



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

## **Taller #1**

**JULIAN ESTEBAN CADENA ROJAS**

**ROGER FABIAN BONILLA CARO**

**MATEO BECERRA PORRAS**

**JACEL THOMÁS ENCISO PINZÓN**

**Universidad Nacional De Colombia – Sede Bogotá**

**Facultad de Ingeniería**

**Grupo: 9**

**Curso: Ingeniería de software I**

## **Levantamiento de requerimientos: Conversación con ChatGPT del Grupo 2**

### **Punto #1 & 2 - Levantamiento de requerimientos:**

Se necesita un sistema que permita a la administradora gestionar su negocio de forma ágil y eficiente, tanto para el manejo de pedidos como para el control de inventario y la comunicación con los clientes. En este sistema, la administradora podrá registrar cada pedido incluyendo detalles como el nombre del cliente, los productos solicitados, el estado del pedido (Pendiente, Pagado, Enviado, Completado o Cancelado), las fechas importantes (de solicitud y entrega), el monto total, y el método de pago utilizado (como transferencia bancaria o efectivo). También será posible modificar pedidos en caso de cambios, visualizar rápidamente una lista de los pedidos recientes y recibir notificaciones automáticas sobre plazos importantes o actualizaciones de estado.

Los clientes, a su vez, recibirán confirmaciones de sus pedidos, así como actualizaciones sobre el estado de los mismos, preferiblemente a través de WhatsApp. Esto asegurará una comunicación rápida y efectiva, sin necesidad de implementar sistemas complejos.

En cuanto al inventario, el sistema debe permitir registrar productos con detalles como nombre, precio, cantidad disponible, descripción y fotos. Este inventario se actualizará automáticamente cada vez que se complete un pedido, y se generarán alertas cuando el stock de algún producto esté por debajo de un nivel crítico. Esto ayudará a mantener el negocio abastecido sin necesidad de supervisión constante.

El sistema también incluirá una sección para gestionar la información de los clientes. Se podrán registrar datos como nombre, contacto, dirección y un historial de pedidos para cada cliente. Esta información se almacenará automáticamente al procesar un pedido, y podrá ser modificada cuando sea necesario, permitiendo un manejo eficiente y personalizado de la relación con cada cliente.

Para facilitar la interacción con los clientes, se implementará un catálogo en línea donde puedan consultar los productos disponibles. Este catálogo será sencillo, con imágenes atractivas, nombres de los productos, precios y una breve descripción. Será accesible desde un enlace que los clientes podrán abrir fácilmente, sin necesidad de descargar aplicaciones.

Además, la administradora contará con un resumen financiero básico que registre los ingresos por ventas y los gastos relacionados con el negocio. Esto permitirá obtener una

visión clara de la ganancia neta, los productos más vendidos y la relación entre ingresos y gastos.

El diseño del sistema priorizará una interfaz simple y funcional. Para la administradora, la pantalla inicial mostrará un resumen con los pedidos pendientes, los productos con bajo stock y los últimos pedidos registrados. También habrá un menú que facilite el acceso a las diferentes secciones, como inventario, pedidos y resumen financiero. Para los clientes, el catálogo será visualmente agradable y fácil de navegar.

El sistema estará diseñado para funcionar principalmente en línea, asegurando que la administradora pueda acceder a él desde cualquier dispositivo, pero con la flexibilidad de realizar tareas básicas offline en caso de necesidad.

Por último, se priorizará la seguridad, garantizando que solo la administradora tenga acceso a la información sensible y asegurando el almacenamiento adecuado de los datos de clientes y transacciones. Se espera que la implementación inicial esté lista en aproximadamente un mes, con un presupuesto estimado de un millón de pesos colombianos con posibilidad de aumentar ligeramente. El diseño será modular, permitiendo añadir funciones adicionales en el futuro, como promociones, cupones o la gestión de un calendario de asesorías.

### **Requerimientos funcionales:**

#### **1. Gestión de pedidos**

- El sistema debe permitir crear, eliminar y editar los pedidos.
- Se debe incluir un apartado donde se pueda visualizar el estado del pedido.
- Se debe visualizar información general del propio pedido como el monto total del pedido, el medio de pago, la fecha del pedido y la fecha estimada de entrega.

#### **2. Gestión de inventario**

- El sistema debe permitir crear productos con su respectiva información relevante (nombre, id, precio, stock, etc.).
- Actualizar automáticamente el inventario al completar un pedido.
- Generar notificaciones hacia la administradora cuando el stock de un producto llega a niveles críticos.
- Visualización de productos en una lista organizada con detalles de disponibilidad y fáciles de entender.

#### **3. Gestión de clientes**

- Se permite el registro y almacenamiento de la información importante de los clientes como nombre, numero de contacto y dirección con posibilidad de modificación.

#### **4. Catálogo en línea:**

- El diseño del catálogo debe incluir imágenes grandes y atractivas para el cliente junto con la información relevante (precio y detalles).
- Acceso desde un enlace sin necesidad de descargar nada.
- Sincronización automática con el inventario.

**5. Notificaciones:**

- El sistema debe notificar a la administradora cuando el stock de un producto es bajo.
- El sistema debe notificar a los clientes sobre la confirmación del pedido y el estado del este mismo.
- Las notificaciones se deben hacer vía WhatsApp (prioridad) o Email (opción secundaria).

**6. Interfaz de usuario:**

- La administradora debe tener un resumen en el inicio del sistema para poder visualizar información relevante y de uso común para ella como:
  - o Pedidos recientes
  - o Productos con bajo stock
  - o Últimos pedidos registrados
- La interfaz para la administradora debe contener un menú con opciones para:
  - o Añadir y modificar productos
  - o Visualizar y gestionar pedidos
  - o Consultar el inventario y modificarlo
  - o Acceder a un resumen financiero
- La visualización de la interfaz para los clientes debe atractiva y debe contener información como confirmaciones de los pedidos.

**7. Gestión de asesorías**

- El usuario debe poder gestionar y agendar asesorías en un calendario de reservas.
- La administradora debe visualizar en un calendario todas las reservas de asesorías pendientes.

**8. Herramientas de promoción**

- Descuentos que los clientes pueden recibir bajo condiciones que la administradora determine.
- Cupones de descuento para los productos.
- Promociones y publicidad a clientes potenciales.

**Requerimientos no funcionales:**

**1. Interfaz y diseño**

- El sistema debe implementar una interfaz sencilla, intuitiva y fácil tanto para la administradora como para los clientes.
- Compatibilidad con dispositivos móviles y computadoras.

**2. Tecnología:**

- Aplicación basada en web para garantizar accesibilidad en diferentes dispositivos sin necesidad de descargas.
- Funcionalidad online como prioridad con opción de realizar tareas básicas en modo offline.

**3. Seguridad:**

- Control exclusivo a la administradora a la información sensible y de importancia para el negocio.
- Almacenamiento seguro de datos de los clientes y transacciones.



### **Punto #3 - Priorización de requerimientos:**

#### **Método MoSCoW**

##### **1. Must Have:**

- **Gestión de pedidos:**

- Tiempo estimado: 5 días**

- La gestión de pedidos se puede lograr mediante frameworks para backend como Node.js, para la base de datos se puede usar MySQL o postgresSQL y se podría incluir APIs para el procesamiento de pagos.

- **Gestión de inventario:**

- **Tiempo estimado: 5 días**

- La gestión de inventario se puede implementar utilizando frameworks backend como Node.js, con bases de datos como MySQL o PostgreSQL para almacenar y actualizar los niveles de stock. Además, se pueden integrar APIs para sincronización con proveedores externos y generar alertas automáticas cuando el inventario esté bajo.

- **Catálogo en línea:**

- Tiempo estimado: 8 días**

- El catálogo en línea se puede desarrollar utilizando frameworks frontend como React.js o Angular para una interfaz dinámica, mientras que el backend puede estar basado en Node.js con bases de datos como MySQL o PostgreSQL para almacenar la información de productos. Se pueden integrar APIs para facilitar la búsqueda, filtrado y actualización de productos en tiempo real.

- **Interfaz de usuario:**

- Tiempo estimado: 8 días**

- La interfaz de usuario se puede crear utilizando frameworks frontend como React.js o Vue.js para ofrecer una experiencia interactiva y dinámica. Estos frameworks permiten diseñar una interfaz intuitiva y responsiva, que se adapta a diferentes dispositivos. Se puede integrar con el backend a través de APIs para gestionar datos en tiempo real, como productos, pedidos e inventario.

- **Interfaz y diseño**

- **Seguridad**

## 2. Should Have:

- **Gestión de clientes:**

- Tiempo estimado: 2 días**

- La gestión de clientes puede optimizarse permitiendo que los usuarios se registren e inicien sesión a través de múltiples métodos de autenticación compatibles con CloudFlare, integrando APIs de terceros como Google, Facebook, y otras plataformas populares.

- **Tecnología**

## 3. Could Have:

- **Notificaciones:**

- **Tiempo estimado: 1 día**

- Las notificaciones se pueden gestionar utilizando servicios como Firebase Cloud Messaging o WebSockets para enviar mensajes en tiempo real desde el backend. Estos servicios permiten que las notificaciones se envíen de manera eficiente a los usuarios, tanto en la web como en aplicaciones móviles, asegurando que los eventos importantes se comuniquen instantáneamente.

## 4. Won't Have:

- **Gestión de asesorías:**

- La gestión de asesorías se puede implementar utilizando frameworks backend como Node.js para manejar las solicitudes y programación de citas, con bases de datos como MySQL o PostgreSQL para almacenar información sobre clientes, asesores y horarios. Se pueden integrar APIs para la gestión de calendarios y recordatorios automáticos, así como notificaciones para confirmar citas o alertar sobre cambios.

- **Herramientas de promoción:**

- Tiempo estimado: 1 día**

- Las herramientas de promoción se pueden implementar utilizando plataformas de marketing por correo electrónico como Mailchimp o SendGrid para crear campañas, junto con frameworks backend como Node.js para gestionar el envío y seguimiento de promociones. También se pueden integrar APIs para la creación de cupones de descuento, promociones en redes sociales y análisis de rendimiento de campañas.





**Punto #4**

**DIAGRAMAS CASO DE USO:**

Diagrama de Caso de Uso: Cliente

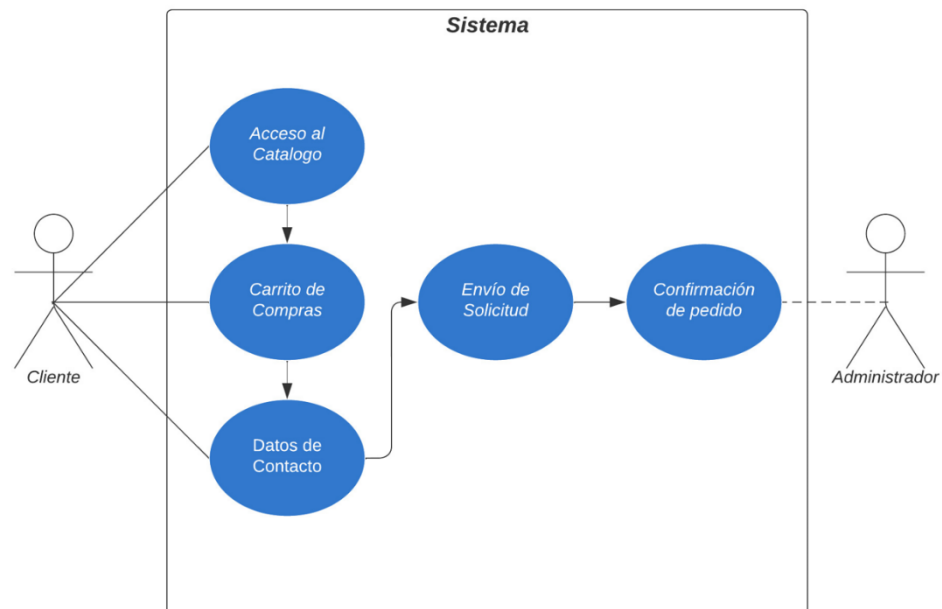
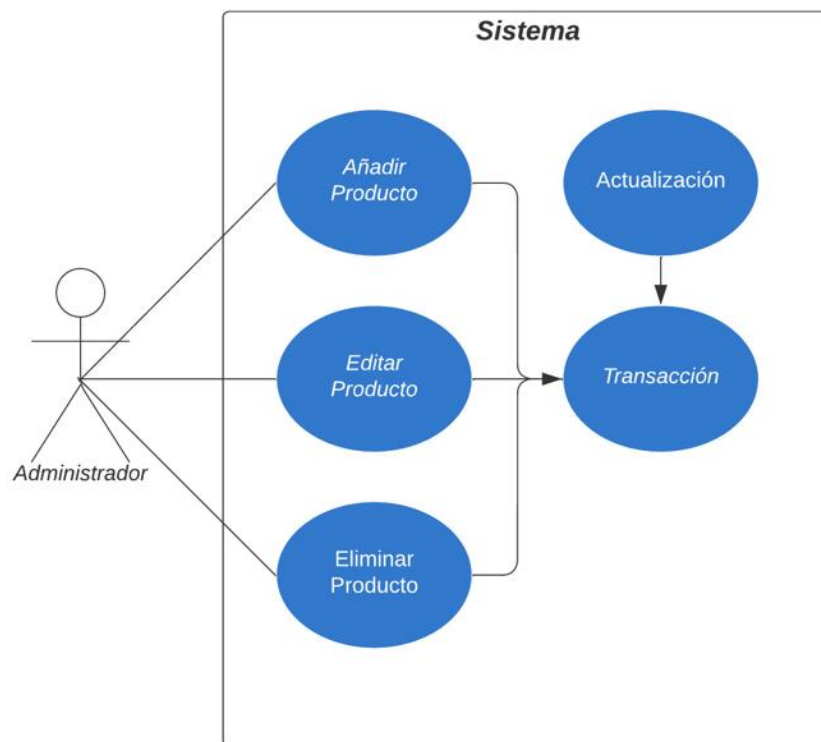


Diagrama de Caso de Uso: Administrador



<b>Caso de Uso del Administrador: Gestionar Inventario</b>	<b>Caso de Uso del Usuario: Realizar Pedido Desde el Catalogo</b>
<b>Breve Descripción</b> La administradora crea, edita o elimina productos en el inventario y visualiza el stock actualizado automáticamente al procesar pedidos. Recibe notificaciones cuando el stock es crítico.	<b>Breve Descripción</b> El cliente consulta el catálogo en línea, selecciona productos y envía la solicitud de pedido.
<b>Descripción Paso por Paso</b>  <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. La administradora accede a la sección "Inventario".</li> <li>2. Selecciona la opción de "Añadir Producto".</li> <li>3. Introduce los datos del producto (nombre, precio, stock, descripción, foto).</li> <li>4. Confirma y guarda los cambios.</li> <li>5. El sistema actualiza la lista de productos.</li> </ol> <b>Flujos alternativos:</b> <ol style="list-style-type: none"> <li>1. Editar Producto: La administradora selecciona un producto existente, edita los datos y guarda los cambios.</li> <li>2. Eliminar Producto: La administradora selecciona un producto y lo elimina del inventario.</li> </ol>	<b>Descripción Paso a Paso</b>  <b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El cliente accede al catálogo a través de un enlace.</li> <li>2. Navega por las categorías y selecciona un producto.</li> <li>3. Agrega el producto al carrito de compras.</li> <li>4. Ingresa sus datos de contacto (nombre, teléfono, dirección).</li> <li>5. Envía la solicitud de pedido.</li> <li>6. El sistema confirma el pedido y notifica al cliente.</li> </ol> <b>Flujos alternativos:</b> <ol style="list-style-type: none"> <li>1. Producto sin stock: El sistema informa que el producto no está disponible y sugiere otros productos relacionados.</li> </ol>



**Punto #5**

**Historia de usuario**



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá

Facultad de Ingeniería

Departamento de Sistemas e Industrial

Ingeniería de Software 1 (2016701)

**Anexo de Documentos Relacionados:**

- [HTTP response status codes – MDN Web Docs](#)

**Descripción conceptual**

<b>Módulo</b>	<b>Gestión de pedidos</b>
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	Visualizar pedidos pendientes en la tienda y mostrar información relevante para poder gestionarlos.

**Descripción técnica**

*La funcionalidad contará con dos endpoint para obtener una lista de pedidos pendientes y así mismo el historial de pedidos retornando datos relevantes como el identificador del pedido, productos, cantidades, precios, y estado. **Backend.***

*Aquí se debe explicar de manera técnica lo esperado,*

<b>URL</b>	<b>Método</b>	<b>Código html</b>
<b>https://user.domain.aws/orders/pending</b>	<b>GET</b>	<b>200</b> <b>500</b>
<b>Descripción</b>		
Devuelve una lista de todos los pedidos pendientes en la tienda.		

Datos de entrada	Datos de salida
<p>Este endpoint no necesita datos de entrada ya que solo va a devolver los pedidos.</p>	<pre>{   "status": "Succesful",   "statusCode": 200,   "data": [     {       "orderId": 101,       "customerName": "Juan Pérez",       "products": [         { "name": "Producto 1", "qty": 2, "price": 50 },         { "name": "Producto 2", "qty": 1, "price": 100 }       ],       "total": 200,       "date": "2024-12-20"     },     {       "orderId": 102,       "customerName": "María López",       "products": [         { "name": "Producto 3", "qty": 1, "price": 150 }       ],       "total": 150,       "date": "2024-12-19"     }   ] }  {   "status": "Error",   "statusCode": 500,   "message": "Error interno del servidor",   "data": {} }</pre>

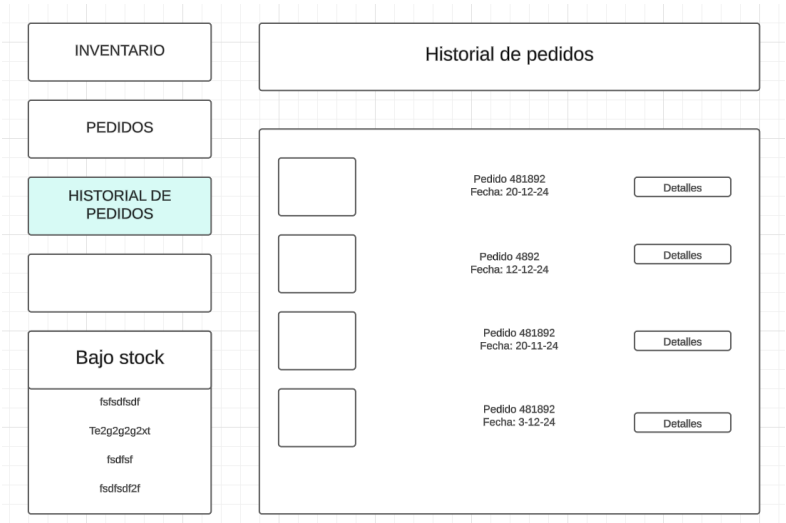
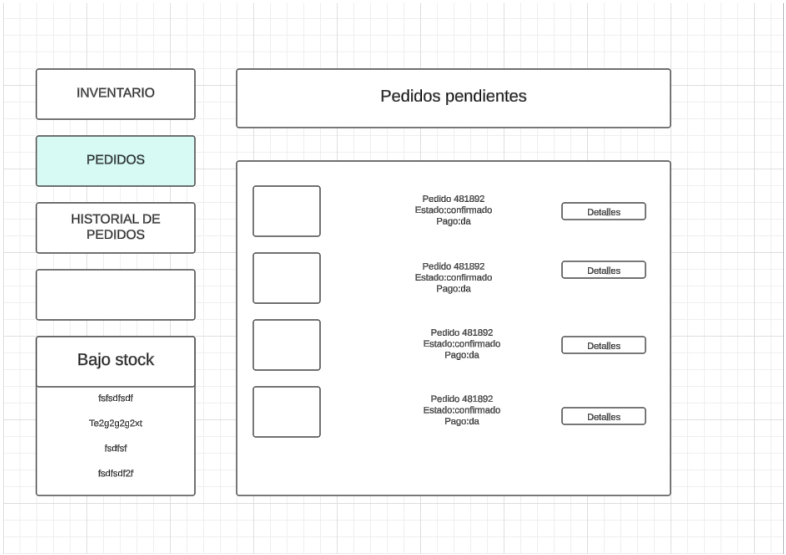
URL	Método	Código html
https://user.domain.aws/orders/history	GET	200 500
Descripción		
Devuelve una lista de todos los pedidos realizados en el pasado en la tienda.		
Datos de entrada	Datos de salida	
Este endpoint no necesita datos de entrada ya que solo va a devolver los pedidos.	<pre>{   "status": "Succesful",   "statusCode": 200,   "data": [     {       "orderId": 103,       "customerName": "Luis Gómez",       "total": 300,       "date": "2024-12-18",       "status": "Delivered"     },     {       "orderId": 104,       "customerName": "Ana Torres",       "total": 150,       "date": "2024-12-17",       "status": "Cancelled"     }   ] }</pre> <pre>{   "status": "Error",   "statusCode": 500,   "message": "Error interno del servidor",   "data": {} }</pre>	

La administradora podrá visualizar la información de los pedidos pendientes así mismo como el historial de todos los pedidos hechos.

Interacción esperada:

Al abrir el apartado de los pedidos deben aparecer en forma de cola de prioridad los pedidos que han sido realizados del mas reciente al más antiguo con su información relevante y cuando se le de click se muestre información relevante del mismo.

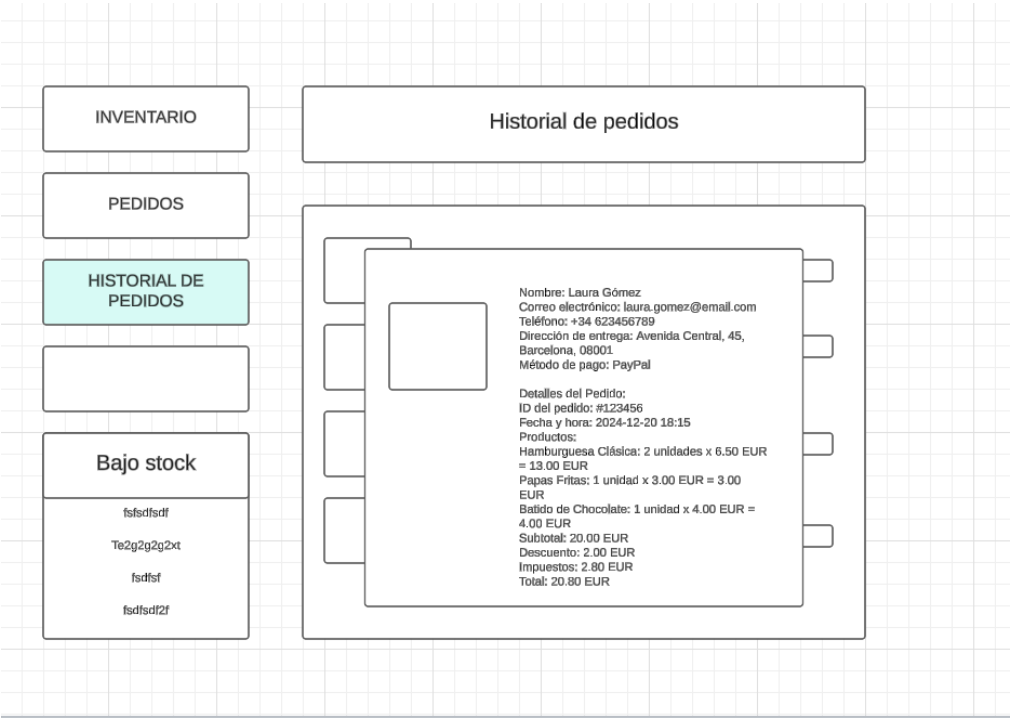
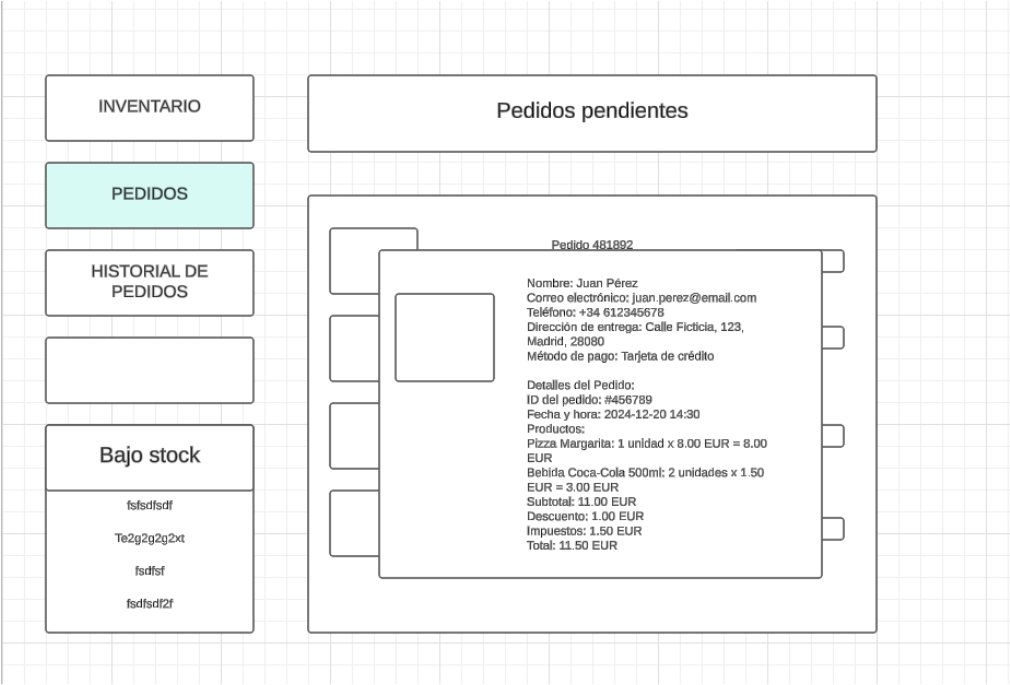
Mockups/Prototipos:





Flujo visual y eventos:

Al darle click a “Detalles” en el producto despliega una ventana con toda la información relevante del producto para ambos casos, historial de pedidos y pedidos pendientes.





## Historia de Usuario - Gestión de inventario

### Anexo de Documentos Relacionados:

- [\*HTTP response status codes\*](#)

### Descripción conceptual

Módulo	<i>Gestión de inventario</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Debe permitir al administrador registrar productos con detalles como nombre, precio, cantidad disponible, descripción y fotos.</i>  <i>Debe actualizar el inventario al completar un pedido y permitir la visualización de productos en una lista organizada.</i>

### Backend

#### Descripción técnica

##### 1. Registrar Producto:

- *Crear un endpoint que permita registrar nuevos productos en la base de datos.*
- *Validar los datos recibidos (nombre no vacío, precio positivo).*
- *Asociar una foto del producto al registro.*

##### 2. Actualizar Producto:

- *Endpoint para actualizar información de un producto existente (precio, stock, descripción, foto.).*
- *Actualizar el stock cuando se complete un pedido.*

##### 3. Consultar Inventario:

- *Endpoint que devuelva una lista de productos, incluyendo nombre, precio, stock actual, descripción y URL de la foto.*
- *Posibilidad de filtrar por nombre del producto.*

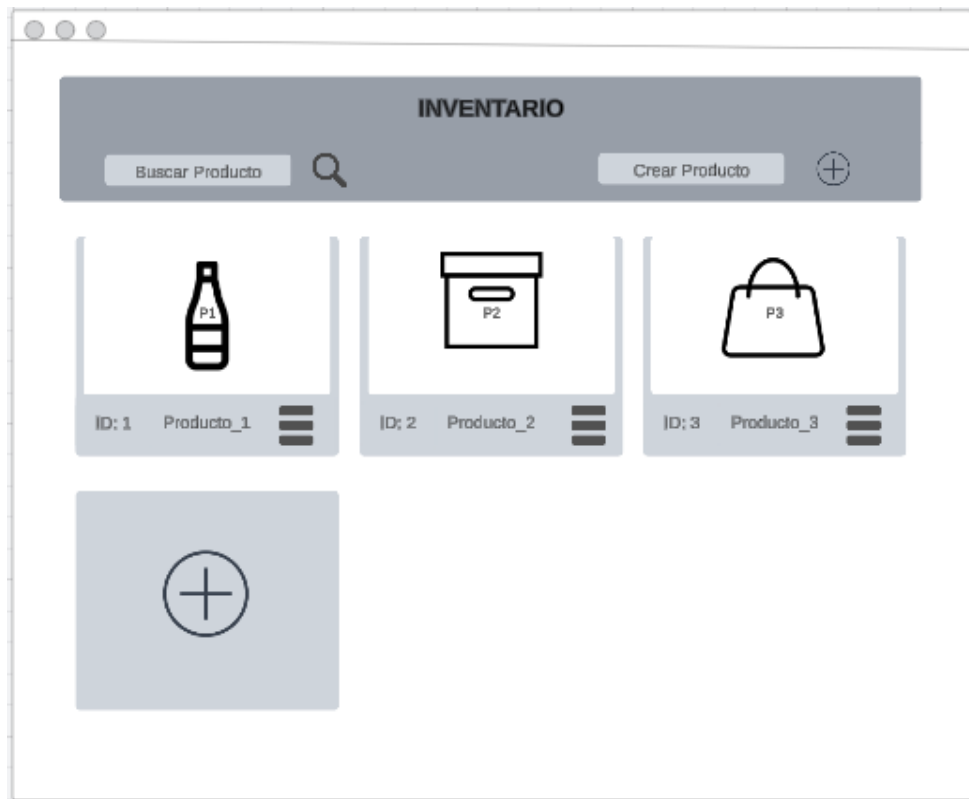
URL	Método	Código html
localhost:8080/api/inventor	POST	201
<b>Caso de uso técnico</b>  <b>Al registrar un producto, debe retornar código 201 y en el atributo data la información del producto creado.</b>		
<b>Datos de entrada</b> <pre>{   "nombre": "Producto 1",   "descripcion": "Descripción Producto 1",   "precio": 15000,   "stock": 25,   "url": "imagen_producto1.jpg" }</pre>		<b>Datos de salida</b> <pre>{   "status": "success",   "data": {     "id": 1,     "nombre": "Producto 1",     "descripcion": "Descripción Producto 1",     "precio": 15000,     "stock": 25,     "url": "imagen_producto1.jpg"   } }</pre>

URL	Método	Código html
localhost:8080/api/inventory/{id}	PUT	200
<b>Caso de uso técnico</b>  <b>Al actualizar un producto, debe retornar el código 200 y, en el atributo data, los datos del producto actualizado.</b>		
<b>Datos de entrada</b>  <pre>{   "descripcion": "nueva descripción producto",   "precio": 17000,   "stock": 30,   "foto": "imagen_actualizada.jpg" }</pre>		<b>Datos de salida</b>  <pre>{   "status": "success",   "data": {     "id": 1,     "nombre": "Producto 1",     "descripcion": "Nueva descripción producto",     "precio": 17000,     "stock": 30,     "foto": "imagen_actualizada.jpg"   } }</pre>

URL	Método	Código html
localhost:8080/api/inventor y	GET	200
<b>Caso de uso técnico</b>  <b>Al consultar, debe retornar código 200 y en el atributo data un array con los datos de los productos disponibles en el inventario.</b>		
<b>Datos de entrada</b>	<b>Datos de salida</b> <pre>{   "status": "success",   "data": [{     "id": 1,     "nombre": "Producto 1",     "descripcion": "Descripción Producto 1",     "precio": 15000,     "stock": 25,     "url": "imagen_producto1.jpg"   }, {     "id": 2,     "nombre": "Producto 2",     "descripcion": "Descripción Producto 2",     "precio": 10000,     "stock": 10,     "url": "imagen_producto2.jpg"   }] }</pre>	

## Frontend

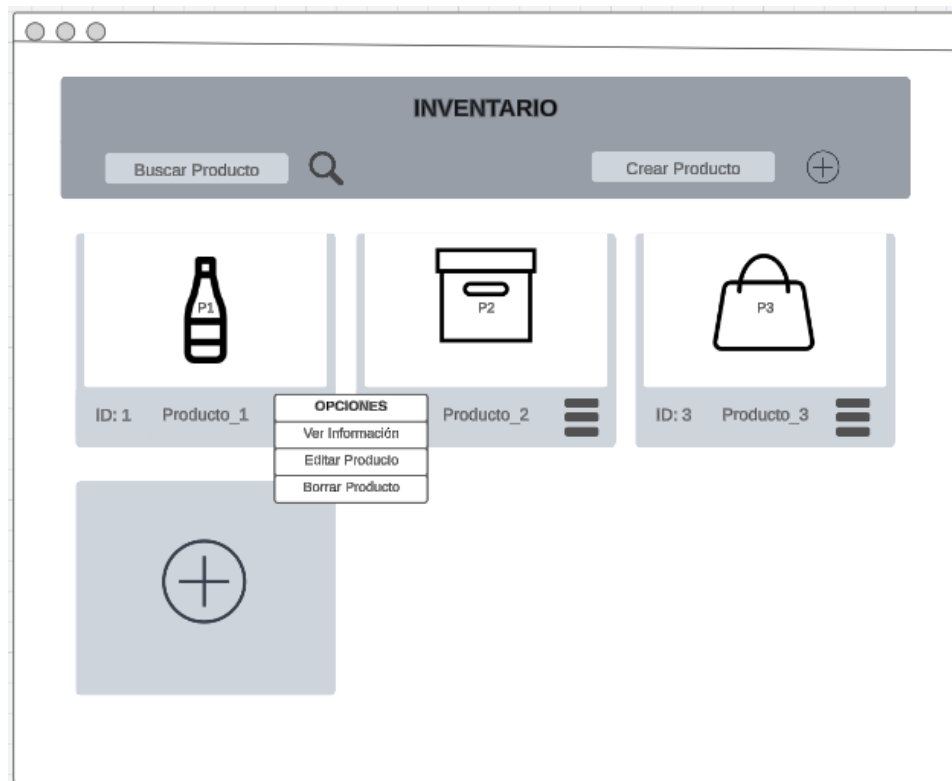
### Página Inventario



***Se muestran los productos en orden de su número de identificación.***

- ***Es posible buscar un elemento específico con el seleccionador “Buscar Producto”.***
- ***Al seleccionar el botón “Crear Producto” o al seleccionar la opción extra del inventario con el icono “+” pasará a la pantalla de creación de producto.***
- ***Al seleccionar el icono con tres líneas horizontales junto al nombre de alguno de los productos, se mostrará un menú desplegable con opciones del producto.***

## Página Inventario – Opciones desplegables

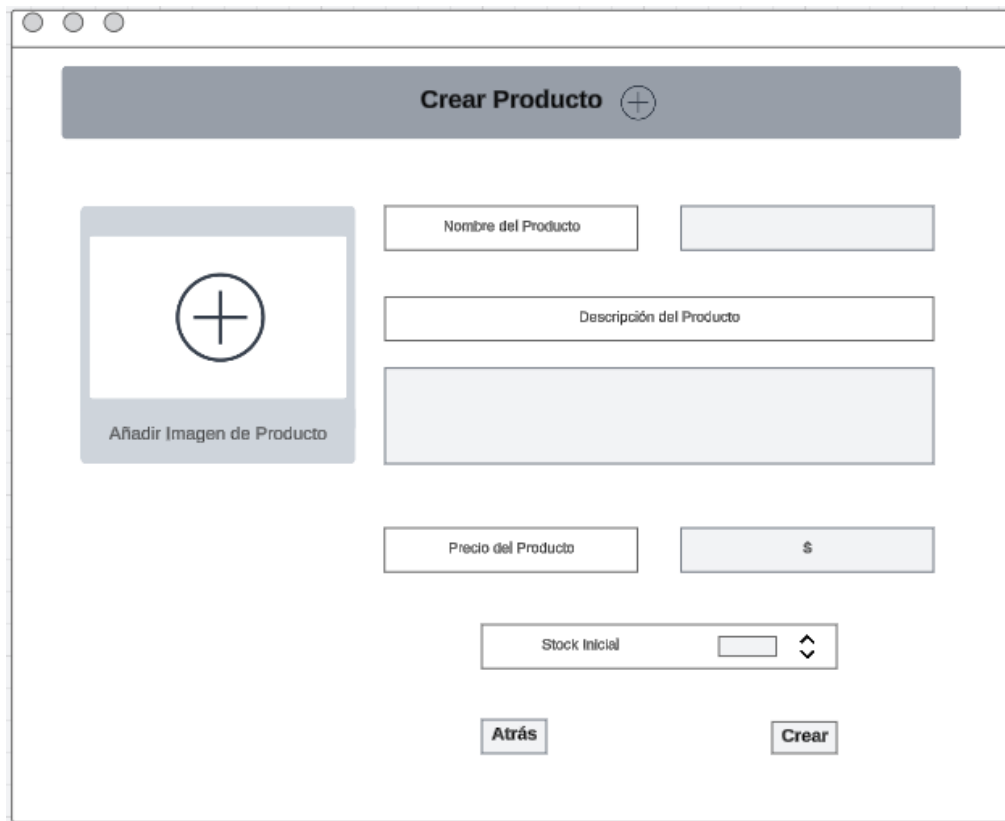


***Se muestra una lista desplegable con opciones por cada producto.***

- ***Al seleccionar “Ver Información” se abrirá la página “Información del producto”***
- ***Al seleccionar “Editar Producto” se abrirá la página “Editar Producto”***
- ***Al seleccionar “Borrar Producto” aparecerá una pregunta de confirmación y se borrará si el usuario la acepta.***



## Página Crear Producto



El formulario 'Crear Producto' está diseñado para ingresar datos de un nuevo producto. Incluye un encabezado con el título 'Crear Producto' y un icono de agregar (+). A la izquierda hay un área para subir una imagen con un icono de agregar y el texto 'Añadir Imagen de Producto'. A la derecha, los campos de entrada son: 'Nombre del Producto' (un solo renglón), 'Descripción del Producto' (un renglón largo), 'Precio del Producto' (un renglón con un símbolo de dólar (\$) a la derecha), y 'Stock Inicial' (un renglón con un selector de flechas hacia arriba y hacia abajo). En la parte inferior, hay dos botones: 'Atrás' y 'Crear'.

***En esta página se muestran opciones que deben ser llenadas a la hora de crear un nuevo producto y que este se guarde en el inventario.***

***Es obligatorio llenar los espacios:***

- ***Nombre del producto***
- ***Precio del Producto***
- ***Stock Inicial***

***También existe la opción de agregar una imagen para el producto, que se mostrará en el inventario y de descripción del producto que podrá ser visualizado en la página “Ver Producto”.***

- ***Para realizar la creación del producto se debe presionar el botón “Crear”.***
- ***Si se desea invalidar los datos y volver a la página de inventario se debe presionar el botón “Atrás”.***

## Página “Información del Producto”

The screenshot shows a web browser window with a title bar. The page has a header bar with the text "Información del Producto". Below the header, on the left, is a product image placeholder showing a bottle labeled "P1". Below the image is a table with two columns: "Identificador de Producto" and "1". To the right of the image, there are several input fields and buttons. The first row has "Nombre del Producto" and "Producto\_1". The second row has "Descripción del Producto" and "Descripción del producto 1". The third row has "Precio del Producto" and "\$ 15.000". The fourth row has "Stock Inicial" and "25". At the bottom right is a button labeled "Atrás".

***Se mostrará la información relacionada al producto elegido.***

- ***Hay que tener en cuenta que no se permite la edición de los valores del producto desde esta instancia.***
- ***El botón “Atrás” volverá a la página “Inventario”.***
- 

## Página “Editar Producto”

The screenshot shows a web browser window with a title bar. The page has a header bar with the text "Editar Producto" and a small icon of a wrench and screwdriver. Below the header, on the left, is a product image placeholder showing a plus sign inside a circle, with the text "Editar Imagen de Producto" below it. To the right of the image, there are several input fields and buttons. The first row has "Nombre del Producto" and "Producto\_1". The second row has "Descripción del Producto" and "Descripción del producto 1". The third row has "Precio del Producto" and "\$ 15.000". The fourth row has "Stock" and "25" with a dropdown arrow. At the bottom are two buttons: "Cancelar" and "Realizar Cambios".

***Se mostrará la información relacionada al producto elegido.***

- ***Desde esta página es posible cambiar los valores de:***
  - ***Nombre de producto***
  - ***Descripción del producto***
  - ***Precio del producto***
  - ***Stock***
  - ***Imagen del producto***
- ***Tenga en cuenta que el valor de identificación de producto no puede ser cambiado.***
- ***El botón “Realizar Cambios” confirma y actualiza la información nueva del producto y lo regresará a la página “Inventario”.***
- ***El botón “Atrás” lo regresará a la página “Inventario”.***





## Historia de Usuario - *Módulo de Gestión de Clientes*

### Anexo de Documentos Relacionados:

- [HTTP Bad Request \(400\) vs Not Found \(404\): Which status code to use?](#)
- [201 Created](#)
- [200 OK](#)

### Descripción conceptual

Módulo	<i>Módulo de Gestión de Clientes</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>La funcionalidad permite a la administradora registrar y almacenar información importante de los clientes, como su nombre, número de contacto y dirección. También incluye la posibilidad de modificar esta información en caso de errores o actualizaciones.</i>

### Descripción técnica

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

#### Backend

URL	Método	Código html
/clients	POST	201 (Creación exitosa), 400 (Error en la entrada)
<b>Descripción</b> Permite registrar un nuevo cliente.		
<b>Datos de entrada</b>		<b>Datos de salida</b>

<pre>{   "nombre": "Juan Pérez",   "numeroContacto": "+57 320 123 4567",   "direccion": "Calle 123, Bogotá" }</pre>		<pre>{   "status": "success",   "data": {     "id": 1,     "nombre": "Juan Pérez",     "numeroContacto": "+57 320 123 4567",     "direccion": "Calle 123, Bogotá"   } }</pre>
URL	Método	Código html
/clients/:id	PATCH	200 (Actualización exitosa), 404 (Cliente no encontrado)
<b>Caso de uso técnico</b> Permite modificar la información de un cliente existente.		
<b>Datos de entrada</b> <pre>{   "direccion": "Calle 456, Medellín" }</pre>		<b>Datos de salida</b> <pre>{   "status": "success",   "data": {     "id": 1,     "nombre": "Juan Pérez",     "numeroContacto": "+57 320 123 4567",     "direccion": "Calle 456, Medellín"   } }</pre>

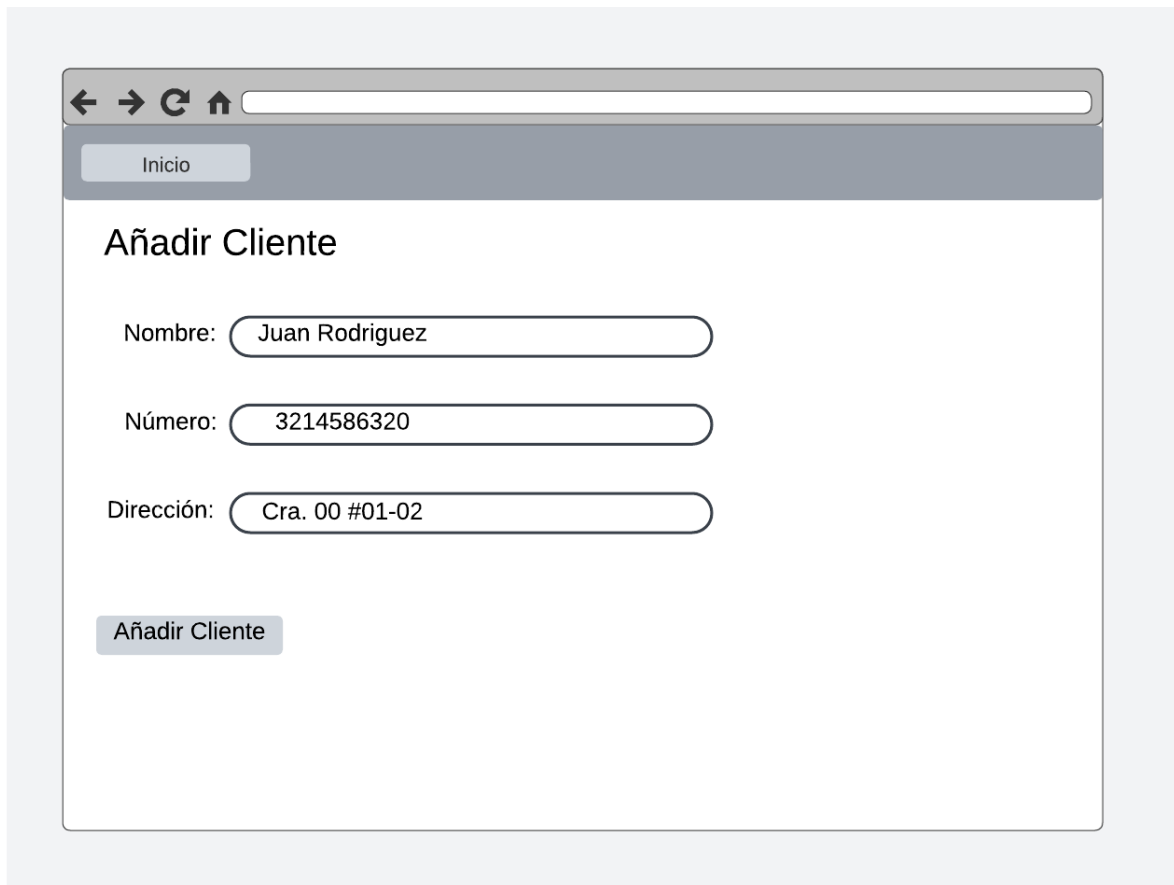
<b>URL</b>	<b>Método</b>	<b>Código html</b>
/clients	GET	200 (Actualización exitosa)
<b>Caso de uso técnico</b> Retorna la lista de todos los clientes registrados.		
<b>Datos de entrada</b>	<b>Datos de salida</b> <pre>{   "status": "success",   "data": [</pre>	

	<pre>{   "id": 1,   "nombre": "Juan Pérez",   "numeroContacto": "+57 320 123 4567",   "direccion": "Calle 123, Bogotá" }, {   "id": 2,   "nombre": "María Gómez",   "numeroContacto": "+57 310 987 6543",   "direccion": "Carrera 45, Medellín" } ]</pre>
--	---

[illegible]



**Formulario para registrar clientes nuevos, con su respectivo nombre, número y dirección.**



The image shows a web browser window with a navigation bar containing 'Inicio'. The main content area is titled 'Añadir Cliente' and contains three input fields: 'Nombre:' with the value 'Juan Rodriguez', 'Número:' with the value '3214586320', and 'Dirección:' with the value 'Cra. 00 #01-02'. Below these fields is a button labeled 'Añadir Cliente'.

***Flujo visual y eventos:***

**Al completar el formulario y enviarlo, se muestra un mensaje de éxito si la creación fue exitosa.**

**Al hacer clic en "Editar" en un cliente de la lista, se abre un modal con los datos prellenados para su edición, similar al formulario para añadir un cliente.**

**Al guardar los cambios, la tabla se actualiza dinámicamente.**

**Notas Adicionales:**

- **Validación de entrada para campos obligatorios (nombre y número de contacto).**
- **La tabla debe incluir una barra de búsqueda para facilitar la localización de clientes específicos.**





#### Anexo de Documentos Relacionados:

- [Twilio WhatsApp API](#)
- [SendGrid Email API](#)

#### Descripción conceptual

Módulo	Notificaciones
Descripción de la(s) funcionalidad(es) requerida(s):	Enviar notificaciones a la administradora cuando el stock de un producto es bajo y a los clientes sobre la confirmación de pedidos y su estado, priorizando WhatsApp como canal principal y utilizando Email como opción secundaria.

#### Descripción técnica

La funcionalidad contará con dos endpoints principales: uno para generar notificaciones de bajo stock y otro para gestionar notificaciones relacionadas con los pedidos.

Aquí se debe explicar de manera técnica lo esperado,

URL	Método	Código html
<a href="https://user.domain.aws/notifications/low-stock">https://user.domain.aws/notifications/low-stock</a>	POST	200 500
<b>Descripción</b> Envía una notificación a la administradora sobre el bajo stock de un producto.		
<b>Datos de entrada</b> { "productId": 12, "productName": "Producto X", "currentStock": 5, "threshold": 10 }		<b>Datos de salida</b> { "status": "Successful", "statusCode": 200, "message": "Notificación enviada exitosamente", "data": {} } { "status": "Error", "statusCode": 500, "message": "Error interno del servidor", "data": {} }

<b>URL</b> <b>https://user.domain.aws/n</b>	<b>Método</b>	<b>Código html</b> <b>200</b>
<b>otifications/order</b>	<b>POST</b>	<b>500</b>
<b>Descripción</b> Envía una notificación al cliente sobre la confirmación de un pedido o su actualización de estado.		
<b>Datos de entrada</b> { "orderId": 101, "customerId": 202, "customerName": "Juan Pérez", "contact": { "whatsapp": "+573001112223", "email": "juan.perez@email.com" }, "message": "Su pedido ha sido confirmado." }		<b>Datos de salida</b> { "status": "Succesful", "statusCode": 200, "message": "Notificación enviada exitosamente", "data": {} } { "status": "Error", "statusCode": 500, "message": "Error interno del servidor", "data": {} }

La administradora recibirá una notificación de bajo stock al identificarse que un producto ha alcanzado el límite establecido. Los clientes recibirán notificaciones automáticas por WhatsApp prioritariamente, y por Email como alternativa, sobre la confirmación o cambios en el estado de sus pedidos.

*Interacción esperada:*

*Al generarse una notificación, esta deberá enviarse al destinatario correspondiente por el canal prioritario configurado. En el caso de las notificaciones de bajo stock, el mensaje incluirá información sobre el producto afectado.*

Mockups/Prototipo

## Notificación de Bajo Stock

ID del Producto:

Nombre del Producto:

Stock Actual:

Umbral:

## Notificación de Pedido

ID del Pedido:

ID del Cliente:

Nombre del Cliente:

WhatsApp:

Email:

Mensaje:

*Flujo visual y eventos:*

**1. Notificación de Bajo Stock**

**Notificación Enviada:** El producto "Producto X" tiene un stock bajo (5 unidades). Se notificó a la administradora por **WhatsApp**.

**2. Notificación de Pedido**

**Notificación Enviada:** El cliente "Juan Pérez" fue notificado de la confirmación de su pedido a través de **WhatsApp**.

**3. Error en Notificación de Pedido**

**Error:** No se pudo enviar la notificación al cliente "María García" por **WhatsApp**. Se utilizó el método alternativo: **Email**.