



Universidad Nacional de Colombia - sede Bogotá  
Facultad de Ingeniería  
Departamento de Sistemas e Industrial  
Curso: Ingeniería de Software 1 (2016701)

### Taller del módulo de requerimientos

Alejandro Alejandro Arguello Munoz

Steven David Alfonso Galindo

Daniel Santiago Delgado Pinilla

Juan Luis Vergara Novoa

Antes que nada queremos recalcar que para la elaboración del presente taller hemos usado ayuda de modelos de inteligencia artificial pertenecientes a OpenAI.

El presente archivo contiene las soluciones de los numerales 1-5. Los puntos del proyecto aún continúan en discusión, sin embargo presentamos un esquema inicial de los posibles requerimientos del mismo.

En base a las conversaciones obtenidas en clase la clase práctica de levantamiento de requerimientos, realizar lo siguiente:

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 cuento ofrecía la cliente para el desarrollo 😊**

GP 1: [CONVERSACIÓN](#)

GP 2: [CONVERSACIÓN](#)

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.**

## Requerimientos Funcionales Conversación grupo 2

A partir de la entrevista, se han identificado las siguientes necesidades funcionales, organizadas por prioridad:

### 1. Organización de Pedidos:

- Registro detallado de cada pedido, incluyendo información del cliente, productos solicitados, estado de pago y entrega.
- Capacidad para modificar y actualizar el estado de los pedidos en tiempo real.

### 2. Control de Inventario:

- Gestión automática del inventario que se actualiza conforme se realizan ventas.
- Alertas automáticas cuando el stock de un producto alcanza niveles críticos.

### 3. Gestión de Ventas y Pagos:

- Registro automático de transacciones de pago.
- Integración con pasarelas de pago locales como Nequi y Daviplata.
- Generación de comprobantes de pago.

### 4. Catálogo en Línea:

- Plataforma accesible donde los clientes puedan visualizar productos, precios y disponibilidad.
- Funcionalidad para que los clientes realicen pedidos directamente desde el catálogo.

### 5. Promoción de Productos:

- Herramientas para integrar el catálogo con redes sociales.
- Opciones para enviar promociones y actualizaciones a clientes potenciales.

### 6. Agenda para Asesorías (Prioridad Baja):

- Calendario integrado para que los clientes puedan agendar citas para servicios de asesoría.

### 7. Notificaciones Automáticas:

- Alertas para el administrador sobre el estado de pedidos e inventario.
- Notificaciones para clientes sobre el estado de sus pedidos.

### 8. Gestión de Datos y Seguridad:

- Almacenamiento seguro de información de productos, clientes y pedidos.
- Roles de acceso diferenciados y respaldo automático de datos.

### 9. Interfaz de Usuario Personalizada:

- Diseño visual atractivo y funcional acorde a la identidad del negocio.
- Evitar elementos molestos como pop-ups invasivos y procesos de compra complicados.

### 10. Visualización y Panel de Control Administrativo:

- Dashboard con resúmenes de pedidos pendientes, productos con bajo stock y últimos registros.

- Navegación clara para acceder a diferentes secciones del sistema.

### **Requerimientos No Funcionales**

#### **1. Accesibilidad y Usabilidad:**

- Plataforma accesible desde dispositivos móviles y computadoras.
- Interfaz intuitiva y fácil de usar, adaptada a usuarios con poca experiencia tecnológica.

#### **2. Disponibilidad y Rendimiento:**

- Funcionamiento predominantemente online, con opciones básicas disponibles offline.
- Tiempo de carga optimizado para garantizar una experiencia fluida, especialmente en dispositivos móviles.

#### **3. Seguridad:**

- Protección de datos sensibles de clientes y transacciones.
- Acceso restringido a la información administrativa.

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

- Capacidad para agregar funcionalidades adicionales en el futuro sin necesidad de una reestructuración completa del sistema.

#### **5. Personalización y Diseño:**

- Diseño visual atractivo que refleje la identidad del negocio, utilizando una paleta de colores cálidos y naturales.
- Evitar elementos que puedan resultar molestos para los usuarios, como pop-ups invasivos o procesos de compra complicados.



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)

## Must Have

1. **Organización de Pedidos**
  - Registro detallado de cada pedido, incluyendo información del cliente, productos solicitados, estado de pago y entrega.
  - Capacidad para modificar y actualizar el estado de los pedidos en tiempo real.
2. **Control de Inventario**
  - Gestión automática del inventario que se actualiza conforme se realizan ventas.
  - Alertas automáticas cuando el stock de un producto alcanza niveles críticos.
3. **Gestión de Ventas y Pagos**
  - Registro automático de transacciones de pago.
  - Integración con pasarelas de pago locales como Nequi y Daviplata.
  - Generación de comprobantes de pago.
4. **Catálogo en Línea**
  - Plataforma accesible donde los clientes puedan visualizar productos, precios y disponibilidad.
  - Funcionalidad para que los clientes realicen pedidos directamente desde el catálogo.
5. **Notificaciones Automáticas**
  - Alertas para el administrador sobre el estado de pedidos e inventario.
  - Notificaciones para clientes sobre el estado de sus pedidos.
6. **Accesibilidad y Usabilidad (Requerimiento No Funcional)**
  - Plataforma accesible desde dispositivos móviles y computadoras.
  - Interfaz intuitiva y fácil de usar, adaptada a usuarios con poca experiencia tecnológica.
7. **Seguridad (Requerimiento No Funcional)**
  - Protección de datos sensibles de clientes y transacciones.
  - Acceso restringido a la información administrativa.

## Should Have

1. **Promoción de Productos**
  - Herramientas para integrar el catálogo con redes sociales.
  - Opciones para enviar promociones y actualizaciones a clientes potenciales.

2. **Gestión de Datos y Seguridad** (Requerimiento No Funcional)
  - Almacenamiento seguro de información de productos, clientes y pedidos.
  - Roles de acceso diferenciados y respaldo automático de datos.
3. **Interfaz de Usuario Personalizada** (Requerimiento No Funcional)
  - Diseño visual atractivo y funcional acorde a la identidad del negocio.
  - Evitar elementos molestos como pop-ups invasivos y procesos de compra complicados.
4. **Visualización y Panel de Control Administrativo**
  - Dashboard con resúmenes de pedidos pendientes, productos con bajo stock y últimos registros.
  - Navegación clara para acceder a diferentes secciones del sistema.

### Could Have

1. **Agenda para Asesorías (Prioridad Baja)**
  - Calendario integrado para que los clientes puedan agendar citas para servicios de asesoría.
2. **Disponibilidad y Rendimiento** (Requerimiento No Funcional)
  - Funcionamiento predominantemente online, con opciones básicas disponibles offline.
  - Tiempo de carga optimizado para garantizar una experiencia fluida, especialmente en dispositivos móviles.
3. **Escalabilidad** (Requerimiento No Funcional)
  - Capacidad para agregar funcionalidades adicionales en el futuro sin necesidad de una reestructuración completa del sistema.
4. **Personalización y Diseño** (Requerimiento No Funcional)
  - Diseño visual atractivo que refleje la identidad del negocio, utilizando una paleta de colores cálidos y naturales.
  - Evitar elementos que puedan resultar molestos para los usuarios, como pop-ups invasivos o procesos de compra complicados.

### Won't Have

- **Otros elementos mencionados en la conversación:** Cualquier requerimiento adicional que no haya sido mencionado en la conversación no será considerado en esta fase.



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

### Caso de Uso: Realizar Pedidos desde el Catálogo en Línea

<b>Actor:</b> <b>Usuario registrado precondiciones (registrado y logueado)</b>
Permitir al cliente realizar pedidos de productos desde el catálogo en línea
Descripción paso por paso
<ol style="list-style-type: none"><li>5. Inicio de sesión: El cliente registrado inicia sesión en la aplicación utilizando sus credenciales.</li><li>6. <b>Acceso al catálogo:</b> El cliente selecciona la opción de "<b>Catálogo</b>" desde el menú principal de la aplicación.</li><li>7. Visualización de productos: El cliente navega por el catálogo, explorando los productos disponibles, sus descripciones, precios y niveles de disponibilidad.</li><li>8. El Selección de productos:<ol style="list-style-type: none"><li>a. El cliente selecciona uno o más productos para su pedido:</li><li>b. Presiona el botón "<b>Agregar al carrito</b>" en el producto deseado.</li><li>c. Ajusta la cantidad de cada producto en el carrito.</li></ol></li><li>9. Revisión del carrito: El cliente accede a su carrito para revisar los productos seleccionados, verificar cantidades y ver el total del pedido.</li><li>10. Proceder al pago: El cliente presiona el botón "Realizar Pedido" en la página del carrito</li><li>11. El cliente selecciona el método de pago:<ol style="list-style-type: none"><li>a. Tarjeta de crédito/débito.</li><li>b. PayPal u otros.</li></ol></li><li>12. El cliente ingresa los datos de pago.</li><li>13. Confirmación del pedido: El sistema procesa el pedido y realiza las siguientes acciones: Si el pago es exitoso:<ol style="list-style-type: none"><li>a. Muestra una confirmación en pantalla.</li><li>b. Envía un correo electrónico con el recibo del pedido.</li><li>c. Actualiza el inventario y el estado del pedido.</li></ol></li></ol>

**Actor: Usuario registrado precondiciones (registrado y logueado)**

Si el pago es exitoso:

- d. Muestra un mensaje de error
- e. Permite al cliente intentar nuevamente desde el paso 7.

**14. Finalización del pedido:**

El cliente regresa a la página principal o elige seguir navegando en el catálogo.

**Excepciones:**

- Si el inventario de un producto cambia durante el proceso de pedido y no hay stock disponible:
  - El sistema notifica al cliente y elimina el producto del carrito.
  - Permite al cliente continuar con los productos restantes o cancelar el pedido.

## Caso de Uso: Gestión de Inventario

**Actor:** Administrador

Permitir al administrador gestionar el inventario, actualizando el stock y recibiendo alertas sobre productos con niveles críticos.

**Descripción paso por paso**

1. Inicio de sesión: El administrador accede al sistema utilizando sus credenciales.
2. Acceso a la sección de inventario: Selecciona la opción "Inventario" en el menú principal.
3. Visualización del inventario: El sistema muestra una lista de productos con información de stock.
4. Actualización del inventario:
  - a. El administrador selecciona un producto.
  - b. Ingresa la cantidad a agregar o reducir en el stock.
  - c. Presiona el botón "Actualizar".
5. Recepción de alertas:
  - a. Si el stock de algún producto está en niveles críticos, el sistema genera una alerta visible para el administrador.
6. Confirmación: El sistema confirma la actualización del inventario.

**Excepciones:**

**Actor:** Administrador

- Si el sistema no puede actualizar el inventario (fallos técnicos), muestra un mensaje de error y permite reintentar.

### Caso de Uso: Modificar el Estado de un Pedido

**Actor:** Administrador

Permitir al administrador modificar y actualizar el estado de un pedido en tiempo real.

Descripción paso por paso

1. **Inicio de sesión:** El administrador accede al sistema con sus credenciales.
- **Acceso a la sección de pedidos:** Selecciona la opción "Gestión de pedidos" en el menú principal.
- **Búsqueda de un pedido:**
  - El administrador ingresa un criterio de búsqueda (número de pedido, cliente, etc.).
  - El sistema muestra el pedido correspondiente.
- **Modificación del estado del pedido:**
  - El administrador selecciona el pedido.
  - Cambia su estado (por ejemplo, de "Pendiente" a "Enviado").
  - Presiona el botón "Actualizar".
- **Confirmación:**
  - El sistema actualiza el estado del pedido en tiempo real.
  - Muestra un mensaje de confirmación.

**Excepciones:**

- Si el pedido no existe, el sistema muestra un mensaje de error.

### Caso de Uso: Enviar Notificaciones Automáticas de Pedido

**Actor:** Sistema (accionado automáticamente)

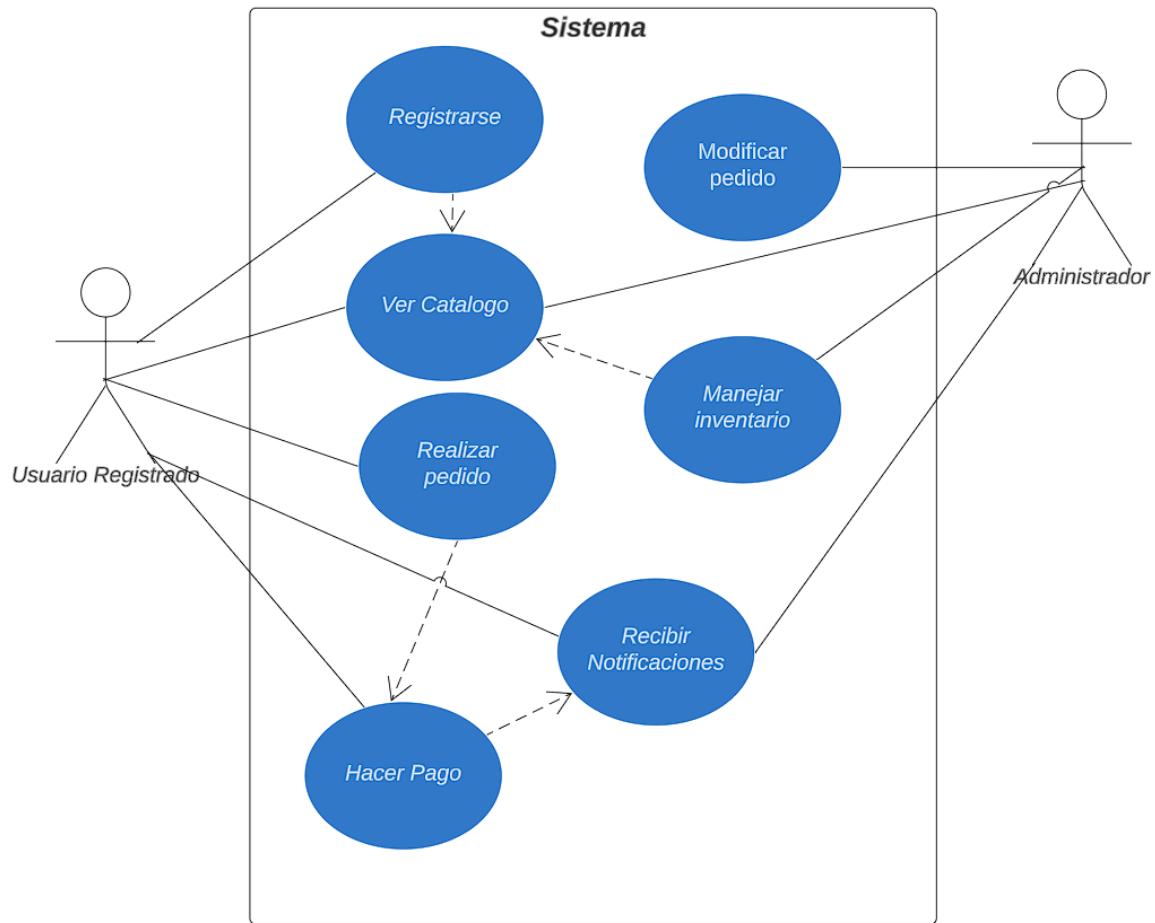
Notificar al cliente sobre cambios en el estado de su pedido.

1. **Detectar un cambio** en el estado de un pedido (por ejemplo, "Procesando" a "Enviado").
2. **Generar una notificación** automática con la información actualizada del pedido.
3. **Enviar la notificación** al cliente mediante correo electrónico o mensaje en la aplicación.
4. **Actualizar el historial** de notificaciones en la cuenta del cliente.

**Excepciones:**

- Si el cliente no tiene configurado un correo electrónico válido, el sistema almacena la notificación en la aplicación.

**Diagrama de Casos de Uso**





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

### Anexo de Documentos Relacionados:

- **Diagrama de Casos de Uso General:** [Insertar enlace o ubicación del documento]
- **Mzanal de Usuario Administrativo:** [Insertar enlace o ubicación del documento]
- **Especificaciones Técnicas del Sistema de Inventario:** [Insertar enlace o ubicación del documento]
- **Políticas de Gestión de Stock y Reabastecimiento:** [Insertar enlace o ubicación del documento]
- **Mockups y Prototipos Visuales:** [Insertar enlaces o ubicaciones de imágenes]
- **Documentación de Integración con Pasarelas de Pago:** [Insertar enlace o ubicación del documento]

### Descripción conceptual

<b>Módulo</b>	Gestión de Pedidos
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	Registro detallado de cada pedido con los datos del cliente, productos solicitados, estado de pago y entrega. Capacidad para modificar y actualizar el estado de los pedidos en tiempo real.

### Descripción técnica

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

#### Backend

URL <a href="https://tusitio.com/api/orders">https://tusitio.com/api/orders</a>	Método POST	Código html 201 Created 400
<b>Descripción</b>		
Al crear un nuevo pedido en el sistema, debe retornar el código 201, en caso de error retorna 400		
<b>Datos de entrada</b> 201 { "cliente": { "nombre": "Carlos López", "telefono": "123456789", "email": "	<b>Datos de salida</b> 201: { "status": "success", "data": { "id": 1001, "cliente": {	

<pre> "carlos.lopez@gmail.com" }, "productos": [     {         "id": 1,         "cantidad": 2         "preciounidad": 1000     }     {         "id": 3         "cantidad": 1         "preciounidad": 2000     } ] "estado": "Pendiente", "pago": "Sin pagar" } </pre>	<pre> "nombre": "Carlos López", "telefono": "123456789", "email": "<a href="mailto:carlos.lopez@gmail.com">carlos.lopez@gmail.com</a>" }, "productos": [     {         "id": 1,         "cantidad": 2         "preciounidad": 1000     }     {         "id": 3         "cantidad": 1         "preciounidad": 2000     } ] "estado": "Pendiente", "pago": "Sin pagar", "total": 130, "fecha": "2024-12-12T10:30:00Z" } </pre>
	<pre> 400: {     "status": "error",     "message": "Faltan campos requeridos o los datos proporcionados son inválidos" } </pre>

URL <a href="https://tusitio.com/api/orders">https://tusitio.com/api/orders</a>	Método GET	Código html 200 OK
<b>Caso de uso técnico</b> Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios		
<b>Datos de entrada</b>		<b>Datos de salida</b> <pre> {     "status": "success",     "data": [         {             "id": 1001,             "cliente": "Carlos López",             "estado": "Pendiente",             "total": 130         },         {             "id": 1002,             "cliente": "María Pérez",             "estado": "Entregado",             "total": 200         }     ] } </pre>

	]
--	---

URL <a href="https://tusitio.com/api/orders">https://tusitio.com/api/orders</a>	Método PATCH	Código html 200 422
<b>Caso de uso técnico</b>		
Al editar, debe retornar código 200 y en caso de error código 422		
<b>Datos de entrada</b>  <b>200:</b> { "estado": "Entregado", "pago": "Pagado" }	<b>Datos de salida</b>  <b>200:</b> { "status": "success", "data": { "id": 1001, "estado": "Entregado", "pago": "Pagado" } }	
	<b>422:</b> { "status": "unprocessable", "message": "No existe el parámetro <nomxbre>" }	

URL <a href="https://tusitio.com/api/products">https://tusitio.com/api/products</a>	Método POST	Código html 200 402
<b>Caso de uso técnico</b>		
Al editar, debe retornar código 200 y en caso de error código 422		
<b>Datos de entrada</b>  <b>200:</b> { "productos": [ { "id": 1, "cantidad": 2 }, { "id": 4, "cantidad": 1 } ] }	<b>Datos de salida</b>  <b>200:</b> { "status": "success", "message": "Productos agregados al pedido con éxito", "data": { "id": 1001, "productos": [ { "id": 1, "nombre": "Producto A", "cantidad": 2 } ] } }	

	<pre>         "cantidad": 2, "precio": 50 },         { "id": 4, "nombre": "Producto D",         "cantidad": 1, "precio": 70 }     ],     "total_actualizado": 270 } } </pre>
	<b>402:</b> <pre> {     "status": "error",     "message": "No se pudo agregar     productos. Datos inválidos o producto sin     stock." } </pre>

### Frontend

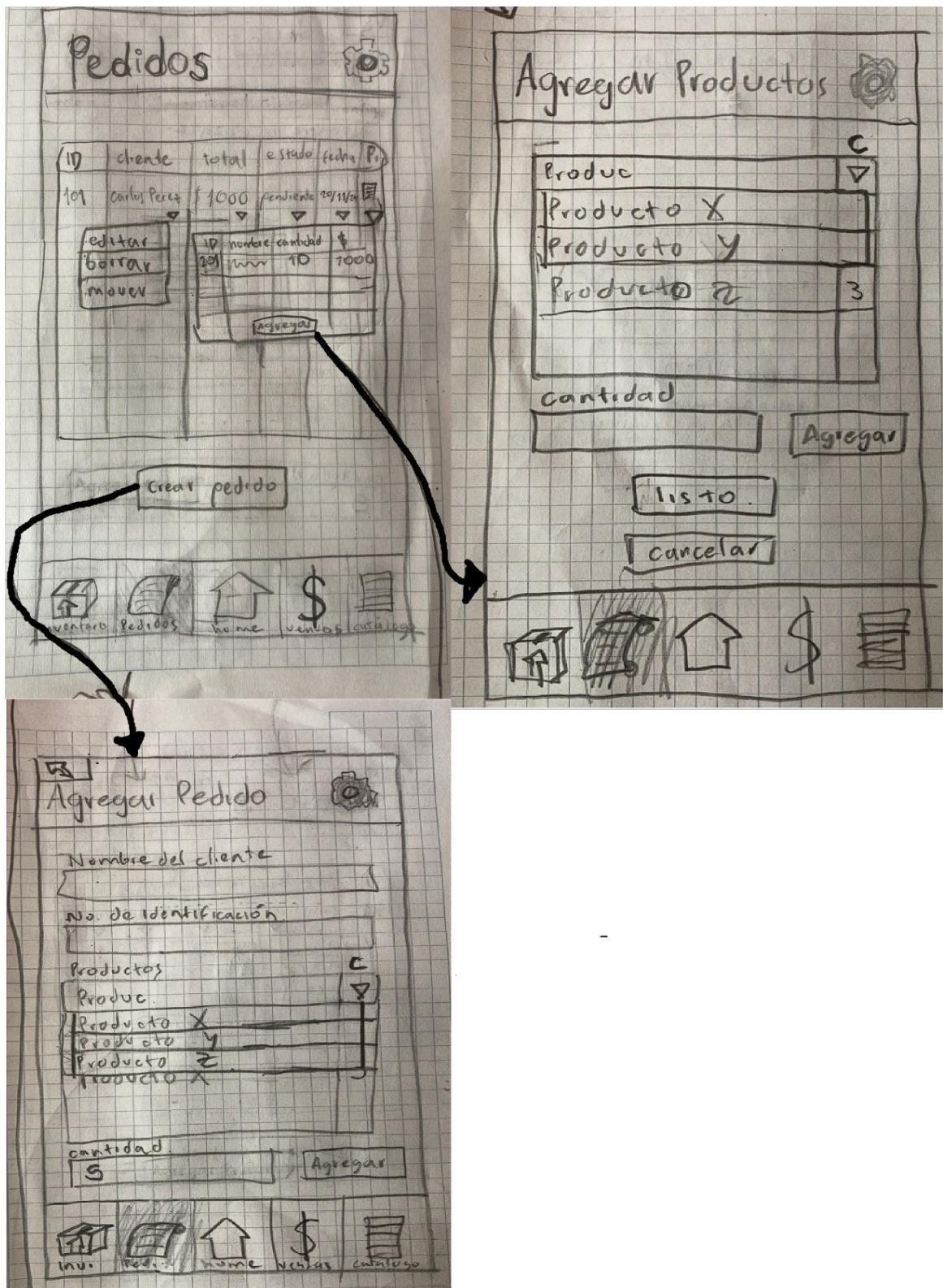
*En este apartado se debe describir no solo con mockups, si no con una descripción narrada lo que debe hacer la HU funcionalmente o visualmente, detalles para que hagan una buena descripción*

#### *Interacción esperada:*

- Al ingresar al módulo de pedidos, se debe mostrar una tabla con los pedidos existentes, incluyendo columnas para ID, cliente, total, estado, y opciones de acción.
- Botón de “Ver productos” que abre una tabla de productos.
- Botón de “Crear pedido” que abre un formulario modal para ingresar los detalles.
- Opciones de editar estado directamente desde la tabla (dropdown en columna).

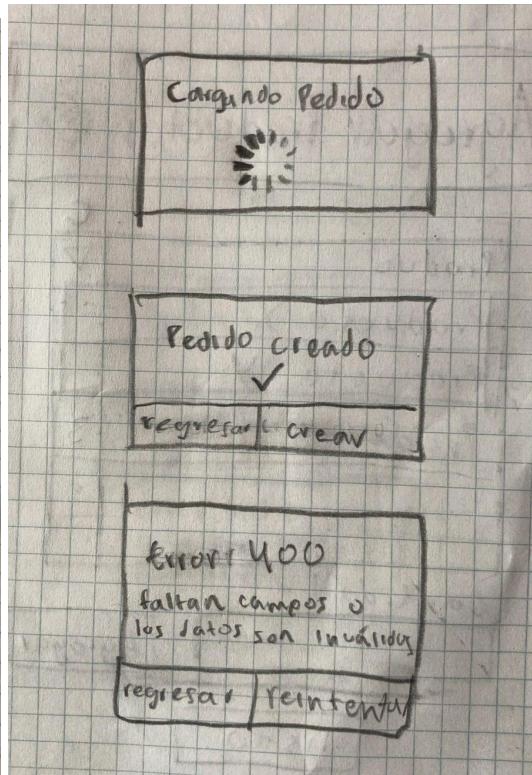
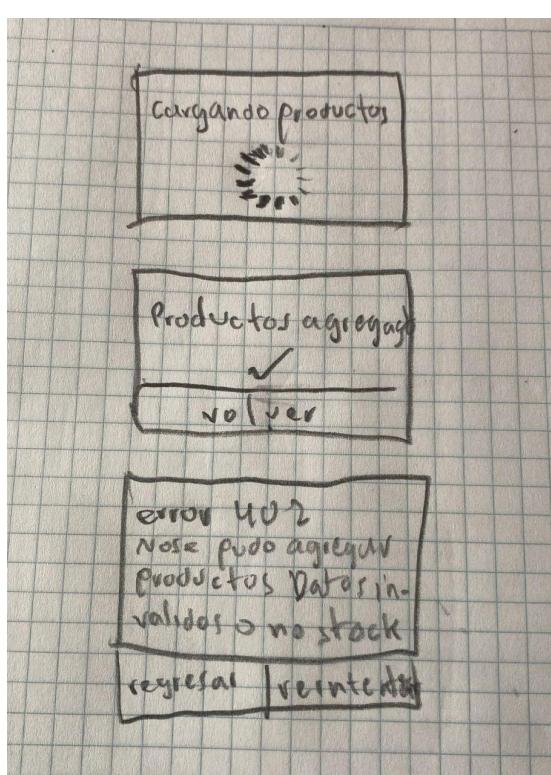
#### *Mockups/Prototipos:*

- Pantalla principal
- Modal: agregar pedidos
- Agregar productos



Flujo visual y eventos:

- Clic en “Crear pedido” -> Abre formulario modal.
- Completar formulario -> Clic en “Guardar” -> Mostrar spinner de carga -> Confirmación o error.
- Dropdown en columna “Estado” -> Selección actualiza pedido en tiempo real.
- Dropdown en columna “Productos” -> Mostrar tabla de productos
- Clic en “Agregar” en dropdown de productos -> Abre modal agregar productos.
- Completar formulario -> Clic en listo -> mostrar spinner de carga -> confirmación o error.





## Historia de Usuario #2

### Anexo de Documentos Relacionados:

- **Diagrama de Casos de Uso General:** [Insertar enlace o ubicación del documento]
- **Manual de Usuario Administrativo:** [Insertar enlace o ubicación del documento]
- **Especificaciones Técnicas del Sistema de inventario:** [Insertar enlace o ubicación del documento]
- **Políticas de Gestión de Stock y Reabastecimiento:** [Insertar enlace o ubicación del documento]
- **Mockups y Prototipos Visuales:** [Insertar enlaces o ubicaciones de imágenes]
- **Documentación de Integración con Pasarelas de Pago:** [Insertar enlace o ubicación del documento]

### Descripción conceptual

Módulo	Control de inventario
Descripción de la(s) funcionalidad(es) requerida(s):	Permitir al administrador gestionar el inventario de productos de manera automática y manual, actualizando los niveles de stock conforme se realizan ventas y generando alertas automáticas cuando el stock de un producto alcanza niveles críticos. Además, proporcionar una interfaz intuitiva y accesible desde dispositivos móviles y computadoras, integrando notificaciones vía WhatsApp y la posibilidad de modificar y visualizar la información de inventario de manera sencilla.

**Descripción técnica****Backend****Obtener Lista de Productos**

URL	Método	Código html
https://tusitio.com/api/inventory	GET	200 OK
<b>Caso de uso técnico</b>		
Retorna una lista completa de productos con sus detalles y niveles de stock actuales.		
Datos de entrada	<b>Datos de salida</b>	
	<pre>{   "status": "success",   "data": [     {       "id": 201,       "nombre": "Pulsera",       "descripcion": "Pulsera artesanal hecha a mano.",       "precio": 15000,       "stock_actual": 15,       "stock_minimo": 10,       "categoria": "Artesanías",       "imagen_url":       "https://tusitio.com/images/pulsera.jpg"     },     ...   ] }</pre>	

**Agregar Nuevo Producto**

URL	Método	Código html
/api/inventory/products	POST	201 Created 400 Bad Request
Agrega un nuevo producto al inventario.		
<b>Datos de entrada</b>  <pre>{   "nombre": "Maceta Pintada",   "descripcion": "Maceta decorativa pintada a mano.",   "precio": 25000,   "stock_inicial": 8,   "stock_minimo": 5,   "categoria": "Artesanías",   "imagen_url":   "https://tusitio.com/images/maceta.jpg" }</pre>	<b>Datos de salida</b>  <b>201 Created:</b> <pre>{   "status": "success",   "data": {     "id": 202,     "nombre": "Maceta Pintada",     "descripcion": "Maceta decorativa pintada a mano.",     "precio": 25000,     "stock_actual": 8,     "stock_minimo": 5,     "categoria": "Artesanías",     "imagen_url":     "https://tusitio.com/images/maceta.jpg"   } }</pre>	
 <pre>{   "nombre": "Maceta Pintada",   "descripcion": "Maceta decorativa pintada a mano.",   "precio": 25000,   "stock_inicial": 8,   "stock_minimo": 5,   "categoria": "Artesanías",   "imagen_url":   "https://tusitio.com/images/maceta.jpg" }</pre>	<b>400 Bad Request:</b> <pre>{   "status": "error",   "message": "Datos del producto inválidos." }</pre>	

### Actualizar Stock de Producto

URL	Método	Código html
/api/inventory/products/{id}	PATCH	200 OK 400 Bad Request 404 Not Found
Actualiza el nivel de stock de un producto existente.		
Datos de entrada	Datos de salida	
{ "stock_actual": 12 }	<b>200 OK:</b>  <pre>{     "status": "success",     "data": {         "id": 201,         "nombre": "Pulsera",         "descripcion": "Pulsera artesanal hecha a mano.",         "precio": 15000,         "stock_actual": 12,         "stock_minimo": 10,         "categoria": "Artesanías",         "imagen_url":         "https://tusitio.com/images/pulsera.jpg"     } }</pre>	
{ "stock_actual": 12 }	<b>400 Bad request:</b>  <pre>{     "status": "error",     "message": "Cantidad de stock inválida." }</pre>	
{ "stock_actual": 12 }	<b>404 Not Found:</b>  <pre>{     "status": "error",     "message": "Producto no encontrado." }</pre>	

### **Eliminar Producto del inventario**

URL	Método	Código html
/api/inventory/products/{id}	DELETE	204 No Content 404 Not Found
Elimina un producto del inventario.		
<b>Datos de entrada</b>		<b>Datos de salida</b>
<b>204 OK:</b> {           "message": "Producto eliminado exitosamente."           }		
<b>404 Not Found:</b> {           "status": "error",           "message": "Producto no encontrado."           }		

### Obtener productos con stock crítico

URL	Método	Código html
/api/inventory/alerts	GET	200 OK
Retorna una lista de productos cuyo stock actual está por debajo del stock mínimo definido.		
<b>Datos de entrada</b> <b>200 OK:</b>		<b>Datos de salida</b> <b>200 OK:</b> <pre>{   "status": "success",   "data": [     {       "id": 203,       "nombre": "Llaveros Personalizados",       "descripcion": "Llaveros con diseños personalizados.",       "precio": 12000,       "stock_actual": 3,       "stock_minimo": 5,       "categoria": "Artesanías",       "imagen_url":       "https://tusitio.com/images/llaveros.jpg"     },     ...   ] }</pre>

### Actualizar Stock Automáticamente Tras Venta

URL	Método	Código html
/api/inventory/products/update-stock	PATCH	200 OK 400 Bad Request
Endpoint interno para actualizar el stock automáticamente cuando se realiza una venta.		
<b>Datos de entrada</b>  <pre>{   "productos_vendidos": [     {"id": 201, "cantidad": 2},     {"id": 202, "cantidad": 1}   ] }</pre>	<b>Datos de salida</b>  <b>200 OK:</b> <pre>{   "status": "success",   "message": "Stock actualizado exitosamente." }</pre>	
 <pre>{   "productos_vendidos": [     {"id": 201, "cantidad": 2},     {"id": 202, "cantidad": 1}   ] }</pre>	 <b>400 Bad Request:</b> <pre>{   "status": "error",   "message": "Datos de venta inválidos." }</pre>	

### Frontend

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

## Control Inventario

ID: _____	Stock
Nombre: _____	Producto 1: _____
Descripción: _____	Producto 2: _____
Stock: _____	⋮
Categoría: _____	⋮
	⋮
<input type="button" value="Editar"/>	⋮
<input type="button" value="Eliminar"/>	⋮

→

Inventario

Pedidos

Home

Ventas

Condiciones

Notificar como leído

Eliminar

Producto 1 agotado

Stock producto 15 actualizado

Bajo stock producto 4

Producto 7 eliminado

...

...

...

## Agregar Producto

Nombre del producto: _____
Descripción: _____
Precio: _____
Stock Actual: _____
Stock Mínimo: _____
Categoría: _____
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>

Inventario

Pedidos

Home

Ventas

Condiciones

**Visualizar Inventario:**

- Al ingresar, el administrador ve una tabla con todos los productos y sus detalles.
- Los productos con stock crítico están resaltados en rojo o con un ícono de alerta.

**Agregar Nuevo Producto:**

- El administrador hace clic en agregar producto.
- Aparece un modal con el formulario para ingresar los detalles del producto.
- Al completar y enviar, el sistema valida la información y actualiza la tabla con el nuevo producto.
- Se envía una notificación de confirmación al administrador.

**Editar Producto:**

- El administrador hace clic en editar en la fila del producto.
- Aparece el formulario prellenado con los datos actuales del producto.
- El administrador actualiza los campos necesarios y envía el formulario.
- El sistema valida y actualiza la información, reflejando los cambios en la tabla.

**Eliminar Producto:**

- El administrador hace clic en eliminar en la fila del producto.
- Se muestra una confirmación modal para proceder.
- Al confirmar, el producto se elimina del inventario y desaparece de la tabla.
- Se envía una notificación de eliminación exitosa.

**Recepción de Alertas de Stock Crítico:**

- Cuando el stock de un producto cae por debajo del mínimo, el sistema envía un mensaje de WhatsApp al administrador.
- Opcionalmente, se envía un resumen semanal por email.
- Las alertas también aparecen en la sección de notificaciones.



## Historia de Usuario #3

### Anexo de Documentos Relacionados:

- **Diagrama de Casos de Uso General:** [Insertar enlace o ubicación del documento]
- **Manual de Usuario Administrativo:** [Insertar enlace o ubicación del documento]
- **Especificaciones Técnicas del Sistema de Inventario:** [Insertar enlace o ubicación del documento]
- **Políticas de Atención y Envío de Pedidos:** [Insertar enlace o ubicación del documento]
- **Mockups y Prototipos Visuales:** [Insertar enlaces o ubicaciones de imágenes]
- **Documentación de Integración con Sistemas de Inventario:** [Insertar enlace o ubicación del documento]

### Descripción conceptual

<b>Módulo</b>	Catálogo en Línea
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	Permitir a los clientes visualizar los productos disponibles, incluyendo precios, descripciones, imágenes y niveles de disponibilidad. Integrar una funcionalidad que permita a los clientes realizar pedidos directamente desde el catálogo, con opción de seleccionar cantidades y registrar sus datos de contacto. La plataforma debe ser accesible desde dispositivos móviles y computadoras, garantizando una experiencia de usuario ágil e intuitiva.

## Descripción técnica

### Backend

#### Obtener Lista de Productos Disponibles

URL	Método	Código http
<a href="https://tusitio.com/api/catalog">https://tusitio.com/api/catalog</a>	GET	200 OK
<b>Caso de uso técnico</b>		
Retorna una lista completa de productos con sus detalles, precios y disponibilidad.		
Datos de entrada	Datos de salida	
	{ "status": "success", "data": [ { "id": 301, "nombre": "Maceta Pintada", "descripción": "Maceta decorativa pintada a mano.", "precio": 25000, "stock_actual": 8, "categoria": "Decoración", "imagen_url": "https://tusitio.com/images/maceta.jpg" }, ... ] } }	

**Realizar Pedido**

URL	Método	Código http
https://tusitio.com/api/catalog/orders	POST	201 Created 400 Bad Request
Registra un nuevo pedido realizado por un cliente.		
Datos de entrada	Datos de salida	
<pre>{   "cliente": {     "nombre": "Juan Pérez",     "email": "juan.perez@example.com",     "telefono": "+573001234567"   },   "productos": [     {       "id": 301,       "cantidad": 2     },     {       "id": 302,       "cantidad": 1     }   ] }</pre>	<p><b>201 Created:</b></p> <pre>{   "status": "success",   "data": {     "pedido_id": 501,     "cliente": {       "nombre": "Juan Pérez",       "email": "juan.perez@example.com",       "telefono": "+573001234567"     },     "productos": [       {         "id": 301,         "nombre": "Maceta Pintada",         "cantidad": 2,         "precio_unitario": 25000,         "subtotal": 50000       },       {         "id": 302,         "nombre": "Pulsera Artesanal",         "cantidad": 1,         "precio_unitario": 15000,         "subtotal": 15000       }     ],     "total": 65000   } }</pre>	
	<b>400 Bad Request:</b>	<pre>{   "status": "error",   "message": "Datos del pedido inválidos." }</pre>

	}
--	---

### Consultar Pedido por ID

URL	Método	Código http
<a href="https://tusitio.com/api/catalog/orders/{id}">https://tusitio.com/api/catalog/orders/{id}</a>	PATCH	200 OK 400 Bad Request 404 Not Found
Actualiza el nivel de stock de un producto existente.		
<b>Datos de entrada</b>		<b>Datos de salida</b>
<b>200 OK:</b>		<b>200 OK:</b> <pre>{   "status": "success",   "data": {     "pedido_id": 501,     "cliente": {       "nombre": "Juan Pérez",       "email": "juan.perez@example.com",       "telefono": "+573001234567"     },     "productos": [       {         "id": 301,         "nombre": "Maceta Pintada",         "cantidad": 2,         "precio_unitario": 25000,         "subtotal": 50000       },       {         "id": 302,         "nombre": "Pulsera Artesanal",         "cantidad": 1,         "precio_unitario": 15000,         "subtotal": 15000       }     ]   } }</pre>

	<pre>        ],         "total": 65000     } }</pre>
	<p><b>404 Not Found:</b></p> <pre>{     "status": "error",     "message": "Pedido no encontrado." }</pre>

### Cancelar Pedido

URL	Método	Código http
<a href="https://tusitio.com/api/catalog/orders/{id}">https://tusitio.com/api/catalog/orders/{id}</a>	DELETE	204 No Content 404 Not Found
Elimina un producto del inventario.		
<b>Datos de entrada</b>		<b>Datos de salida</b> <b>204 No content:</b> <pre>{     "message": "Pedido cancelado exitosamente." }</pre>
		<b>404 Not Found:</b> <pre>{     "status": "error",     "message": "Pedido no encontrado." }</pre>



6. Agregar al readme.md del repositorio la siguiente información estructurada como quieran (libres de organizarla y presentarla como quieran siempre y cuando sea legible):
  - Título del repositorio → **Repositorio grupal - Ingeniería de software 1 - 2024-2**
  - Nombres de los integrantes del equipo
  - Nombre de la aplicación/proyecto
  - Descripción del objetivo del proyecto
  - Tecnologías a utilizar (Frameworks, lenguajes de programación y servicios externos)
  - (Opcional) Icono, imagen, logo de su app
7. 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...**

Aquí la plantilla: [ENLACE A LA PLANTILLA](#)

### Para el proyecto se necesita lo siguiente:

Además de realizar los mismos pasos con su proyecto (levantamiento de requerimientos, análisis, casos de uso) para **TODAS** las funcionalidades obtenidas generar su historia de usuario, pueden colocar por módulo cada historia (un CRUD de usuarios + funcionalidades específicas del módulo de usuario, por ejemplo, asignar rol a un usuario)

### Requerimientos Funcionales

A continuación, se detallan los requerimientos funcionales identificados para EcoScan, organizados por prioridad y asociados a procesos específicos dentro del software:

#### Escáner Inteligente

- **Descripción:** Permite a los usuarios tomar una foto de un objeto para que la aplicación lo clasifique automáticamente en la categoría correcta (reciclaje, orgánico, basura general, etc.).
- **Proceso Asociado:** Procesamiento de imágenes y clasificación automática.
- **Especificaciones Técnicas:**
  - Integración de tecnología de visión por computadora.
  - Algoritmos de aprendizaje automático para mejorar la precisión del reconocimiento.

#### Guías Educativas

- **Descripción:** Proporciona contenido educativo con consejos, datos sobre reciclaje y guías para manejar diferentes tipos de residuos de manera sostenible.
- **Proceso Asociado:** Gestión de contenido educativo.
- **Especificaciones Técnicas:**
  - Base de datos de contenido actualizable.
  - Interfaz de usuario intuitiva para la navegación de guías.

#### Historial de Escaneos (Opcional)

- **Descripción:** Permite a los usuarios registrados guardar un historial de los objetos escaneados para evaluar sus hábitos de reciclaje y recibir consejos personalizados.
- **Proceso Asociado:** Almacenamiento y gestión de datos de usuario.
- **Especificaciones Técnicas:**
  - Sincronización en la nube para usuarios registrados.
  - Interfaz de usuario para visualizar el historial.

#### Registro Opcional de Usuarios

- **Descripción:** Los usuarios pueden registrarse con un correo electrónico o cuenta social para acceder a funcionalidades adicionales como el historial de escaneos.

- **Proceso Asociado:** Gestión de autenticación y perfiles de usuario.
- **Especificaciones Técnicas:**
  - Integración con servicios de autenticación (correo electrónico, redes sociales).
  - Seguridad en el almacenamiento de credenciales.

### Recordatorios y Desafíos Ambientales (Opcional)

- **Descripción:** Envía recordatorios de reciclaje y propone desafíos ambientales, como "Recicla 10 botellas esta semana".
- **Proceso Asociado:** Gestión de notificaciones y motivación del usuario.
- **Especificaciones Técnicas:**
  - Sistema de notificaciones push.
  - Módulo para crear y gestionar desafíos.

### Reconocimiento de Materiales con Consejos Específicos

- **Descripción:** Además de clasificar un objeto, detalla el material del que está hecho y ofrece consejos específicos para su limpieza o preparación para reciclaje.
- **Proceso Asociado:** Análisis detallado de materiales.
- **Especificaciones Técnicas:**
  - Base de datos de materiales y sus características.
  - Algoritmos para identificar y asociar materiales con consejos específicos.

### Escáner de Códigos de Barras (Opcional)

- **Descripción:** Permite a los usuarios escanear el código de barras de un producto para identificar el tipo de material y obtener instrucciones de reciclaje específicas.
- **Proceso Asociado:** Lectura y procesamiento de códigos de barras.
- **Especificaciones Técnicas:**
  - Integración con APIs de bases de datos de productos.
  - Interfaz de usuario para escaneo y visualización de resultados.

### Ranking de Usuarios (Opcional)

- **Descripción:** Implementa un sistema de puntos basado en la cantidad de objetos escaneados correctamente, incluyendo un ranking para fomentar la competencia amigable.
- **Proceso Asociado:** Gamificación y motivación del usuario.
- **Especificaciones Técnicas:**
  - Sistema de puntuación y clasificación.
  - Interfaz para visualizar el ranking.

### Notificaciones Educativas

- **Descripción:** Envía notificaciones con tips ambientales, datos curiosos o recordatorios de hábitos sostenibles.

- **Proceso Asociado:** Distribución de contenido educativo mediante notificaciones.
- **Especificaciones Técnicas:**
  - Módulo de gestión de notificaciones.
  - Base de datos de contenidos educativos para notificaciones.

### **Escáner para Residuos Electrónicos**

- **Descripción:** Identifica componentes electrónicos y brinda instrucciones para su correcta disposición.
- **Proceso Asociado:** Clasificación específica de residuos electrónicos.
- **Especificaciones Técnicas:**
  - Base de datos de componentes electrónicos.
  - Algoritmos especializados para el reconocimiento de residuos electrónicos.

### **Detección de Contaminación Cruzada**

- **Descripción:** Advierte sobre materiales que contaminan el reciclaje y sugiere cómo limpiarlos antes de desecharlos.
- **Proceso Asociado:** Análisis de calidad de residuos.
- **Especificaciones Técnicas:**
  - Algoritmos para detectar residuos contaminados.
  - Sistema de recomendaciones para limpieza.

### **Personalización del Perfil (para Usuarios Registrados)**

- **Descripción:** Permite a los usuarios establecer objetivos ambientales personalizados, como reducir la producción de plástico.
- **Proceso Asociado:** Gestión de perfiles y objetivos de usuario.
- **Especificaciones Técnicas:**
  - Interfaz de usuario para establecer y modificar objetivos.
  - Almacenamiento seguro de preferencias de usuario.

### **Estadísticas Ambientales (Opcional)**

- **Descripción:** Muestra estadísticas sobre el impacto de las acciones de reciclaje de los usuarios, así como estadísticas globales.
- **Proceso Asociado:** Análisis y visualización de datos.
- **Especificaciones Técnicas:**
  - Módulo de recopilación y procesamiento de datos.
  - Herramientas de visualización gráfica para estadísticas.

### **Compartir Logros en Redes Sociales**

- **Descripción:** Permite a los usuarios compartir sus logros o estadísticas en redes sociales para inspirar a otros a reciclar.
- **Proceso Asociado:** Integración con redes sociales.

- **Especificaciones Técnicas:**
  - APIs de integración con las principales redes sociales.
  - Funcionalidad para generar contenido compatible.

### Reconocimiento de Objetos Compuestos

- **Descripción:** Identifica objetos hechos de múltiples materiales y proporciona recomendaciones paso a paso para su correcta clasificación.
- **Proceso Asociado:** Clasificación avanzada de materiales.
- **Especificaciones Técnicas:**
  - Algoritmos para descomponer objetos en materiales individuales.
  - Sistema de recomendaciones secuenciales.

### Temporizador para Residuos Orgánicos

- **Descripción:** Ayuda a gestionar residuos orgánicos indicando tiempos estimados de compostaje.
- **Proceso Asociado:** Gestión de residuos orgánicos.
- **Especificaciones Técnicas:**
  - Funcionalidad de temporizador ajustable.
  - Notificaciones relacionadas con el ciclo de compostaje.

### Reconocimiento Sin Conexión

- **Descripción:** Permite realizar escaneos básicos sin necesidad de conexión a internet, almacenando los datos para sincronizarse posteriormente.
- **Proceso Asociado:** Operación offline y sincronización de datos.
- **Especificaciones Técnicas:**
  - Módulo de reconocimiento local con capacidad limitada.
  - Sistema de almacenamiento temporal y sincronización en la nube.

### Requerimientos No Funcionales

#### Accesibilidad y Usabilidad

- **Descripción:** La plataforma debe ser accesible desde dispositivos móviles y computadoras, con una interfaz intuitiva y adaptada para usuarios con poca experiencia tecnológica.
- **Especificaciones Técnicas:**
  - Diseño responsive compatible con Android e iOS.
  - Cumplimiento de estándares de accesibilidad (WCAG).

#### Disponibilidad y Rendimiento

- **Descripción:** La aplicación debe funcionar predominantemente online, con opciones básicas disponibles offline y tiempos de carga optimizados para una experiencia fluida.
- **Especificaciones Técnicas:**
  - Infraestructura en la nube para alta disponibilidad.
  - Optimización de recursos para minimizar tiempos de carga.

## Seguridad

- **Descripción:** Protección de datos sensibles de los usuarios y transacciones, con acceso restringido a información administrativa.
- **Especificaciones Técnicas:**
  - Encriptación de datos en tránsito y en reposo.
  - Implementación de roles y permisos de acceso diferenciados.

## Escalabilidad

- **Descripción:** La aplicación debe tener la capacidad de agregar funcionalidades adicionales en el futuro sin requerir una reestructuración completa del sistema.
- **Especificaciones Técnicas:**
  - Arquitectura modular y basada en microservicios.
  - Base de datos escalable para manejar incrementos en el volumen de datos.

## Personalización y Diseño

- **Descripción:** Diseño visual atractivo que refleje una identidad ambiental, utilizando una paleta de colores naturales y evitando elementos molestos para el usuario.
- **Especificaciones Técnicas:**
  - Uso de colores y símbolos universales para la clasificación de residuos.
  - Diseño UI/UX centrado en la simplicidad y la facilidad de uso.

8. Agregar al readme.md del repositorio la siguiente información estructurada como quieran (libres de organizarla y presentarla como quieran siempre y cuando sea legible):

- Título del repositorio → **Repositorio grupal - Ingeniería de software 1 - 2024-2**
- Nombres de los integrantes del equipo
- Nombre de la aplicación/proyecto
- Descripción del objetivo del proyecto
- Tecnologías a utilizar (Frameworks, lenguajes de programación y servicios externos)
- (Opcional) Icono, imagen, logo de su app

9. Revisé algunos de sus repositorios, es **IMPORTANTE Y OBLIGATORIO** agregar los .gitignore a sus repositorios, deben personalizarlos con sus tecnologías, esto es un paso que vamos a ver en Clean Code en el próximo módulo, pero desde ya les va a ahorrar dolores de cabeza y un par de puntitos, por tanto:

- Investigar cómo funciona un .gitignore en los repositorios
- implementar el gitignore, les recomiendo esta pagina 😊  
<https://www.toptal.com/developers/gitignore>

**Los entregables deben ser así:**

**Para el taller:**

1 pdf con todo el contenido pedido (puntos 1 al 5) para identificar y que me ahorren tiempo calificando y revisando, organizarlo de la siguiente manera:

Hoja de portada

Punto 1

Punto 2

Hoja en blanco

Punto 3

Hoja en blanco

Punto 4

Hoja en blanco

Punto 5 HU 1

Hoja en blanco

....

Punto 5 HU N

Esto me ayudará a identificar mejor cada parte de su documento y que puedan entregármelo en un solo PDF y no llenarme, ni llenarlos de mil archivos....

Este debe ser nombrado y subido al repositorio en la siguiente ruta:

Asignaciones/Taller\_01\_Requerimientos.pdf

**Para el proyecto:**

Aquí sí nos llenamos de documentos 😞

Todos los documentos deben estar separados y organizados de la siguiente manera:

Documentación/Proyecto/Levantamiento\_de\_requerimientos.pdf

Documentación/Proyecto/Análisis\_de\_Requerimientos.pdf

Documentación/Proyecto/Diagrama CU Administrador

Documentación/Proyecto/Diagrama CU Usuario

Documentación/Historias\_de\_Usuario/historia\_01\_<nickname>\_<modulo>.pdf

**Ejemplo:**

Documentación/Historias\_de\_Usuario/historia\_01\_oalvarezr\_inventario.pdf

**ACLARACION IMPORTANTE SUMAMENTE IMPORTANTE PARA QUE NO  
ENTREN EN CRISIS** 

El taller debe ser entregado hasta el punto 5 en la fecha estipulada, sin posibilidad a negociación, con respecto al proyecto no hay fecha límite, sin embargo **sería un indicativo negativo** que lleguen a esa fecha sin estos entregables.