidores, almacenamiento en la nube, dominios y hosting), tiempo de desarrollo, capacitación del equipo, pruebas de calidad, mantenimiento posterior al desarrollo y posibles gastos asociados a cambios o ampliaciones en los requerimientos iniciales.

Una vez ya definido esto, se puede hacer las siguientes estimaciones de tiempo para requerimiento tanto funcional como no funcional:

- **Gestión de pedidos:** Para este requisito funcional se requiere la recopilación de varios lados del sistema, tanto como de la base de datos como el sistema en general, se puede estimar que para la creación de peticiones o llamadas HTTP, las consultas respectivas para traer los productos y las gestión de las respuestas y solicitudes a la vista, todo esto condicionado con la intención del usuario, entre otras cosas, un total de 4 días. Por otro lado, para la creación de la interfaz dinámica con la manera de que el dueño del usuario pueda buscar, organizar y gestionar productos todo desde la vista se puede estimar 3 días. Todo esto teniendo en cuenta la experiencia del usuario.
- **Control de inventarios:** Para este caso se requiere más llamadas a la base de datos, es decir, utilizar las funcionalidades CRUD validando el rol del usuario. Hay que gestionar las consultas tanto en su estructura como en su creación. Otro punto importante es la organización por parte de la interfaz utilizando un filtrado específico para la completa gestión y control de inventarios. Dado esto se puede estimar 5 días para la realización y preparación por parte del servidor y el motor de base de datos, Además, para este caso se usa se estimas más días ya que hay que acaparar todos los escenarios posibles para la generación de consultas. Por parte de la interfaz se requiere más trabajo, ya que se requiere funcionalidades complementarias que ayuden a organizar la información por parte del servidor, por ende se estima 4 días.
- **Catálogo en línea:** Para este requerimiento es más fácil de implementar ya que el servidor solo tiene que hacer la función de READ a la base de datos. Por parte de la interfaz se requiere un poco más de trabajo dado que el catálogo se caracteriza por búsquedas condicionales, clasificación de productos con información como el precio y categoría. Por parte del servidor se puede estimar un total de 2 días. Para la interfaz se requiere un total de 5 días dado todo lo escrito anteriormente.

- **Seguridad:** Para este caso se debe implementar otro apartado como métodos de autenticación como JWT, definición de roles y permisos, gestión de contraseñas (que para este caso se utiliza algoritmos específicos), encriptación de datos de tránsito utilizando por ejemplo HTTPS etc. Evitar las inyecciones SQL y vulnerabilidades de las librerías utilizadas en el sistema. Para cada implementación se requiere un tiempo específico, uno que otro más largo que otro, pero se puede hacer una estimación de 2-3 semanas haciendo un pronóstico así por encima.
- **Gestión Financiera:** Aca entramos con la implementación de un servicio aparte pero importante y es el de pagos. Dependiendo de los deseos del cliente, se puede implementar pasarelas como transacciones bancarias o con la validación de tarjetas de crédito y débito. Manejo de información delicada como el de las transacciones y la generación de reportes con el flujo neto de la compra y venta de productos. El sistema será capaz de guardar todas las ventas y compras por la fecha y hora ya que esto es importante para la generación de reporte. Dado escrito lo anterior, por parte del servidor se puede estimar un total de 3 días ya que se debe gestionar las consultas, la información de las consultas y siempre procurar la seguridad. La interfaz se estima en un total de 4 días.
- **Usabilidad:** Para este requerimiento, solo incluye trabajo por parte de la vista o frontend. Esto requiere el análisis, diseño y estudio de la experiencia del usuario, considerando aspectos como la disposición de elementos en la interfaz, la facilidad de navegación y la comprensión de los flujos de trabajo. Además, se debe garantizar que el diseño sea intuitivo, especialmente para usuarios con poca experiencia tecnológica, empleando prácticas como pruebas de usabilidad y prototipado. Para esto depende el nivel del detalle, pero esto se estima 1-2 semanas.
- **Rendimiento:** Para este requerimiento se necesita un revisión general del código, tanto de las librerías tanto del servidor o como el front, algoritmos utilizados para el tránsito de información en el software, etc. Esto requiere bastante tiempo ya que se deben hacer modificaciones y tener en cuenta la aparición de errores. Se estima un total de 2 semanas.
- **Notificaciones:** Se estima un total de 2 días, ya que por lo general hay módulos o librerías que realizan todo el trabajo.
- **Accesibilidad:** Las tecnologías seleccionadas de por sí tienen bastante compatibilidad con varios navegadores modernos en hoy en día. Pero un factor importante es mirar el rendimiento y la compatibilidad con los navegadores antiguos como firefox. Se estima un total de 1 semana.
- **Escalabilidad:** Para este caso si depende bastante de las nuevas funcionalidades que se quieran implementar. Para este tipo de estimación se deja abierta.
- **Opciones de personalización:** Para las opciones de personalización, en el frontend se debe diseñar un formulario dinámico e intuitivo que permita a los clientes seleccionar o ingresar sus preferencias de productos, integrándose con el catálogo existente. En el backend, es necesario crear una estructura en la base de datos que

vincule estas preferencias con los pedidos y desarrollar lógica para procesar y almacenar los datos enviados. En cuanto a la **interfaz del administrador**, el **frontend** requerirá un panel con tablas interactivas, gráficos y filtros que muestran un resumen claro de productos, pedidos y clientes. El **backend** deberá proporcionar una API que envíe datos relevantes, implemente estadísticas y garantice la seguridad para el acceso exclusivo del administrador.

- Teniendo en cuenta lo anterior, y el trabajo que se debe hacer para más de un rol, se estima un total de 1-2 semanas.

4. Una vez clasificadas las funcionalidades, realizar el diagrama y especificación de caso de uso del administrador y el usuario.

- **Caso de uso 1: Consultar Pedidos**

Actores: Administrador

Descripción/Goal: Permite organizar los pedidos con la información del cliente, el producto, estado del pedido, fecha de entrega y Número de pedido, y consultarla.

Flujo Básico:

1. El administrador accede a la sección "Consultar Pedidos"
2. El sistema lee la base de datos

{Filtrado de pedidos}

3. El administrador ingresa el atributo por el que organizar
4. El sistema filtra los pedidos

{Mostrar resultados}

5. El sistema genera una tabla con la información del pedido
6. El sistema muestra la tabla con los pedidos y su información

Precondiciones:

Deben existir pedidos previos o en curso

Postcondiciones:

Debe existir una visualización con los pedidos deseados

- **Caso de uso 2: Registrar Pedidos**

Actores: Administrador o cliente

Descripción/Goal: Permite registrar los pedidos con la información del cliente, el producto, estado del pedido, fecha de registro y Número de pedido.

Flujo Básico:

1. El usuario selecciona la opción "Registrar pedido"

{Mostrar formulario}

2. El sistema muestra un formulario a llenar
3. El sistema muestra el catálogo de productos con sus precios
4. El usuario llena los datos
 - a. Información de quien hace el pedido
 - b. Producto
 - c. Fecha de registro

{Completar información}

5. El sistema asigna el valor del producto al valor del pedido
6. El sistema genera un ID único para el pedido
7. El sistema asigna el estado del pedido a "Pendiente"
8. El usuario finaliza el proceso de registro

{Guardar el pedido}

9. El sistema registra el pedido con su información
10. El sistema muestra un mensaje de confirmación

Flujo alternativo:

En **{Completar información}** si el producto seleccionado no está en stock, mostrar mensaje "Producto no disponible", retomar en **{Mostrar formulario}**

Precondiciones:

Deben existir productos en Stock

Postcondiciones:

El pedido queda registrado en la base de datos

- Caso de uso 3: Actualizar Pedido

Actores: Administrador

Descripción/Goal: Permite modificar el estado del pedido.

Flujo Básico:

1. El administrador selecciona un pedido
2. El administrador selecciona la opción "Actualizar pedido"

{Actualizar estado}

3. El sistema muestra la información actual del pedido
4. El administrador actualiza el estado del pedido
 - a. Pago
 - b. Enviado

{Guardar estado}

5. El sistema guarda el cambio
6. El sistema muestra un mensaje de confirmación

Flujo alternativo A: Estado Enviado

En **{Guardar estado}** si el estado nuevo es "Enviado", actualizar el inventario, retomar en **{Guardar estado}**

Flujo alternativo B: Estado Pago

En **{Guardar estado}** si el estado nuevo es "Pago", registrar la transacción, retomar en **{Guardar estado}**

Precondiciones:

Deben existir el pedido en la base de datos

Postcondiciones:

El pedido debe tener un nuevo estado en el registro en la base de datos

- Caso de uso 4: Controlar Inventario

Actores: Administrador o sistema (desde el pedido)

Descripción/Goal: Permite al actor gestionar el inventario, viendo la disponibilidad, y actualizarlo

Flujo Básico:

1. El actor accede a la sección “inventario”

{Actualizar información}

2. El sistema muestra los productos disponibles y su stock
3. El actor actualiza la información
 - a. Disponibilidad
 - b. Precio
 - c. Cantidad

{Guardar modificaciones}

4. El sistema guarda los cambios
5. El sistema muestra un mensaje de confirmación

Precondiciones:

El administrador debe tener acceso único

Postcondiciones:

La vista del inventario debe reflejar cualquier modificación hecha

- Caso de uso 5: Registrar transacciones

Actores: Administrador

Descripción/Goal: Permite al administrador llevar un control de las transacciones financieras del negocio

Flujo Básico:

1. El actor accede a la sección “Finanzas”

{Información de transacción}

2. El sistema solicita la información de la transacción
3. El actor proporciona la información
 - a. Naturaleza de la transacción (ingreso, gasto)
 - b. ID de transacción
 - c. Destinatario o remitente
 - d. Monto

- e. Descripción
- 4. El sistema guarda la transacción

{Actualizar balance}

- 6. El sistema actualiza el balance financiero
- 7. El sistema muestra un mensaje de confirmación

Precondiciones:

Debe existir un costo o ingreso asociado

Postcondiciones:

Debe quedar registrado en la base de datos y reflejado en el balance

- **Caso de Uso 6: Ver estadísticas**

Actores: Administrador

Descripción/Goal: Permite al administrador analizar el rendimiento del negocio mediante estadísticas y visualizaciones basadas en datos históricos.

Flujo Básico:

- 1. El actor accede a la sección "Estadísticas" desde el menú principal.
- 2. El sistema procesa los datos históricos de transacciones.
- 3. El sistema presenta una visión general con los datos más importantes del cliente, como productos más vendidos o monto total ganado hasta la fecha.
- 4. El sistema permite al administrador exportar los datos en formato PDF o Excel.

Precondiciones:

- Debe existir información histórica almacenada en la base de datos.
- El actor debe tener permisos de administrador para acceder a esta sección.

Postcondiciones:

- Las estadísticas generadas quedan disponibles para consulta futura.

- El administrador puede realizar exportaciones o tomar decisiones basadas en la información.

- **Caso de Uso 7: Ver el Catálogo de Productos**

Actores: Usuario, Sistema.

Descripción/Goal: Permite al usuario explorar el catálogo de productos disponibles, organizados por categorías y con opciones de búsqueda y filtrado.

Flujo Básico:

1. El usuario accede al sistema (con o sin iniciar sesión).
2. El sistema muestra la página principal con el catálogo de productos.
3. El usuario navega por las categorías o utiliza la barra de búsqueda para encontrar productos específicos.
4. El sistema muestra los productos según la categoría o los términos de búsqueda ingresados.
5. El usuario selecciona un producto para ver detalles como descripción, precio, y opciones de personalización (si aplican).

Flujo Alternativo:

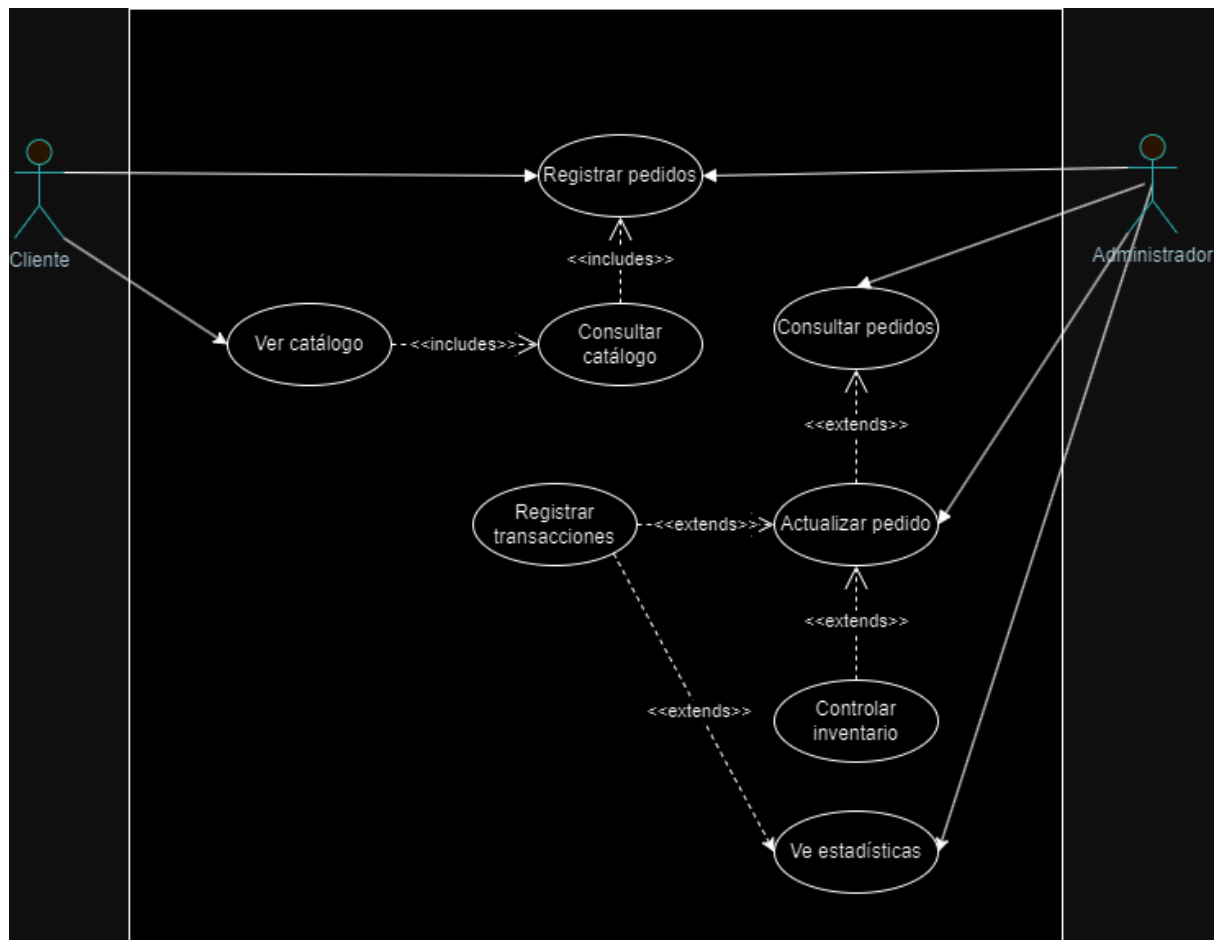
- Si no se encuentra ningún producto que coincida con la búsqueda o filtros, el sistema muestra un mensaje indicando que no hay resultados y sugiere explorar otras categorías.

Precondiciones:

- El sistema debe tener un catálogo actualizado de productos disponibles.
- El usuario debe tener acceso a internet para interactuar con la plataforma.

Postcondiciones:

- El usuario obtiene información detallada sobre los productos disponibles en el sistema.



5. por cada integrante del equipo, hacer uso de la plantilla de historia de usuario propuesta por el instructor y llenarla con alguna funcionalidad que quieran, **no repetir en el mismo equipo...**

Historia de Usuario #1: Gestión de pedidos (Luisa)

Anexo de Documentos Relacionados:

En el proceso de levantamiento de requerimientos se identificó que el cliente tiene dificultades para organizar los pedidos y mantener un registro claro del estado de los mismos (Pendiente, Pagado, Enviado, Completado). Esto genera errores en la gestión de entregas y pagos. El sistema debe abordar estas necesidades con un módulo dedicado a la gestión de pedidos.

Descripción conceptual

Módulo	Gestión de pedidos
Descripción de la(s) funcionalidad(es) requerida(s):	<ul style="list-style-type: none">• El sistema debe permitir al administrador registrar manualmente un pedido ingresando datos como cliente, productos, cantidad, estado del pedido y fecha de entrega.• Los pedidos deben actualizarse automáticamente al cambiar su estado, reflejándose en el inventario correspondiente.• Notificaciones deben enviarse al cliente (por WhatsApp) y al administrador (en la interfaz y/o WhatsApp) al realizar cambios significativos como confirmación de pedido o cambio de estado a "Enviado".• El cliente debe poder consultar el estado de su pedido desde un enlace generado automáticamente.

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/pedidos	GET	200: Solicitud exitosa. 500: Error interno del servidor. 404: Pedido no encontrado.
Descripción		
Obtener la lista de pedidos o un pedido específico.		
Datos de entrada	Datos de salida	
200 Unset <pre>{ "accion": "listar_pedidos" }</pre>	200 Unset <pre>{ "codigo": 200, "datos": [{ "id_pedido": 101, "total": 176000, "estado": "pendiente" }] }</pre>	
404 Unset <pre>{ "accion": "obtener_pedido", "id_pedido": 999 }</pre>	404 Unset <pre>{ "codigo": 404, "mensaje": "Pedido no encontrado.", "datos": null }</pre>	
500 Unset <pre>{ "accion": "listar_pedidos" }</pre>	500 Unset <pre>{ "codigo": 500, "mensaje": "Error interno del servidor. Intente nuevamente más tarde.", "datos": null }</pre>	

URL	Método	Código html
localhost:8080/pedidos	POST	200 (éxito), 500 (error interno del servidor)
Descripción		
Registra un nuevo pedido en el sistema con los datos del cliente, productos solicitados y fecha de entrega.		
Datos de entrada	Datos de salida	
200	200	
Unset { "accion": "registrar_pedido", "id_cliente": 201, "productos": [{"id_producto": 1, "cantidad": 2}], "fecha_entrega": "2024-12-25", "estado": "pendiente" }	Unset { "codigo": 200, "mensaje": "Pedido registrado exitosamente", "datos": { "id_pedido": 101, "total": 176000 } }	
500	500	
Unset { "id_cliente": 201, "productos": [{"id_producto": 1, "cantidad": 2}], "fecha_entrega": "2024-12-25", "estado": "pendiente" }	Unset { "codigo": 500, "mensaje": "Error interno del servidor. Intente nuevamente más tarde.", "datos": null }	

URL localhost:8080/pedidos/{id_pedido}	Método Update	Código html 200 (éxito), 404 (pedido no encontrado), 500 (error interno del servidor)
Descripción Permite actualizar el estado o los productos asociados a un pedido.		
Datos de entrada 200:	Datos de salida 200:	

Unset <pre>{ "accion": "actualizar_pedido", "estado": "completado" }</pre>	Unset <pre>{ "codigo": 200, "mensaje": "Pedido actualizado exitosamente", "datos": { "id_pedido": 101 } }</pre>
404 Unset <pre>{ "accion": "actualizar_pedido", "estado": "cancelado" }</pre>	404 Unset <pre>{ "codigo": 404, "mensaje": "Pedido no encontrado.", "datos": null }</pre>
500 Unset <pre>{ "accion": "actualizar_pedido", "estado": "completado" }</pre>	500 Unset <pre>{ "codigo": 500, "mensaje": "Error interno del servidor. Intente nuevamente más tarde.", "datos": null }</pre>

URL	Método	Código html
localhost:8080/pedidos/{id_pedido}	Delete	200 (éxito), 404 (pedido no encontrado), 500 (error interno del servidor)
Descripción		
Elimina un pedido existente según su ID.		
Datos de entrada	Datos de salida	
200: Unset Ninguno	200:	

	Unset <pre>{ "codigo": 200, "mensaje": "Pedido eliminado exitosamente", "datos": { "id_pedido": 101 } }</pre>
404 Unset Ninguno	404 Unset <pre>{ "codigo": 404, "mensaje": "Pedido no encontrado.", "datos": null }</pre>
500 Unset Ninguno	500 Unset <pre>{ "codigo": 500, "mensaje": "Error interno del servidor. Intente nuevamente más tarde.", "datos": null }</pre>

Frontend

Interacción esperada:

- El administrador puede visualizar un listado de pedidos en formato tabla, ordenados por fecha de creación o estado.
- Al seleccionar un pedido, se despliega un detalle con información específica, incluyendo cliente, productos, total y estado.
- Botones para:
 - Cambiar el estado del pedido (marcar como completado o cancelado).
 - Editar productos en el pedido.
 - Generar notificaciones automáticas para el cliente.

Mockups/Prototipos:

- **Pantalla de lista de pedidos:**
Un diseño con una tabla principal que incluye columnas para el ID del pedido, y un buscador por las categorías del detalle del pedido, (nombre del cliente, productos del pedido)
- **Detalle del pedido:**
Vista emergente o página específica que muestra:

- Información del cliente.
- Productos solicitados (nombre, cantidad, precio).
- Total del pedido.
- Botones de acción (modificar datos, eliminar pedido).

Este sería el mockup de la interfaz para ver el detalle de un producto, incluye los elementos de la lista de pedidos con la opción de búsqueda, la ventana del detalle de pedido y un espacio para poner gráficas relacionadas con los datos, o por ejemplo un calendario con las fechas de entrega o de compra marcadas.



Y este para agregar un pedido nuevo, con los campos del detalle del producto abiertos.



Flujo visual y eventos:

1. **Registro de pedido:**
 - Formulario con campos obligatorios (cliente, productos, fecha de entrega).
 - Mostrar un mensaje de confirmación tras enviar el formulario.
2. **Actualización del estado del pedido:**
 - Cambiar entre estados (pendiente, en proceso, completado, cancelado) con un clic.
 - Mostrar una notificación emergente con la confirmación de la acción.
3. **Notificación automática al cliente:**
 - Enviar un correo o mensaje vía WhatsApp con los detalles del pedido y su estado.
 - Confirmar visualmente el envío de la notificación al administrador.

Notas Adicionales

Consideraciones técnicas:

- *Frameworks sugeridos: Node.js para el backend y React.js o Vue.js para el frontend.*
- *Base de datos relacional como MySQL o PostgreSQL para manejar los datos de pedidos y productos.*
- *Sistema de notificaciones con integración de servicios como Twilio o correo electrónico.*
- *Validación en el backend para garantizar que los datos enviados sean correctos (productos existentes, cliente registrado).*

Historia de Usuario #2: Control de inventario

Anexo de Documentos Relacionados:

En el proceso de levantamiento de requerimientos se identificó que el cliente tiene dificultades para mantener un registro claro y actualizado de su inventario, ya que lo maneja en una hoja de excel que no actualiza regularmente, además de que no tiene presente cuando cuenta con bajo stock . Esto genera errores en la gestión de Stock y catálogo. El sistema debe abordar estas necesidades con un módulo dedicado a la gestión de inventario.

Descripción conceptual

Módulo	Gestión de inventario
Descripción de la(s) funcionalidad(es) requerida(s):	<ul style="list-style-type: none">• El sistema debe permitirle al administrador consultar el inventario que tiene disponible en el momento (su stock) .• El sistema debe actualizar la información del stock en el inventario automáticamente luego de completada una venta.• El sistema debe permitirle al administrador hacer modificaciones al stock, específicamente añadir stock .• El sistema debe permitir al administrador eliminar productos y que estos desaparezcan del catálogo

	<p>automáticamente.</p> <ul style="list-style-type: none"> El sistema debe generar una alerta cuando el stock de un producto se esté agotando.
--	---

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/Inventario	GET	200
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los productos y su stock		
Datos de entrada <pre>Unset { "accion": "listar_productos" }</pre>		Datos de salida <pre>Unset { "status": "success", "data": [{ "Nombre del producto": "Pulsera", "Precio": 15000 "Cantidad en Stock": 10 "url": "imagen.html" }] }</pre>

URL	Método	Código html
localhost:8080/inventario/Nombre_del_producto	PATCH	200 422
Caso de uso técnico Al completar una venta, debe retornar código 200 y disminuir el stock en el inventario		
200: Datos de entrada <pre>Unset { "status": "success", "data": {</pre>	200: Datos de salida <pre>Unset { "status": "success",</pre>	

<pre> "Cantidad en Stock": -=1 } }</pre>	<pre> "data": { "Nombre del producto": "Pulsera", "Precio": 15000 "Cantidad en Stock": 15-1 "url": "imagen.html" } }</pre>
422: <pre> Unset { "status": "success", "data": { "Cantidad en Stock": -=1 } }</pre>	422: <pre> Unset { "status": "unprocessed", "message": "No existe el parámetro <Cantidad en Stock>" }</pre>

URL	Método	Código html
localhost:8080/inventario/Nombre_del_producto	PATCH	200 422
Caso de uso técnico Al modificar, debe retornar código 200 y actualizar los cambios en el inventario		
Datos de entrada 200: <pre>Unset { "status": "success", "data": { "Cantidad en Stock": 15 } }</pre>	Datos de salida 200: <pre>Unset { "status": "success", "data": { "Nombre del producto": "Pulsera", "Precio": 15000 "Cantidad en Stock": 15 "url": "imagen.html" } }</pre>	

	<pre> }</pre>
422: <pre> Unset { "status": "success", "data": { "Cantidxd en Stock": 15 } }</pre>	422: <pre> Unset { "status": "unprocessed", "message": "No existe el parámetro <Cantidxd en Stock>" }</pre>

URL	Método	Código html
localhost:8080/inventario/Nombre_del_producto	DELETE	200 404
Caso de uso técnico Al borrar, debe retornar código 200 y modificar la lista del inventario		
200: Datos de entrada	200: Datos de salida <pre> Unset { "codigo": 200, "mensaje": "Pedido eliminado exitosamente", "datos": { "Nombre del producto": "Pulsera" } }</pre>	
404:	404: <pre> Unset { "codigo": 404, "mensaje": "Producto no encontrado.", "datos": null }</pre>	

URL	Método	Código html
localhost:8080/inventario/Nombre_del_producto/advertencia	POST	200
Caso de uso técnico Debe alertar cuando la cantidad en stock sea menor al limite establecido		
Datos de entrada 200:	Datos de salida 200: <pre>Unset { "codigo": 200, "mensaje": "Alerta de bajo stock enviada correctamente.", "datos": { "Nombre del Producto": Pulsera, "cantidad_actual": 2, "umbral_minimo": 5 } }</pre>	

Frontend

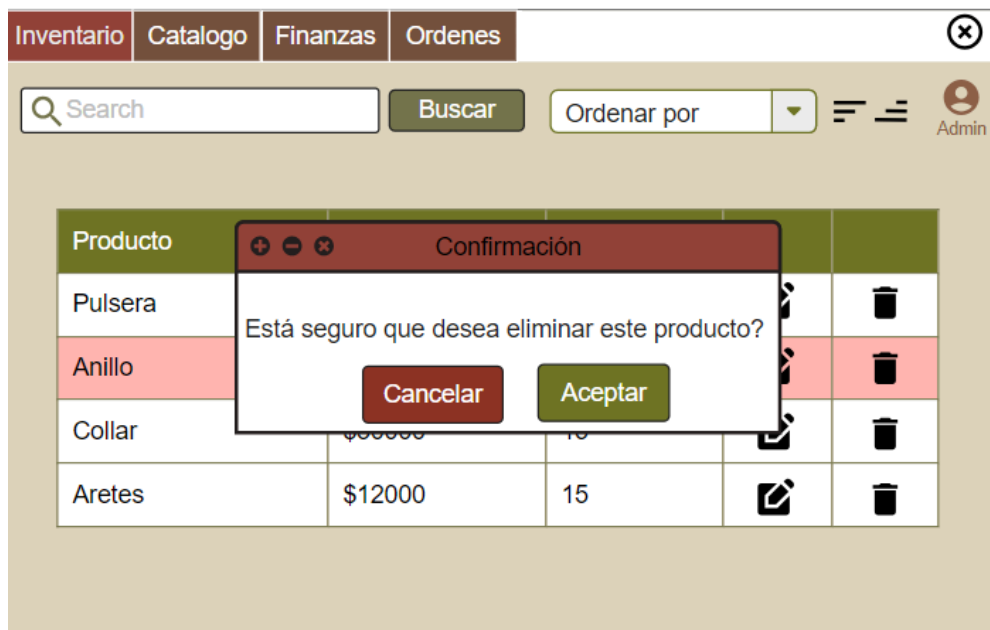
Interacción esperada:

- El administrador podrá visualizar un listado de productos en formato tabla con columnas para:
 - Nombre del producto.
 - Precio.
 - Stock disponible.
- El administrador contará con una barra de búsqueda para localizar productos por nombre.
- El administrador contará con botones para filtrar/organizar la visualización por:
 - Cantidad en stock de menor a mayor y viceversa.
 - Precio de menor a mayor y viceversa.
- Entre la visualización de la tabla de los productos, se resaltará visualmente en rojo los productos con bajo stock
- Botones y acciones adicionales:
 - Añadir Stock, que abre una pestaña con un espacio para la nueva cantidad y muestra un mensaje de guardado exitosamente

- Eliminar producto, cada producto en la vista en la tabla contará con un botón de eliminado, que una vez oprimido requerirá de una confirmación

Mockups/Prototipos:





Flujo visual y eventos:

- Consulta del inventario:**
 - La tabla principal muestra los productos actuales con sus datos.
 - Al buscar o filtrar, la tabla se actualiza dinámicamente.
- Añadir stock:**
 - Al seleccionar un producto y hacer clic en "Añadir Stock", se despliega un formulario emergente.
 - Al enviar el formulario, se actualiza el stock y se muestra un mensaje de éxito.
- Eliminar producto:**
 - Al hacer clic en "Eliminar Producto", se pide confirmación al administrador.
 - Tras la confirmación, se elimina el producto del inventario mostrando un mensaje de éxito.
- Alertas de bajo stock:**
 - Los productos con bajo stock se resaltan en la tabla.
 - Se envía una notificación
- Actualización automática del inventario:**
 - Después de completar una venta, el stock del producto afectado se ajusta automáticamente.

Notas adicionales

Consideraciones técnicas:

- Framework frontend:** React.js o Vue.js .
- Backend:** Integrado con una API RESTful para realizar operaciones como consultar, actualizar o eliminar productos.
- Alertas:** Implementar notificaciones visuales con librerías como Toastr o SweetAlert.

Historia de Usuario #3: Reportes y estadísticas

Anexo de Documentos Relacionados:

Durante el levantamiento de requerimientos, el cliente mencionó la necesidad de contar con reportes automáticos y gráficos que les permitan tomar decisiones basadas en datos históricos y tendencias. Actualmente, carecen de herramientas que les ofrezcan visualizaciones claras y procesables sobre el rendimiento del negocio.

Descripción conceptual

Módulo	Gestión de inventario
Descripción de la(s) funcionalidad(es) requerida(s):	<ul style="list-style-type: none">El sistema debe permitir al administrador generar reportes de ventas, gastos e ingresos para cualquier rango de fechas definido.El sistema debe permitir exportar los reportes generados en formato PDF o Excel.El sistema debe proporcionar un resumen mensual automático que se envíe por correo al administrador.

Descripción técnica

Backend

URL	Método	Código http
localhost:8080/reporte?fecha_inicio={}&fecha_fin={}	GET	200 500 404
Descripción		
Al realizar una solicitud, el sistema genera y entrega un reporte detallado de métricas y estadísticas relevantes relacionadas con el rendimiento de eventos, usuarios y ventas. Los datos se devuelven en formato JSON para su fácil procesamiento.		
Datos de entrada	Datos de salida	
200	200	
<pre>Unset { "accion": "ver_reportes", "filtro": { "fecha_inicio":</pre>	<pre>Unset {</pre>	

<pre>"2024-12-01", "fecha_fin": "2024-12-31" } }</pre>	<pre>"status": "success", "code": 200, "data": { "productos_vendidos": 1200, "ingresos_totales": 1500000, "mejor_producto": { "nombre": "Artesanía 1", "vendidos": 500, "ingresos": 750000 } }</pre>
500 Unset <pre>{ "accion": "ver_reportes", "filtro": { "fecha_inicio": "2024-12-01", "fecha_fin": "2024-12-31" } }</pre>	500 Unset <pre>{ "status": "error", "code": 500, "message": "Error interno del servidor. Intente nuevamente más tarde." }</pre>
404 Unset <pre>{ "accion": "ver_reportes", "filtro": { "fecha_inicio": "2024-12-01", "fecha_fin": "2024-12-31" } }</pre>	404 Unset <pre>{ "status": "error", "code": 404, "message": "No se encontraron reportes para el rango de fechas proporcionado." }</pre>

URL	Método	Código html
localhost:8080/catálogo+productos/nombre?nombre_producto&categoria={}&Precio_Máx={}&marca{}&stock_min={}	POST	200 414 503
Descripción Se proporcionan más parámetros para el método de búsqueda en la base de datos. Retorna una lista de productos condicionados a la búsqueda que hizo el usuario.		
Datos de entrada		Datos de salida
200 <pre>Unset { "accion": "ver_catalogo", "filtros": { "categoria": "", "precio_max": --, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } }</pre>		200 <pre>Unset { "codigo": 200, "mensaje": "Catálogo obtenido con éxito", "datos": [{ "id_producto": 1, "nombre": "Bufanda tejida", "categoria": "Textiles", "marca": "Manos Creadoras", "precio": 25.0, "stock": 15, "imagen_url": "img.jpg", "descripcion": "Bufanda hecha a mano con materiales 100% naturales." }, { "id_producto": 2, "nombre": "Camino de mesa artesanal", "categoria": "Textiles", "marca": "Arte Vivo", "precio": 30.0, "stock": 8, "imagen_url": "img.jpg", "descripcion": "Decorativo para mesa elaborado con tejidos tradicionales." }, { "id_producto": 3, "nombre": "Bolsa de algodón tejida", "categoria": "Textiles", "marca": "Raíces Artesanales", "precio": 40.0, "stock": 10, "imagen_url": "img.jpg", "descripcion": "Bolsa ecológica hecha con algodón 100% orgánico." }] }</pre>

	<pre> }] } </pre>
414 <pre> Unset { "accion": "ver_catalogo", "filtros": { "categoria": "Categoría con un nombre extremadamente largo que excede los límites", "precio_max": 1000000000, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } } </pre>	414 <pre> Unset { "codigo": 414, "mensaje": "La URI es demasiado larga. Por favor, reduzca los parámetros de búsqueda.", "datos": null } </pre>
503 <pre> Unset { "accion": "ver_catalogo", "filtros": { "categoria": "Textiles", "precio_max": 50, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } } </pre>	503 <pre> Unset { "codigo": 503, "mensaje": "El servicio no está disponible en este momento. Inténtelo más tarde.", "datos": null } </pre>

URL	Método	Código html
-----	--------	-------------

localhost:8080/users/catalogo+producto/favoritos/	POST	200 404
Descripción Para este apartado el usuario tendrá de añadir productos favoritos para facilitar la búsqueda dentro del catálogo.		
Datos de entrada 200: <pre>Unset { "accion": "agregar_favorito", "id_usuario": 101, "id_producto": 3 }</pre>	Datos de salida 200: <pre>Unset { "codigo": 200, "mensaje": "Producto agregado a favoritos exitosamente.", "datos": { "id_usuario": 101, "id_producto": 3, "fecha_favorito": "2024-12-19T10:00:00Z" } }</pre>	
404 <pre>Unset { "accion": "agregar_favorito", "id_usuario": 101, "id_producto": 3 }</pre>	404 <pre>Unset { "codigo": 404, "mensaje": "Producto no encontrado.", "datos": null }</pre>	

Frontend

Interacción esperada:

- Al ingresar a la página de reportes y estadísticas, el usuario puede visualizar un panel interactivo que muestra gráficos y tablas dinámicas con las métricas más relevantes.

Filtros:

- Campos de filtro como:
 - Rango de fechas
 - Producto
- Un botón llamado **"Aplicar filtros"** actualiza el catálogo en tiempo real.

Mockups/Prototipos:

Pantalla principal: Un diseño limpio con un encabezado que incluye:

- Ícono de filtro.

Resumen ejecutivo: Una tarjeta cuadrada que incluye

- Producto más vendido.
- Ingresos obtenidos.

Flujo visual y eventos:

Carga inicial:

- Mostrar un spinner mientras los datos del catálogo se cargan desde el servidor.

Aplicación de filtros:

- Mostrar un mensaje temporal **"Aplicando filtros..."** al hacer clic en **"Aplicar filtros"**.
- Actualizar el resumen y las gráficas dinámicamente sin recargar la página.

Error de servidor (503):

- Mostrar una notificación emergente o modal indicando **"El servicio no está disponible en este momento. Intente más tarde."**

Ejemplo del apartado:

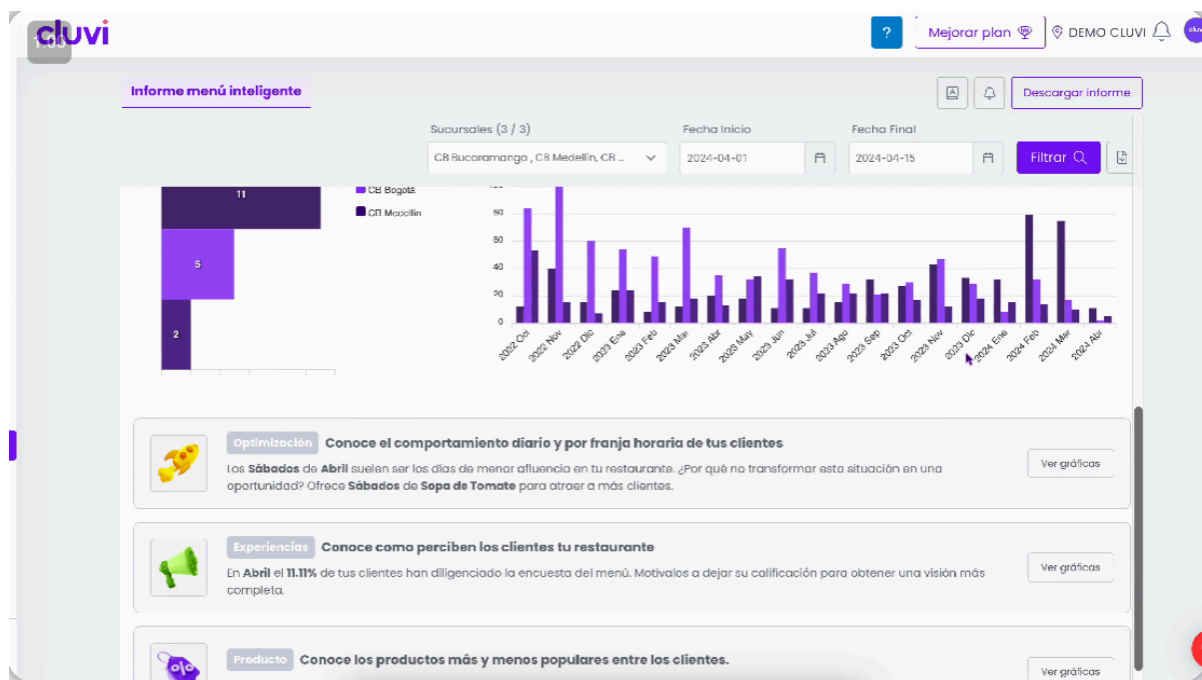


Imagen obtenida de <https://www.cluvi.com/>

Historia de Usuario #4: Filtros (Andres)

Anexo de Documentos Relacionados:

Dado que es un sistema de compra y venta de productos, en el levantamiento de requerimientos se vio la necesidad de que los usuario, más allá de solo saber los productos disponibles, también tengan la posibilidad de hacer diferentes filtros para hacer más cómoda la experiencia en el sistema.

Descripción conceptual

Módulo	Catálogo en línea
Descripción de la(s) funcionalidad(es) requerida(s):	<ul style="list-style-type: none">• Permitir a los usuarios visualizar un catálogo de productos organizados en categorías como artesanías y accesorios para decoración.• Cada producto debe mostrar: nombre, imagen, precio, descripción breve y disponibilidad.• Filtrar y buscar productos por categorías, precio o palabras clave.• El usuario puede marcar productos como favoritos para acceder rápidamente a ellos en el futuro.

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/catalogo+productos	GET	200 500 404
Descripción Al presionar un botón o algo que estimule el sistema para mostrar el catálogo de productos. Se retorna una lista de objetos en formatos JSON con los productos por defecto.		
Datos de entrada 200 <pre>Unset { "accion" : "ver_catalogo" }</pre>	Datos de salida 200 <pre>Unset { "status": "success",</pre>	

	<pre> "code": 200, "data": [{ "id":1, "nombre":"jarron de ceramica", "precio": 82000, "imagen": "url_img_jarron.jpg" "descripcion":"Jarrón artesanal ideal para decoración de interiores.", "disponibilidad": "en stock", "cantidad":3, "categoria": "ceramica" }, { "id": 2, "nombre": "Tejido artesanal", "precio": 12000, "imagen": "url_imagen_2.jpg", "descripcion": "Manta tejida a mano con lana natural de oveja.", "disponibilidad": "en stock" "cantidad":20, "categoria" : "textil" }] } </pre>
500 <pre> Unset { "accion" : "ver_catalogo"} </pre>	500 <pre> Unset { "status": "error", "code": 500, "message": "Error interno del servidor. No se pudo cargar el catálogo en este momento. Intente nuevamente más tarde." } </pre>
404 <pre> Unset { "accion" : "ver_catalogo"} </pre>	404 <pre> Unset {"status":"error", "code":404} </pre>

URL	Método	Código html
localhost:8080/catálogo+productos/nombre?nombre_producto&categoria={}&Precio_Máx={}&marca{}&stock_min={}	POST	200 414 503
Descripción Se proporcionan más parámetros para el método de búsqueda en la base de datos. Retorna una lista de productos condicionados a la búsqueda que hizo el usuario.		
Datos de entrada		Datos de salida
200 <pre>Unset { "accion": "ver_catalogo", "filtros": { "categoria": "", "precio_max": --, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } }</pre>		200 <pre>Unset { "codigo": 200, "mensaje": "Catálogo obtenido con éxito", "datos": [{ "id_producto": 1, "nombre": "Bufanda tejida", "categoria": "Textiles", "marca": "Manos Creadoras", "precio": 25.0, "stock": 15, "imagen_url": "img.jpg", "descripcion": "Bufanda hecha a mano con materiales 100% naturales." }, { "id_producto": 2, "nombre": "Camino de mesa artesanal", "categoria": "Textiles", "marca": "Arte Vivo", "precio": 30.0, "stock": 8, "imagen_url": "img.jpg", "descripcion": "Decorativo para mesa elaborado con tejidos tradicionales." }, { "id_producto": 3, "nombre": "Bolsa de algodón tejida", "categoria": "Textiles", "marca": "Raíces Artesanales", "precio": 40.0, "stock": 10, "imagen_url": "img.jpg", "descripcion": "Bolsa ecológica hecha con algodón 100% orgánico." }] }</pre>

	<pre> }] } </pre>
414 <pre> Unset { "accion": "ver_catalogo", "filtros": { "categoria": "Categoría con un nombre extremadamente largo que excede los límites", "precio_max": 1000000000, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } } </pre>	414 <pre> Unset { "codigo": 414, "mensaje": "La URI es demasiado larga. Por favor, reduzca los parámetros de búsqueda.", "datos": null } </pre>
503 <pre> Unset { "accion": "ver_catalogo", "filtros": { "categoria": "Textiles", "precio_max": 50, "stock_min": 1 }, "orden": { "campo": "precio", "direccion": "asc" } } </pre>	503 <pre> Unset { "codigo": 503, "mensaje": "El servicio no está disponible en este momento. Inténtelo más tarde.", "datos": null } </pre>

URL	Método	Código html
-----	--------	-------------

localhost:8080/users/catalogo+producto/favoritos/	POST	200 404
Descripción Para este apartado el usuario tendrá de añadir productos favoritos para facilitar la búsqueda dentro del catálogo.		
Datos de entrada 200: <pre>Unset { "accion": "agregar_favorito", "id_usuario": 101, "id_producto": 3 }</pre>	Datos de salida 200: <pre>Unset { "codigo": 200, "mensaje": "Producto agregado a favoritos exitosamente.", "datos": { "id_usuario": 101, "id_producto": 3, "fecha_favorito": "2024-12-19T10:00:00Z" } }</pre>	
404 <pre>Unset { "accion": "agregar_favorito", "id_usuario": 101, "id_producto": 3 }</pre>	404 <pre>Unset { "codigo": 404, "mensaje": "Producto no encontrado.", "datos": null }</pre>	

Frontend

Interacción esperada:

- Al ingresar a la página, el usuario visualiza el catálogo de productos en un diseño de cuadrícula (grid) con tarjetas individuales para cada producto.
- Cada tarjeta contiene:
 - Imagen del producto.
 - Nombre del producto.
 - Precio.
 - Marca.

- Botón para agregar a favoritos.
 - Botón para ver detalles.
- Un filtro y un orden dinámico en la parte superior de la página permiten al usuario ajustar los resultados mostrados.

Filtros:

- Campos de filtro como:
 - Categoría (dropdown).
 - Precio máximo (campo numérico).
 - Stock mínimo (campo numérico).
- Un botón llamado **"Aplicar filtros"** actualiza el catálogo en tiempo real.

Favoritos:

- Al hacer clic en el botón **"Agregar a Favoritos"**, el producto se agrega a la lista de favoritos y el botón cambia a **"Quitar de Favoritos"**.

Detalle de Producto:

- Al hacer clic en el botón **"Ver Detalles"**, se abre un modal o se redirige a una página específica con información más detallada del producto.

Mockups/Prototipos:

Pantalla principal: Un diseño limpio con un encabezado que incluye:

- Campo de búsqueda global.
- Ícono de favoritos.
- Ícono de usuario o carrito de compras.

Tarjeta de producto: Una tarjeta cuadrada que incluye:

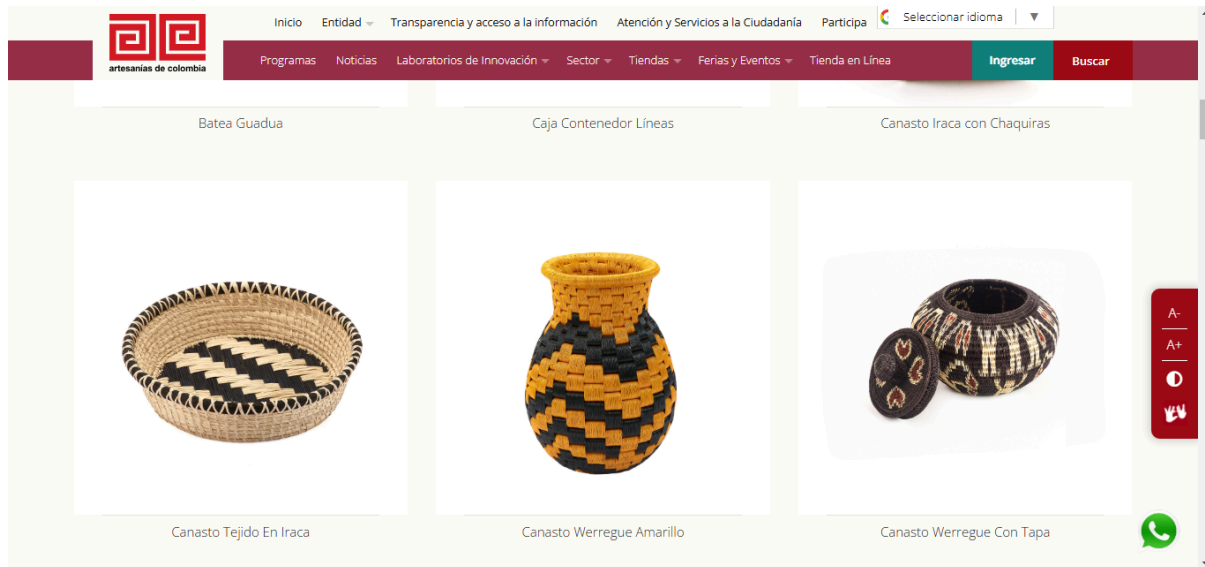
- Imagen destacada (centrada).
- Texto alineado debajo (nombre del producto, precio, marca).
- Botones interactivos.

Filtros: Un panel horizontal o barra lateral para filtrar resultados.

Para dar una idea más gráfica vamos a tomar el ejemplo del catálogo de la siguiente página Web:

<https://artesantiasdecolombia.com.co/PortalAC/Catalogo/CatalogoIndex.jsf>

Imagen:



Flujo visual y eventos:

Carga inicial:

- *Mostrar un spinner mientras los datos del catálogo se cargan desde el servidor.*

Aplicación de filtros:

- *Mostrar un mensaje temporal **"Aplicando filtros..."** al hacer clic en **"Aplicar filtros"**.*
- *Actualizar el catálogo dinámicamente sin recargar la página.*

Agregar a favoritos:

- *Cambiar el ícono del botón a una estrella llena al agregar un producto a favoritos.*
- *Mostrar un mensaje de confirmación **"Producto agregado a favoritos"**.*

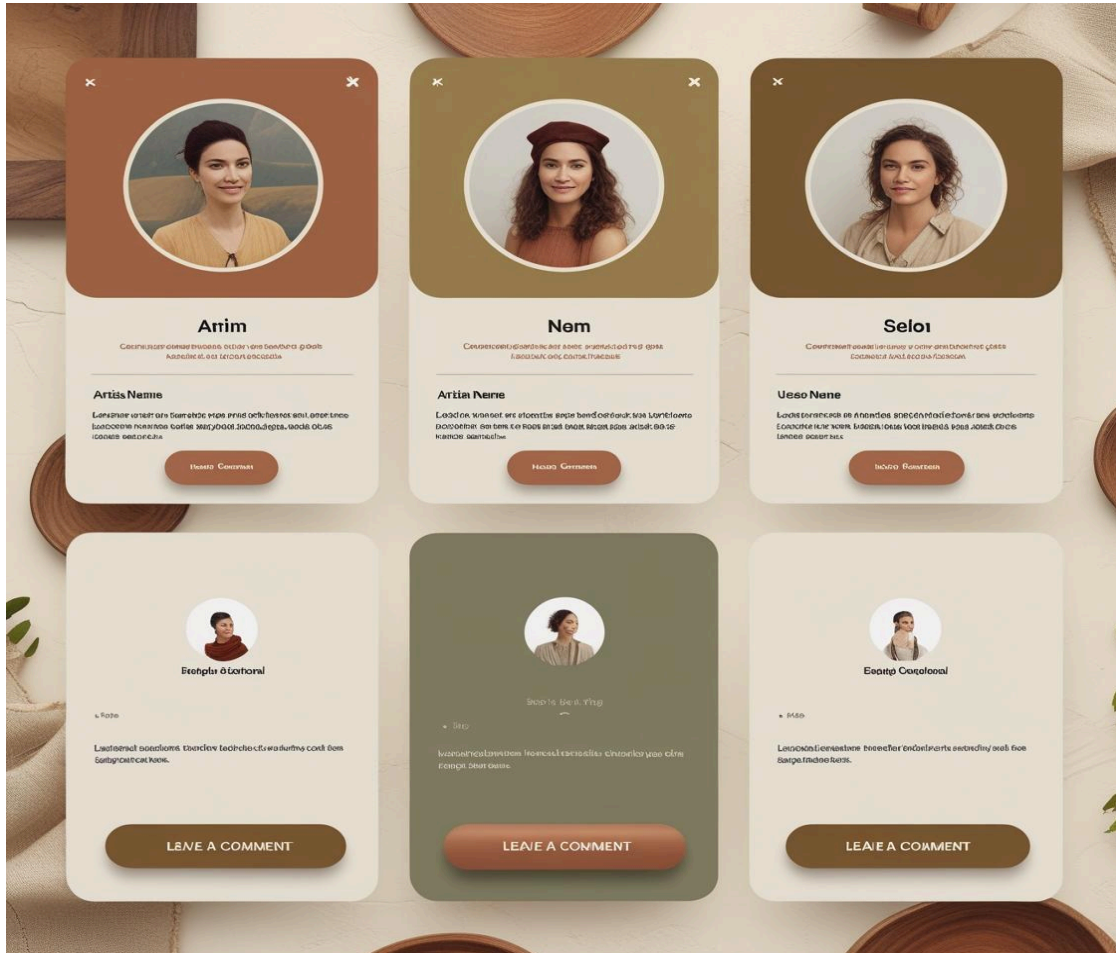
Error de servidor (503):

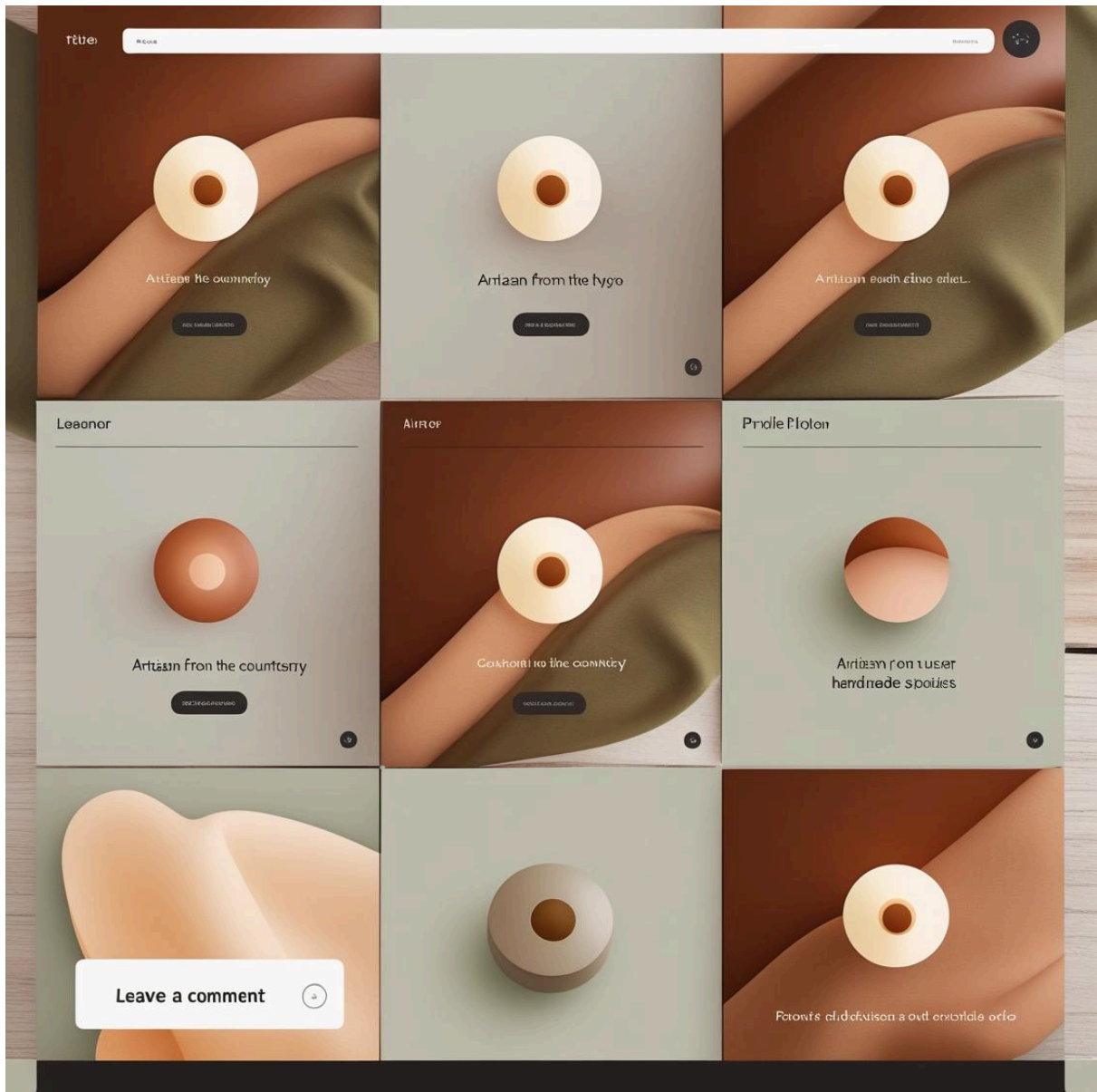
- *Mostrar una notificación emergente o modal indicando **"El servicio no está disponible en este momento. Intente más tarde."***

Detalle del producto:

- *Al hacer clic en **"Ver Detalles"**, mostrar:*
 - *Imagen ampliada del producto.*
 - *Descripción detallada.*
 - *Botón **"Agregar al carrito"** (si aplica).*

En este mockup se puede ver una lista de usuarios con información general de los mismos registrados en el sistema. En la segunda imagen se cree que es un catálogo de productos que alberga un buscador donde se puede hacer los diferentes tipos de filtros.





Imágenes generadas por Leonardo AI.

Notas Adicionales:

El desarrollo del catálogo requiere dependencias como HTML, CSS, y JavaScript, con Bootstrap para diseño responsivo y Vue.js o React para interacción dinámica. La API debe gestionar datos JSON y retornar mensajes claros, incluso en errores. Las imágenes se alojarán en un servicio externo (como AWS S3). Es crucial validar entradas en el servidor, usar notificaciones (e.g., SweetAlert2) y diseñar para navegadores modernos con enfoque "mobile-first". Los prototipos pueden elaborarse en Figma para mejorar la experiencia de usuario y gestionar la experiencia de usuario por ejemplo: paleta de colores o para hacer cierta funcionalidad, disminuir a lo máximo posible la cantidad de pasos.