



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá

Facultad de Ingeniería

Departamento de Sistemas e Industrial

Curso: Ingeniería de Software 1

Daniel Alejandro Duitama Correa

Edwin Felipe Pinilla Peralta

Miguel Angel Martinez Fernandez

Juan Sebastian Umaña Camacho

Taller requerimientos de usuario

Punto 2

2.1. Levantamiento de requerimientos

Contexto: La aplicación está destinada a una emprendedora que gestiona un negocio de venta de productos artesanales y decorativos por Internet. Actualmente utiliza herramientas manuales como WhatsApp y Excel, lo que genera problemas de organización, seguimiento de pedidos e inventario. El objetivo es crear un software que facilite la gestión de su negocio y potencie su crecimiento.

Procesos Clave y Funcionalidades:

1. Gestión de Pedidos:

- Crear tareas por pedido, que incluyan:
 - Nombre del cliente.
 - Productos solicitados (con cantidades).
 - Estado del pedido (Pendiente, Pagado, Enviado, Completado, Cancelado).
 - Fecha límite de entrega.
- Posibilidad de modificar los datos del pedido en caso de cambios.
- Registro automático de pedidos en el historial del cliente.

2. Control de Inventario:

- Registro de productos con los siguientes atributos:
 - Nombre, precio, descripción, y cantidad disponible.
 - Fotos del producto.
- Actualización automática del inventario al completar tareas/pedidos.
- Alertas de stock bajo enviadas vía WhatsApp.

3. Catálogo en Línea:

- Plataforma accesible para clientes donde puedan:
 - Ver productos disponibles organizados en categorías.
 - Consultar precios y descripciones.
 - Iniciar pedidos rellenando un formulario.
- Integración con el inventario para mostrar disponibilidad en tiempo real.

4. Gestión de Finanzas:

- Registro automático de ventas y pagos.
- Opciones para registrar gastos relacionados con compras de inventario o materiales.
- Generación de reportes básicos de ingresos, egresos y ganancia neta.

5. Notificaciones y Comunicación:

- Notificaciones automáticas para clientes al registrar pedidos, indicando:
 - Resumen del pedido.
 - Confirmación de registro.
 - Detalles de envío.
- Integración de notificaciones para la administradora sobre:
 - Pedidos nuevos.
 - Productos con bajo stock.
 - Resúmenes semanales de inventario enviados por email.

6. Flexibilidad de Crecimiento:

- Arquitectura que permita agregar módulos futuros, como:
 - Gestión de citas para asesorías.
 - Promociones automatizadas.
 - Comentarios y opiniones de clientes.

Especificaciones Técnicas:

1. Interfaz:

- Interfaz web responsiva para acceso desde dispositivos móviles y computadoras.
- Diseño limpio, organizado y amigable para usuarios no técnicos.
- Uso de colores cálidos como beige, terracota, gris y verde oliva.

2. Datos a Almacenar:

- Productos: Nombre, precio, stock, descripción, fotos.
- Clientes: Nombre, contacto, dirección, historial de pedidos.
- Pedidos: ID, fecha, productos solicitados, estado, total, medio de pago.
- Finanzas: Ingresos, egresos, ganancia neta.

3. Notificaciones:

- WhatsApp como canal principal para notificaciones urgentes.
- Emails para resúmenes y reportes semanales.

4. Acceso y Seguridad:

- Acceso restringido a la administradora para la gestión completa.
- Clientes con acceso limitado al catálogo y formulario de pedidos.

5. Integraciones:

- Pasarelas de pago locales (Nequi, Daviplata) para facilitar transacciones.
- Posibilidad de incluir una calculadora de envío basada en la ubicación del cliente.

6. Almacenamiento de Datos:

- Formulario para ingreso manual por parte de la administradora.
- Datos de clientes registrados automáticamente al realizar pedidos.

Punto 3

3.1.Listado de Funcionalidades Clasificadas con el Método MoSCoW:

Must Have:

- Gestión de Pedidos: Crear tareas por pedido con los datos del cliente, productos, estado y fecha límite.
- Gestión de Pedidos: Modificar datos del pedido.
- Gestión de Pedidos: Registrar pedidos automáticamente en el historial del cliente.
- Control de Inventario: Registro de productos con nombre, precio, descripción y stock.
- Control de Inventario: Actualización automática del inventario al completar pedidos.
- Control de Inventario: Alertas de stock bajo.
- Catálogo en Línea: Visualización de productos por categorías.
- Catálogo en Línea: Integración con inventario para mostrar disponibilidad.
- Notificaciones: Notificaciones automáticas para clientes y administradora por WhatsApp.

Should Have:

- Gestión de Finanzas: Registro automático de ingresos por ventas.
- Gestión de Finanzas: Registro de egresos.
- Gestión de Finanzas: Reportes de ingresos, egresos y ganancia neta.
- Notificaciones Avanzadas: Resúmenes semanales de inventario por email.
- Notificaciones Avanzadas: Confirmación de pedido con link al cliente.

Could Have

- Flexibilidad de Crecimiento: Arquitectura escalable para agregar módulos futuros como gestión de citas o promociones.
- Calculadora de Envío: Funcionalidad para estimar costos de envío según ubicación.
- Sección de Opiniones: Espacio donde los clientes puedan dejar comentarios sobre los productos.

Won't Have

- Desarrollo de una aplicación móvil nativa.

3.2.Estimado de Tiempo y Complejidad

Recursos Asumidos:

Cantidad de desarrolladores: 1 Full Stack.

Tecnologías propuestas:

- Frontend: React.js para una interfaz web responsiva y moderna.
- Backend: Node.js con Express para la lógica de negocio.
- Base de datos: MongoDB para flexibilidad en la gestión de datos.
- Notificaciones: Integración con Twilio API (WhatsApp) y nodemailer (email).
- Hosting: Servicios en la nube como AWS para despliegue y escalabilidad.

Estimación de Desarrollo:

Duración Total Estimada: 16 semanas (4 meses), considerando un solo desarrollador.

Costo Aproximado:

- Tasa promedio por desarrollador: 1300000
- Horas estimadas por semana por desarrollador: 24
- Costo total: 5200000

Punto 4

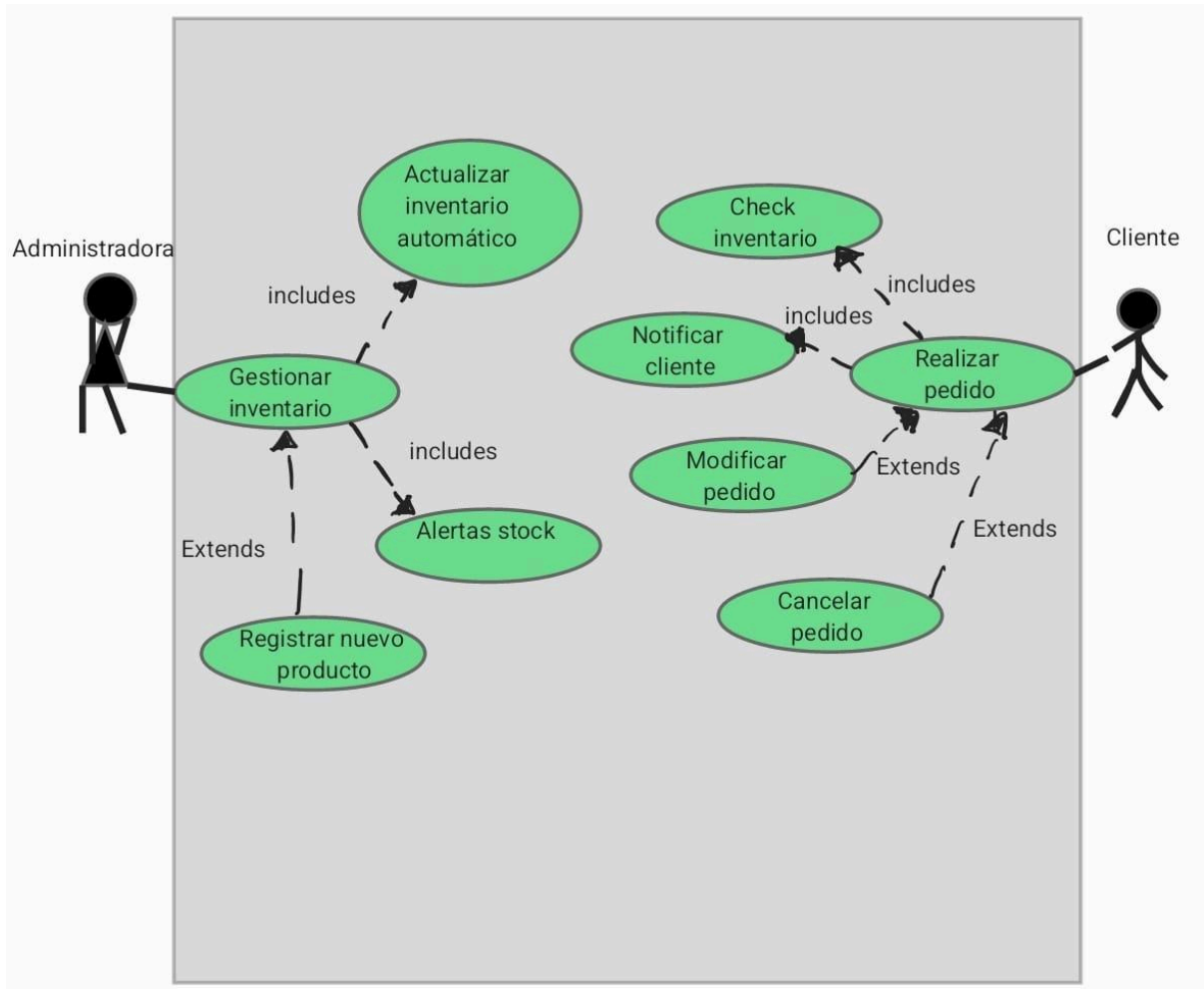
Crear y gestionar pedidos.
Actor Administradora
Breve descripción La administradora puede registrar nuevos pedidos, actualizarlos y ver su estado.
Precondiciones <ul style="list-style-type: none">• La administradora ha ingresado al aplicativo con sus credenciales.
Descripción paso a paso <ol style="list-style-type: none">1. La administradora inicia sesión en la aplicación2. Accede al módulo de pedidos3. Introduce los datos del cliente (nombre, contacto, dirección)<ul style="list-style-type: none">○ Si los datos del cliente están incompletos, se solicita corrección antes de continuar.4. Selecciona los productos, sus cantidades y la fecha de entrega.<ul style="list-style-type: none">○ Si algún producto no está disponible se muestra una alerta y la administradora puede actualizar el inventario.5. Confirma la creación del pedido6. El sistema guarda el pedido y lo registra en el historial del cliente.7. Notifica al cliente sobre el nuevo pedido.
Relaciones (Posibles conexiones con otros casos de uso)

Realizar pedidos
Actor Cliente
Breve descripción Los clientes pueden iniciar pedidos desde el catálogo en línea.
Precondiciones <ul style="list-style-type: none">• El catálogo debe estar disponible en línea.
Descripción paso a paso

1. El cliente accede al catálogo en línea.
2. Navega por las categorías y selecciona productos.
 - Si algún producto no tiene suficiente stock, el sistema lo indica y el cliente puede ajustar las cantidades
3. Rellena un formulario de pedido con sus datos de contacto.
 - Si no se proporciona un número de contacto válido, el sistema no permite proceder con el pedido.
4. Confirma el pedido.
5. El sistema verifica la disponibilidad en inventario.
6. Notifica al cliente sobre la creación del pedido

Relaciones

(Posibles conexiones con otros casos de uso)



Punto 5

5.1. Historias de Usuario - Backend

HU 1.

URL	Método	Código http
POST /api/orders	POST	<ul style="list-style-type: none">201 (Pedido creado correctamente)400 (Error en los datos de entrada)
Caso de uso técnico <ul style="list-style-type: none">Al realizar la solicitud, debe retornarse un código 201 si el pedido se almacena correctamente en la base de datos.En caso de error en los datos de entrada, retornar un código 400 con el detalle del error.		
Datos de entrada 200: <pre>{ "cliente": "Juan Pérez", "productos": [{"id": 1, "cantidad": 2}, {"id": 3, "cantidad": 1}], "estado": "Pendiente", "fechaEntrega": "2024-12-30"}</pre>	Datos de salida 201: <pre>{ "status": "success", "data": { "id": 101, "cliente": "Juan Pérez", "productos": [{"id": 1, "cantidad": 2}, {"id": 3, "cantidad": 1}], "estado": "Pendiente", "fechaEntrega": "2024-12-30" }}</pre>	
422: <pre>{ "cliente": "", "productos": [{"id": 1, "cantidad": 2}, {"id": 3, "cantidad": 1}], "estado": "Pendiente", "fechaEntrega": "2024-12-30"}</pre>	422: <pre>{ "status": "error", "message": "El campo cliente es obligatorio."}</pre>	

URL	Método	Código http
PATCH/api/products/{id}	PATCH	<ul style="list-style-type: none">200 (Actualización realizada con

		éxito) <ul style="list-style-type: none"> 404 (Producto no encontrado)
Caso de uso técnico <ul style="list-style-type: none"> Al recibir una solicitud válida, debe actualizarse el stock del producto en la base de datos y retornar un código 200. Si el producto no existe, retornar un código 404. 		
200: Datos de entrada	200: Datos de salida <pre>{ "status": "success", "data": { "id": 1, "nombre": "Pulsera artesanal", "stock": 8 } }</pre>	
404:	404: <pre>{ "status": "error", "message": "El producto con ID 2 no existe." }</pre>	

URL POST/api/notifications	Método POST	Código http <ul style="list-style-type: none"> 200 (Notificación enviada correctamente) 500 (Error en el servidor)
Caso de uso técnico <ul style="list-style-type: none"> Al recibir una solicitud válida, debe actualizarse el stock del producto en la base de datos y retornar un código 200. Si el producto no existe, retornar un código 404. 		
200: Datos de entrada <pre>{ "numeroCliente": "+573001234567", "mensaje": "Tu pedido #101 ha sido marcado como Enviado." }</pre>	200: Datos de salida <pre>{ "status": "success", "message": "Notificación enviada correctamente." }</pre>	
500:	500:	

<pre>{ "numeroCliente": "+57300123", "mensaje": "Tu pedido #102 ha sido marcado como Enviado." }</pre>	<pre>{ "status": "error", "message": "El número introducido debe tener 10 dígitos" }</pre>
--	--

HU 2.

Descripción conceptual

- Integración de notificaciones para la administradora sobre:
 - Pedidos nuevos.
 - Productos con bajo stock.

Módulo	<i>Módulo de Inventario</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<p><i>Se requiere una funcionalidad que notifique al administrador del inventario cuando:</i></p> <ol style="list-style-type: none"><i>1. Se registre un nuevo pedido.</i><i>2. Un producto alcance niveles críticos de stock (por debajo de un umbral configurable).</i> <p><i>Las notificaciones deben ser enviadas vía correo electrónico y mostradas en una sección de "Alertas" dentro del sistema de gestión.</i></p>

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/notification s/new-order	POST	<ul style="list-style-type: none">● 201: Notificación creada exitosamente● 400: Error en los datos enviados
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de los usuarios		
Datos de entrada <pre>{ "orderId": 123, "productName": "Producto X", "quantity": 10</pre>	Datos de salida <pre>{ "status": "success", "message": "Notificación enviada al administrador." }</pre>	

}	
---	--

URL	Método	Código html
localhost:8080/notifications/low-stock	POST	200 422
Datos de entrada <pre>{ "productId": 456, "productName": "Producto Y", "currentStock": 5 }</pre>		Datos de salida <pre>{ "status": "success", "message": "Notificación de bajo stock enviada." }</pre>

Frontend

Para la HU funcionalidad de Notificaciones de Stock

Correo Electrónico:

- Plantilla de email automático para cada tipo de notificación.
- Ejemplo de correo:
 - Asunto: "[Alerta] Nuevo Pedido Registrado"
 - Cuerpo:
 - "Se ha registrado un nuevo pedido con ID 123 que incluye el producto Producto X. Por favor revise el sistema para más detalles."
- Asunto: "[Alerta] Producto con Bajo Stock"
 - Cuerpo:
 - "El producto Producto Y ha alcanzado un nivel de stock crítico (5 unidades). Se recomienda reabastecerlo lo antes posible."

Datos de entrada

- Desde el formulario de configuración: Umbral de stock crítico para cada producto.

Datos de salida

Notificaciones visibles en el frontend y enviadas por email.

HU 3.

Descripción conceptual

Módulo	Gestión de Pedidos
Descripción de la(s) funcionalidad(es) requerida(s):	<p><i>El sistema debe permitir la creación automática de tareas asociadas a cada pedido. Estas tareas deben incluir:</i></p> <ul style="list-style-type: none">• <i>Nombre del cliente.</i>• <i>Productos solicitados, especificando las cantidades.</i>• <i>Estado del pedido (Pendiente, Pagado, Enviado, Completado, Cancelado).</i>• <i>Fecha límite de entrega.</i> <p><i>Las tareas serán visibles y editables desde el panel de administración para su gestión.</i></p>

Descripción técnica

Backend

URL	Método	Código html
/api/orders/create-task	POST	Endpoint encargado de generar una tarea basada en la información del pedido recibido.
Descripción		
Datos de entrada <pre>{ "order_id": 456, "client_name": "Juan Pérez", "products": [{ "product_id": 123, "name": "Producto A", "quantity": 2 }, { "product_id": 456, "name": "Producto B", "quantity": 1 }], "status": "Pendiente",</pre>	Datos de salida <pre>Codigo 200 (ok) { "status": "success", "task_id": 789, "message": "Tarea creada exitosamente." }</pre>	

<pre>"delivery_deadline": "2024-12-30" }</pre>	
--	--

URL	Método	Descripción
/api/orders/update-task	PATCH	Actualizar la información de una tarea de pedido existente.
Descripción		
Datos de entrada <pre>{ "task_id": 789, "status": "Enviado" }</pre>		Datos de salida <pre>Codigo 200 (ok) { "status": "success", "message": "Tarea actualizada exitosamente." }</pre>

Frontend

Para la HU 3.

Vista en el Panel de Administración:

- Mostrar un listado de tareas por pedido con columnas:
 - Nombre del cliente.
 - Productos y cantidades.
 - Estado del pedido (editable).
 - Fecha límite de entrega.
- Incluir un botón para marcar la tarea como "Completada".
- Resaltar con colores los pedidos según su estado:
 - Rojo: Pendiente.
 - Amarillo: Enviado.
 - Verde: Completado.

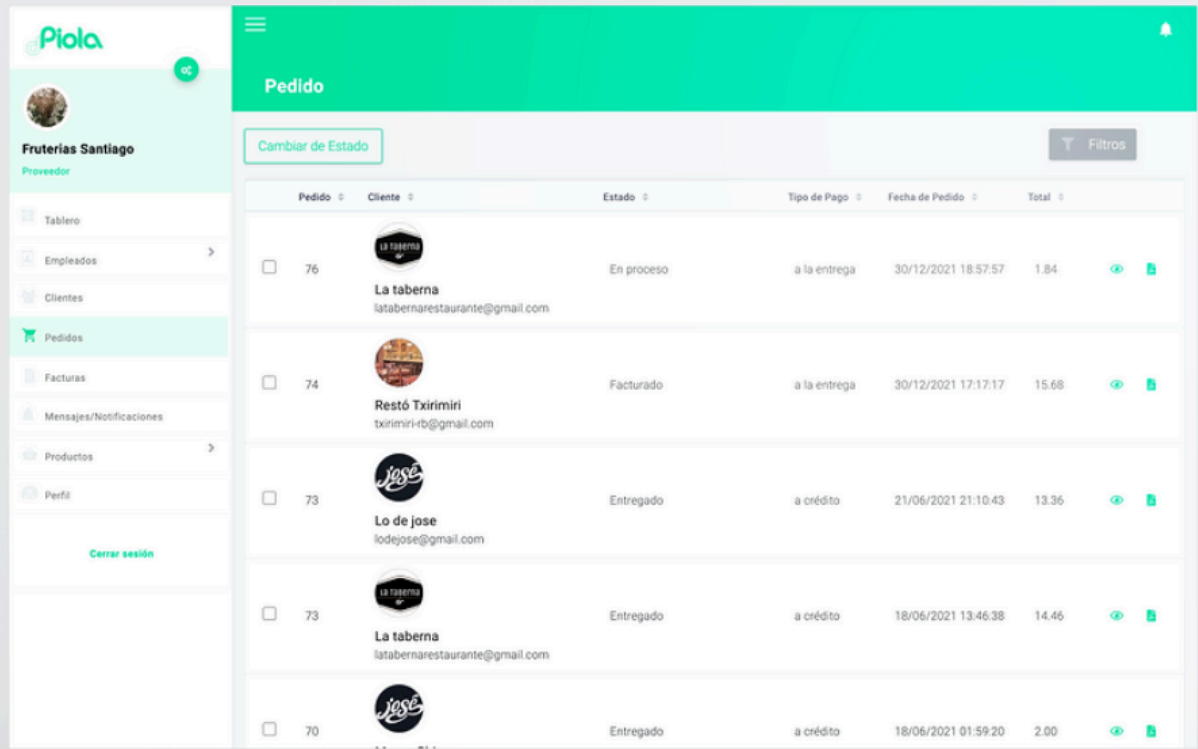
Mock Up Gestión de pedidos:

Nombre del cliente: Visualización clara del nombre para identificar rápidamente al comprador.






Productos solicitados y cantidades: Listado de artículos con sus respectivas cantidades para facilitar la preparación del pedido.

Estado del pedido: Indicadores visuales que reflejan si el pedido está Pendiente, Pagado, Enviado, Completado o Cancelado.

Fecha límite de entrega: Campo que muestra la fecha estimada de entrega para asegurar el cumplimiento de los plazos.



The screenshot shows the Piola app interface for a provider named "Fruterías Santiago". The left sidebar contains navigation options: Tablero, Empleados, Clientes, Pedidos (highlighted), Facturas, Mensajes/Notificaciones, Productos, and Perfil. The main area displays a list of orders under the heading "Pedido". The table has columns for Pedido, Cliente, Estado, Tipo de Pago, Fecha de Pedido, and Total. The data is as follows:

Pedido	Cliente	Estado	Tipo de Pago	Fecha de Pedido	Total
<input type="checkbox"/> 76	 La taberna latabernaestaurante@gmail.com	En proceso	a la entrega	30/12/2021 18:57:57	1.84
<input type="checkbox"/> 74	 Restó Txirimiri txirimiri-rb@gmail.com	Facturado	a la entrega	30/12/2021 17:17:17	15.68
<input type="checkbox"/> 73	 Lo de Jose lodejose@gmail.com	Entregado	a crédito	21/06/2021 21:10:43	13.36
<input type="checkbox"/> 73	 La taberna latabernaestaurante@gmail.com	Entregado	a crédito	18/06/2021 13:46:38	14.46
<input type="checkbox"/> 70	 Lo de Jose lodejose@gmail.com	Entregado	a crédito	18/06/2021 01:59:20	2.00

Recuperado de: <https://www.piolapp.com/en/proveedor>

