

**TALLER 1**  
**INGENIERÍA DE SOFTWARE**

**PRESENTADO POR:**

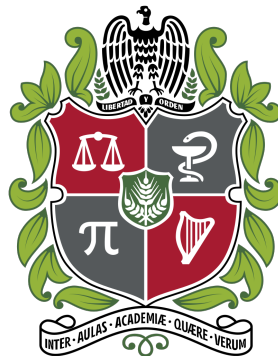
[Michael Stiven Betancourt Gelves](#)

[Santiago Bejarano Ariza](#)

[Johan Sebastian Roa Rodriguez](#)

**PROFESOR**

[Oscar Eduardo Alvarez Rodriguez](#)



**Universidad Nacional de Colombia**  
**Facultad de Ingeniería**  
**2024**

## Punto 1

Para el presente taller elegimos la conversación #2 para realizar un levantamiento de requerimientos, análisis de requerimientos, y desarrollo de casos de uso porque cumple con características clave que permiten abordar aspectos fundamentales del proceso de desarrollo de software, considerando su viabilidad, relevancia y potencial de impacto en el sistema final.

### 1. Claridad y especificidad

La historia de usuario seleccionada describe de manera clara y concreta una necesidad del usuario final. Esto facilita identificar los requerimientos iniciales sin ambigüedades, permitiendo definir rápidamente los objetivos del sistema y evitando malentendidos. Una historia de usuario bien definida es el punto de partida ideal para estructurar casos de uso, escenarios y diagramas necesarios en las fases de análisis y diseño.

### 2. Relevancia y aplicabilidad

La historia de usuario aborda una funcionalidad central y esencial del sistema. Al enfocarnos en un caso que representa un proceso crítico dentro del sistema, aseguramos que los esfuerzos invertidos en su desarrollo generen un impacto significativo tanto en la experiencia del usuario como en los objetivos del proyecto. Esto también justifica el análisis detallado, ya que cualquier error en esta parte podría afectar severamente el funcionamiento global.

### 3. Complejidad manejable

Esta historia de usuario tiene un nivel de complejidad que permite abordar todos los aspectos clave del ciclo de desarrollo de software:

- **Levantamiento de requerimientos:** Identificación de funcionalidades clave y dependencias externas.
- **Casos de uso:** Análisis de interacciones entre usuarios y el sistema, incluyendo flujos alternativos.
- **Diseño y desarrollo:** Planificación de arquitecturas y componentes necesarios.
- **Pruebas:** Identificación de criterios de aceptación claros basados en la historia.

Su alcance es lo suficientemente amplio como para explorar diversos aspectos del proyecto, pero sin ser excesivamente complejo, lo que permite concentrarse en las mejores prácticas.

### 4. Representación de escenarios reales

La historia de usuario refleja una necesidad realista y común, lo que asegura que los hallazgos y resultados del análisis serán aplicables a contextos reales. Esto no solo aumenta la utilidad del análisis, sino que también mejora las habilidades del equipo al abordar problemas prácticos.

## **5. Posibilidad de iteración**

Elegir una historia de usuario fundamental permite que los procesos de desarrollo e iteración se construyan sobre una base sólida. La funcionalidad descrita puede ser fácilmente expandida o conectada con otras partes del sistema en futuras fases del proyecto, haciendo que este análisis inicial tenga un valor estratégico.

## **6. Evaluación de la triada de desarrollo**

La historia elegida facilita evaluar cómo afecta la triada de tiempo, costo y alcance. Al ser una funcionalidad clave, permite analizar detalladamente:

- Los recursos necesarios.
- El tiempo requerido para implementar la funcionalidad.
- Las restricciones presupuestarias y cómo estas influyen en el desarrollo.

## **7. Pruebas de tecnologías y procesos**

La implementación de esta historia de usuario permitirá validar tecnologías y metodologías que se planean usar en el proyecto. Esto incluye la integración de herramientas como frameworks, APIs externas o infraestructuras necesarias, identificando posibles limitaciones o ajustes desde las primeras etapas.

## **8. Impacto directo en el valor del sistema**

La funcionalidad descrita en la historia de usuario seleccionada tiene un impacto directo en la satisfacción del usuario final, dado que aborda una necesidad específica que, al ser cubierta, incrementa la percepción de valor del sistema. Elegir una funcionalidad con un impacto tan visible asegura que el tiempo invertido en análisis y diseño sea bien aprovechado.

## Punto 2

### Requerimientos funcionales:

#### 1. **Gestor de Pedidos**

- Crear y gestionar pedidos como tareas que incluyan los datos del cliente, productos solicitados, estado del pedido y fecha de entrega.
- Actualizar el estado de los pedidos (pendiente, pagado, enviado, completado, cancelado).
- Notificar automáticamente al cliente sobre el registro de su pedido y cambios de estado a través de WhatsApp.
- Hacer que de alguna forma los clientes puedan consultar los detalles de su pedido.

#### 2. **Control de Inventario**

- Actualizar automáticamente el inventario al completar un pedido.
- Notificar a la dueña por WhatsApp cuando el stock de un producto esté por agotarse.
- Permitir la modificación manual del inventario para agregar o retirar productos.
- Visualizar en tiempo real los productos disponibles, sus cantidades y descripciones.

#### 3. **Catálogo en Línea**

- Mostrar un catálogo organizado con productos divididos en categorías como por ejemplo: artesanías, accesorios, decoración.
- Integrar descripciones, precios, y fotos de los productos.
- Anunciar más algunos productos en promoción.

#### 4. **Gestión de Clientes**

- Registrar automáticamente información de clientes al realizar un pedido.
- Permitir la modificación manual de datos del cliente.
- Almacenar el historial de pedidos por cliente para consultas futuras.

#### 5. **Panel de Administración**

- Hacer un resumen visual en el inicio de la aplicación con:
  - Pedidos pendientes.
  - Productos que se estén agotando
  - Historial de ventas.
- Que se pueda acceder y editar de información de pedidos, productos y clientes.

#### 6. **Gestor de Finanzas**

- Registrar ingresos automáticamente al completar ventas.
- Registrar manualmente gastos relacionados con materiales y productos.
- Generar reportes de ganancia neta.

#### 7. **Promociones y Marketing**

- Integrar el catálogo con redes sociales para aumentar la visibilidad del negocio.
- Permitir envío de promociones o descuentos a los clientes por WhatsApp.

#### 8. **Compatibilidad con varios dispositivos**

- Diseñar el sistema como una página web responsive que se pueda acceder desde celulares y pc.

- Tener una interfaz que sea intuitiva y amigable para usuarios con poco manejo de los dispositivos electrónicos.

#### **Requerimientos No Funcionales:**

##### **1. Acceso y Seguridad**

- Sistema protegido con autenticación de usuario.
- Administrador con acceso exclusivo a la información sensible.

##### **2. Almacenamiento de Datos**

- Base de datos para registrar productos, clientes, pedidos y finanzas.

##### **3. Notificaciones**

- Integración con WhatsApp para notificaciones automatizadas.

##### **4. Escalabilidad**

- Posibilidad de agregar nuevas funcionalidades, como un calendario para reservas de asesorías, sin necesidad de tener que reconstruir el sistema.

##### **5. Rendimiento**

- Sistema optimizado para tiempos de carga rápidos tanto en celular como en pc.

##### **6. Usabilidad**

- Tener un diseño intuitivo que facilite la navegación para usuarios con poca experiencia técnica.



### Punto 3

## Clasificación Completa de Funcionalidades con MoSCoW

### Must Have (Obligatorias)

#### 1. Gestor de Pedidos

- Crear y gestionar pedidos.
- Actualizar estados de pedidos.
- Notificar al cliente sobre el registro y cambios vía WhatsApp.
- Consulta de detalles de pedidos.
- **Tiempo estimado:** 10 días.

#### 2. Control de Inventario

- Actualización automática al completar un pedido.
- Notificaciones de bajo stock por WhatsApp.
- Modificación manual del inventario.
- Visualización en tiempo real de productos y cantidades.
- **Tiempo estimado:** 12 días.

#### 3. Gestión de Clientes

- Registro automático al realizar un pedido.
- Modificación manual de datos.
- Historial de pedidos por cliente.
- **Tiempo estimado:** 8 días.

#### 4. Panel de Administración

- Resumen visual: pedidos pendientes, productos agotándose, historial de ventas.
- Acceso y edición de información (pedidos, productos y clientes).
- **Tiempo estimado:** 8 días.

#### 5. Acceso y Seguridad

- Autenticación de usuario.
- Acceso exclusivo del administrador a datos sensibles.
- **Tiempo estimado:** 5 días.

#### 6. Compatibilidad con Varios Dispositivos

- Página web responsive para PC y móviles.
- Interfaz intuitiva y amigable para usuarios con poca experiencia técnica.
- **Tiempo estimado:** 7 días.

---

### Should Have (Recomendadas)

#### 7. Catálogo en Línea

- Mostrar productos organizados por categorías con fotos, descripciones y precios.
- Anunciar productos en promoción.
- **Tiempo estimado:** 10 días.

## 8. Gestor de Finanzas

- Registro automático de ingresos.
- Registro manual de gastos.
- Reportes de ganancia neta.
- **Tiempo estimado:** 10 días.

## 9. Promociones y Marketing

- Integración con redes sociales.
- Envío de promociones o descuentos vía WhatsApp.
- **Tiempo estimado:** 7 días.

## 10. Notificaciones Automatizadas

- Integración completa con WhatsApp para notificaciones automatizadas.
  - **Tiempo estimado:** 5 días.
- 

## Could Have (Opcionales)

### 11. Escalabilidad

- Posibilidad de agregar nuevas funcionalidades sin reestructurar el sistema.
- **Tiempo estimado:** 5 días.

### 12. Rendimiento

- Optimización básica para tiempos de carga rápidos en PC y móviles.
  - **Tiempo estimado:** 5 días.
- 

## Won't Have (Excluidas por ahora)

### 13. Almacenamiento de Datos

- Diseño avanzado de bases de datos para crecimiento a largo plazo.
- **Razonamiento:** Una base simple como SQLite es suficiente para un proyecto universitario.

### 14. Extras No Funcionales

- Soporte para funcionalidades adicionales complejas no requeridas en este alcance inicial.
- 

## Estimación Total

**Tiempo total: 72 días (aproximadamente 2 meses y medio).**

Las funcionalidades prioritarias (**Must Have** y **Should Have**) pueden desarrollarse en paralelo para cumplir el tiempo límite del proyecto.

## Distribución por semanas:

### 1. Semana 1-2:

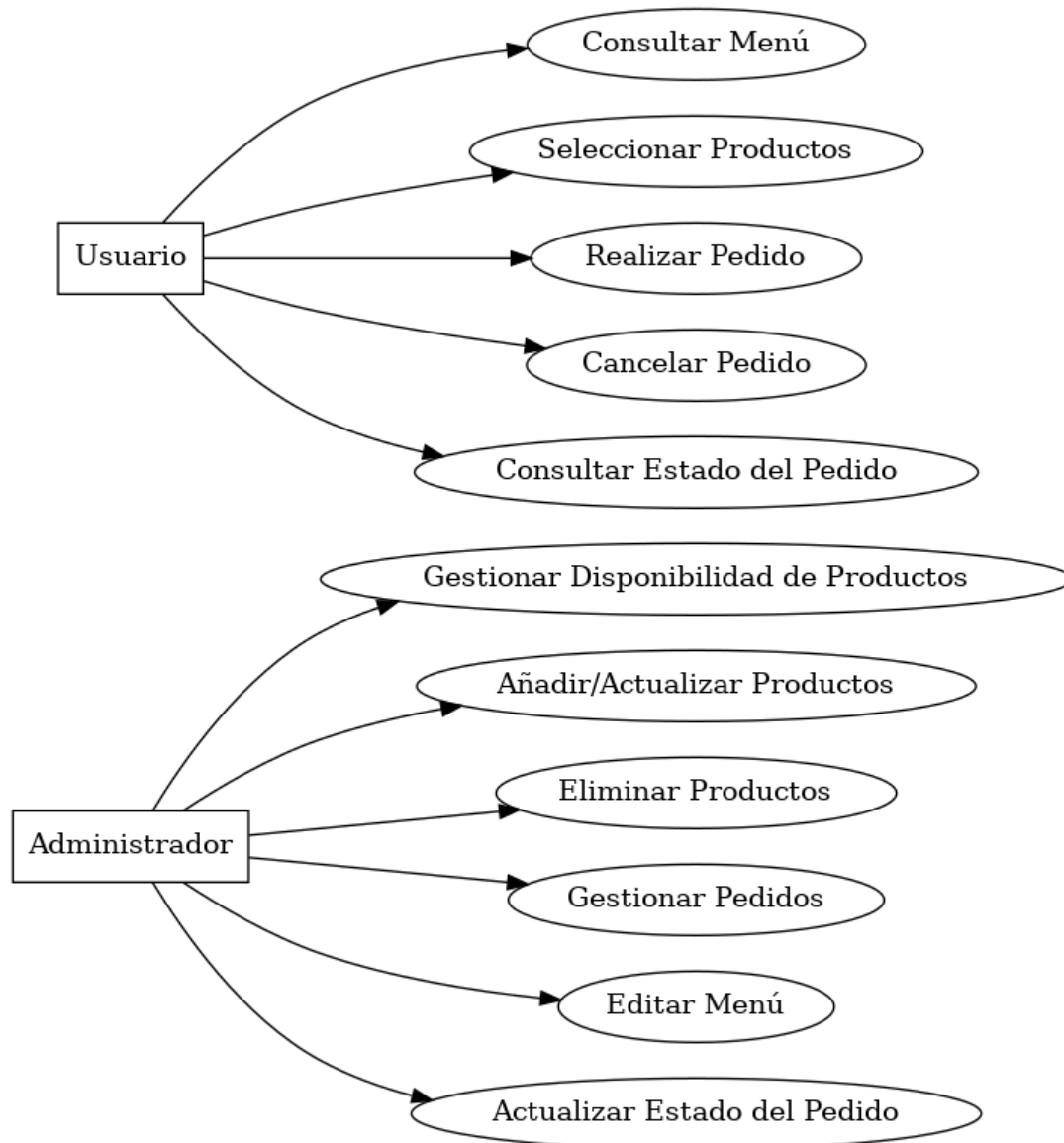
- Gestor de Pedidos.



- Control de Inventario.
- 2. **Semana 3:**
  - Gestión de Clientes.
  - Panel de Administración.
- 3. **Semana 4:**
  - Catálogo en Línea.
  - Promociones y Marketing.
- 4. **Semana 5:**
  - Acceso y Seguridad.
  - Compatibilidad con Varios Dispositivos.
- 5. **Semana 6:**
  - Gestor de Finanzas.
  - Notificaciones Automatizadas.
- 6. **Semana 7:**
  - Escalabilidad y Rendimiento.
  - Pruebas y ajustes finales.



#### Punto 4



#### Explicación de los casos de uso:

##### Para el administrador:

1. **Gestionar Disponibilidad de Productos:** Permite activar o desactivar productos según su disponibilidad.
2. **Añadir/Actualizar Productos:** Incluye la creación de nuevos productos o la edición de los existentes.
3. **Eliminar Productos:** Facilita la eliminación de productos del sistema.

4. **Editar Menú:** Implica la organización y categorización de los productos en el menú.
5. **Gestionar Pedidos:** Permite al administrador procesar y gestionar los pedidos realizados.
6. **Actualizar Estado del Pedido:** Cambia el estado de un pedido (por ejemplo, de "en preparación" a "listo para entrega").

**Para el usuario:**

1. **Consultar Menú:** Los usuarios pueden ver los productos disponibles en el menú.
2. **Seleccionar Productos:** Permite elegir productos específicos para el pedido.
3. **Realizar Pedido:** Completa el pedido seleccionando los productos deseados.
4. **Cancelar Pedido:** Da la opción de cancelar un pedido antes de ser procesado.
5. **Consultar Estado del Pedido:** Permite a los usuarios verificar el estado de su pedido en tiempo real.



## Punto 5 HU1

### Historia de Usuario #1

#### Anexo de Documentos Relacionados:

- Documento de requerimientos funcionales

#### Descripción conceptual

Módulo	Gestor de Pedidos
Descripción de la funcionalidad requerida:	El sistema debe permitir crear y gestionar pedidos, incluyendo datos del cliente, productos solicitados, estado del pedido y fecha de entrega. Además, debe actualizar los estados de los pedidos (pendiente, pagado, enviado, completado, cancelado) y notificar automáticamente al cliente sobre el registro y los cambios de estado del pedido a través de WhatsApp. Los clientes deben poder consultar los detalles de sus pedidos.

#### Descripción técnica

##### Backend

URL	Método	Código html
localhost:8080/api/orders	POST	201 400

Descripción	
Crear un nuevo pedido con los datos del cliente, productos y estado inicial (pendiente).	
<div>Datos de entrada</div> <div>201:</div> <div>{</div> <div>  "cliente": {</div> <div>    "nombre": "Juan Pérez",</div> <div>    "telefono": "+573001234567"</div> <div>  },</div> <div>  "productos": [</div> <div>    { "id": 1, "cantidad": 2 },</div> <div>    { "id": 2, "cantidad": 1 }</div> <div>  ],</div> <div>  "fecha_entrega": "2024-12-25"</div> <div>}</div>	<div>Datos de salida</div> <div>201:</div> <div>{</div> <div>  "cliente": {</div> <div>    "nombre": "Juan Pérez",</div> <div>    "telefono": "+573001234567"</div> <div>  },</div> <div>  "productos": [</div> <div>    { "id": 1, "cantidad": 2 },</div> <div>    { "id": 2, "cantidad": 1 }</div> <div>  ],</div> <div>  "fecha_entrega": "2024-12-25"</div> <div>}</div>

```
400:
{
  "cliente": {
    "nombre": "Juan Pérez",
    "telefono": "+573001234567"
  },
  "productos": [
    { "id": 1, "cantidad": 2 },
    { "id": 2, "cantidad": 1 }
  ],
  "fecha_entrega": "2024-12-25"
}
```

```
400:
{
  "status": "unprocessed",
  "message": "No existe el parámetro  
<nomxbre>"
}
```

## Frontend

El frontend estará diseñado con un enfoque responsivo, asegurando que la aplicación funcione en dispositivos móviles y de escritorio. Los elementos principales incluyen:

### Interacción esperada:

- El usuario interactúa con un formulario para ingresar los datos del cliente, seleccionar los productos y establecer la fecha de entrega.
- Un botón de "Crear Pedido" permite enviar la información.
- Una vez creado, el sistema redirige al usuario a la vista del pedido con su información detallada.

### Mockups/Prototipos:

- Formulario de creación de pedido con campos para:
  - Datos del cliente.
  - Selector de productos con cantidad.
  - Calendario para la fecha de entrega.
- Tabla con la lista de pedidos mostrando:



- ID del pedido, cliente, estado, fecha de entrega y acción para editar/actualizar estado.



### Flujo visual y eventos:

1. El usuario llena los campos del formulario y presiona "Crear Pedido".
2. Se muestra un spinner mientras se procesa la solicitud.
3. Una notificación confirma que el pedido fue creado exitosamente.
4. La información del pedido se agrega a la tabla de pedidos.

(Mockup generado con IA)



## Punto 5 HU 2

### Descripción conceptual

<b>Módulo</b>	<i>Catálogo de Productos</i>
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	<i>El sistema debe permitir a los clientes que puedan ver un catálogo en línea con los productos disponibles, organizados en categorías. Los productos deben mostrar información relevante como: nombre, descripción, precio, disponibilidad y una imagen acorde al producto. Además, el catálogo debe ser responsive para una experiencia fluida en celulares y pc.</i>

### Backend

<b>URL</b> localhost:8080/productos	<b>Método</b> GET	<b>Código html</b> 200
<b>Descripción</b> Al consultar, retorna un código 200 y una lista de productos categorizados.		
<b>Datos de entrada</b>	<b>Datos de salida</b> <pre>{   "status": "success",   "data": [     {       "id": 1,       "nombre": "Pulsera Artesanal",       "descripcion": "Pulsera hecha a mano con materiales naturales.",       "precio": 15000,       "stock": 10,       "categoria": "Artesanías",       "imagen": "url_imagen_pulsera.jpg"     },     ...   ] }</pre>	

<b>URL</b> localhost:8080/productos/{id}	<b>Método</b> GET	<b>Código html</b> 200
---	----------------------	---------------------------

<b>Descripción</b> Devuelve los detalles de un producto específico.	
<b>Datos de entrada</b> <pre>{   "id": 1 }</pre>	<b>Datos de salida</b> <pre>{   "status": "success",   "data": {     "id": 1,     "nombre": "Pulsera Artesanal",     "descripcion": "Pulsera hecha a mano con materiales naturales.",     "precio": 15000,     "stock": 10,     "categoria": "Artesanías",     "imagen": "url_imagen_pulsera.jpg"   } }</pre>

### Frontend

Los usuarios deben poder navegar por el catálogo mediante categorías visibles en un menú, además cada producto se muestra en un formato de tarjeta con imagen, nombre, descripción, precio y disponibilidad, sumado a esto al hacer clic en un producto, el usuario accede a una vista detallada con información completa del mismo.



(Mockup realizado con ayuda de IA)



### Punto 5 HU 3

#### Descripción conceptual

<b>Módulo</b>	<i>Módulo de Gestión de Inventario</i>
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	<i>El sistema debe permitir actualizar automáticamente el inventario al completar un pedido, notificar a la dueña por WhatsApp cuando el stock de un producto esté por agotarse, permitir la modificación manual del inventario para agregar o retirar productos y visualizar en tiempo real los productos disponibles, sus cantidades y descripciones.</i>

#### Descripción técnica

##### Backend

URL	Método	Código HTTP
Localhost:8080/api/inventory/update	PATCH	<b>200</b>
Descripción		
Actualiza el inventario al completar un pedido.		
Datos de entrada	Datos de salida	
{ "order_id": 12345 }	{ "status": "success", "updated_products": [ {"product_id": 1, "stock": 8}, {"product_id": 2, "stock": 15} ] }	

URL	Método	Código HTTP
Localhost:8080/api/inventory/low-stock	GET	200
Descripción Devuelve los productos con bajo inventario y envía notificaciones.		
Datos de entrada	Datos de salida <pre>{   "status": "success",   "low_stock_products": [     {"product_id": 3, "name":       "Producto X", "stock": 2}   ] }</pre>	

URL	Método	Código HTTP
Localhost:8080/api/inventory/manual	POST	200
Descripción Permite al administrador modificar manualmente el inventario.		
Datos de entrada <pre>{   "product_id": 5,   "adjustment": -3 }</pre>	Datos de salida <pre>{   "status":     "success",   "product_id": 5,   "new_stock": 7 }</pre>	

**Frontend**



El frontend tendrá un diseño ajustable al tamaño de la pantalla y con opción para seleccionar cualquiera de las opciones solicitadas por el administrador para hacer el control del inventario.

*Interacción esperada:*

- *Dashboard del administrador:*
  - *Se muestra una tabla con los productos disponibles, incluyendo columnas para nombre, cantidad en stock y descripción.*
  - *Botones de acción para editar manualmente el inventario (agregar o restar unidades).*
  - *Indicador visual (alerta roja) para productos con bajo stock.*
- *Vista del usuario final:*
  - *Una lista de productos con nombre, descripción y disponibilidad.*
  - *Productos no disponibles se muestran con un mensaje de "Agotado" y deshabilitados para seleccionarlos.*

*Flujo visual y eventos:*

- *Dashboard:*
  - *Al completar un pedido, el backend actualiza automáticamente el inventario y muestra los cambios en tiempo real.*
  - *Si un producto tiene menos de 5 unidades, se envía una notificación por WhatsApp y aparece una alerta visual.*
  - *Al hacer clic en el botón "Editar", se abre un modal para ajustar manualmente el inventario.*
- *Vista de usuario:*
  - *La interfaz carga productos disponibles al inicio.*
  - *Si un producto está agotado, el botón de compra aparece deshabilitado con un mensaje explicativo.*



(Mockup realizado con ayuda de IA)