



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - Sede Bogotá

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

Asignatura:

Ingeniería de Software I

Taller del módulo de requerimientos

Natalia Carolina Bautista Sanchez

Daniel Santiago Cocinero Jimenez

Baruj Vladimir Ramirez Escalante

Juan Daniel Ramirez Mojica

Profesor:

Oscar Eduardo Alvarez Rodriguez

2024 - II

Bogotá D.C.

Índice

Primer Punto	3
Selección de Conversación para Desarrollo del Taller	3
Segundo Punto	3
Redacción del Levantamiento de Requerimientos	3
1. Contexto:	3
2. Resumen inicial del problema:	3
3. Requerimientos identificados:	6
Tercer Punto	9
Análisis de Requerimientos y Clasificación con MoSCoW	9
1. Preguntas que surgieron:	9
2. Clasificación MoSCoW:	10
3. Estimación	11
Cuarto Punto	16
Diagrama de Caso de Uso	16
Caso de Uso 1: Consultar costo de envío	16
Caso de Uso 2: Creación de carritos	17
Caso de Uso 3: Buscar producto	17
Caso de Uso 4: Calificar producto	18
Caso de Uso 5: Iniciar sesión	18
Caso de Uso 6: Gestionar inventario	19
Caso de Uso 7: Gestionar usuarios	19
Caso de Uso 8: Consultar reporte de ventas	20
Quinto Punto	23
Historias de Usuario	23
Historia de Usuario #1	23
Historia de Usuario #2	28
Historia de Usuario #3	31
Historia de Usuario #4	36
Interacción esperada:	37
Flujo visual y eventos:	37
Notas Adicionales:	37

Primer Punto

Selección de Conversación para Desarrollo del Taller

Para el desarrollo del taller, se decidió trabajar con la conversación del **GP2** ([CONVERSACIÓN](#)), ya que esta logró proporcionar explicaciones detalladas y claras. De este modo, se busca facilitar el análisis y la definición de requerimientos, permitiendo una mayor precisión al traducir la información en funcionalidades concretas.

Segundo Punto

Redacción del Levantamiento de Requerimientos

1. Contexto:

Se presenta un nuevo cliente, una pequeña empresa (o emprendimiento) en sus etapas iniciales. Este cliente tiene poca experiencia en el uso de tecnologías y en la gestión de negocios y busca orientación para organizar sus procesos de ventas y operaciones. Su actividad principal es la venta de productos artesanales y accesorios decorativos, con la intención de ofrecer asesorías en decoración. Sin embargo, aún no tiene claridad sobre en qué áreas enfocarse ni cómo llevarlas a cabo de manera efectiva.

Actualmente, gestiona sus operaciones de manera manual, utilizando herramientas como WhatsApp y Excel; lo cual le ha generado problemas de organización, como la falta de control sobre el inventario, dificultades para identificar pagos recibidos y la necesidad de responder repetitivamente a consultas de los clientes sobre su catálogo. Además, el cliente está interesado en mejorar la experiencia del usuario mediante herramientas como recordatorios automáticos, un calendario para agendar asesorías y un sistema de gestión de ventas.

Adicionalmente, muestra un alto grado de indecisión y necesita una solución que le permita estructurar sus procesos de manera sencilla y efectiva.

2. Resumen inicial del problema:

El sistema a desarrollar es una plataforma de gestión para un negocio de artesanías y productos personalizados. Este software permitirá al administrador gestionar productos, clientes y pedidos de manera eficiente, además de ofrecer a los clientes una experiencia de compra amigable, rápida y personalizada.

01. Gestión de Productos: El cliente necesita mantener un control sobre su inventario, permitiendo la adición, edición y eliminación de productos.

Funcionalidades:

- a. Almacenar información de cada producto, incluyendo datos como: nombre, precio, cantidad disponible, descripción y fotos opcionales.
- b. Permitir la edición de datos de productos existentes, como ajustes de precio o actualización de stock.
- c. Visualizar una lista de productos con su cantidad actual en stock.
- d. Generar alertas para productos con bajo inventario e informarlos.

Especificaciones técnicas:

- a. Base de datos para almacenar los productos.
- b. Interfaz administrativa con formularios para añadir, editar o eliminar productos.
- c. Sistema de notificaciones (internas) para el inventario bajo por medio de WhatsApp.

02. Gestión de Pedidos: El sistema debe registrar pedidos hechos por los clientes, permitiendo realizar cambios de manera flexible.

Funcionalidades:

- a. Registrar un pedido a través de un formulario que permita ingresar datos del cliente y los productos solicitados.
- b. Dar la opción para que los clientes completen un formulario para iniciar el pedido.
- c. Visualizar un historial de pedidos con detalles como: cliente, productos solicitados, estado del pedido, fecha de solicitud y monto total.
- d. Editar datos de pedidos, permitiendo ajustes en productos, cantidades o dirección de envío.
- e. Filtrar pedidos según estado (Pendiente, Enviado, Completado, Cancelado).

Especificaciones técnicas:

- a. Base de datos para pedidos, vinculada a clientes y productos.
- b. Interfaz con opciones de filtrado por estado o fecha.
- c. Sistema de actualización en tiempo real de los pedidos.

03. Gestión de Clientes: La plataforma debe permitir la creación y actualización de registros de clientes.

Funcionalidades:

- a. Registrar a un cliente incluyendo nombre, contacto, dirección.
- b. Permitir la edición de los datos de los clientes para actualizar información.
- c. Visualizar un listado de clientes con su historial de pedidos.

Especificaciones técnicas:

- a. Base de datos para clientes.
- b. Interfaz administrativa para búsquedas y actualizaciones de datos de cliente.

04. Visualización de Finanzas: Se desea que permita visualizar métricas financieras relacionadas con el negocio (más adelante).

Funcionalidades:

- a. Registrar automáticamente los ingresos por pedidos y sus gastos.
- b. Generar un resumen de ingresos, gastos y ganancias.
- c. Visualizar gráficos para tendencias de ventas y productos populares.

Especificaciones técnicas:

- a. Integración con datos de pedidos y costos registrados.
- b. Panel financiero con gráficos.

05. Diseño y Experiencia de Usuario: El sistema debe tener un diseño atractivo y funcional, adaptado a las necesidades del administrador y sus clientes.

Funcionalidades:

- a. Ofrecer una navegación intuitiva con menús claros y un diseño limpio.

- b. Mostrar un resumen rápido en el inicio para el administrador, incluyendo pedidos recientes, productos en bajo stock y métricas.
- c. Incluir un banner destacado para los clientes, mostrando productos populares o en oferta (más adelante).
- d. Implementar una herramienta para calcular costos de envío antes de completar el pedido.

Especificaciones técnicas:

- a. Usar framework para el frontend y realizar interfaces llamativas e interactivas.
- b. Integrar APIs para el cálculo de los costos de envío.

06. Personalización con el Cliente: El sistema debe mostrar la personalización y conexión emocional con los usuarios.

Funcionalidades:

- a. Incluir una sección donde los clientes puedan dejar reseñas y subir fotos de los productos adquiridos.
- b. Mostrar mensajes de bienvenida “nota personalizada” en la página de inicio.
- c. Crear una galería de inspiración que muestre ejemplos de productos que estén usando los clientes.

Especificaciones técnicas:

- a. Integración de una sección de reseñas vinculada a pedidos completados.
- b. Funcionalidad para la moderación de contenido por parte del administrador.

3. Requerimientos identificados:

- 01. Registro e inicio de sesión: Módulo para crear y autenticar usuarios.
- 02. Catálogo de productos: Visualización de productos con soporte para búsqueda.

03. Carrito de compras: Permite a los usuarios agregar productos, revisar cantidades y proceder a la pasarela de pagos.
04. Calculadora de costo de envío.
05. Pasarela de pagos.
06. Notificaciones automáticas de pedidos por WhatsApp al cliente.
07. Notificaciones automáticas de stock bajo por WhatsApp al administrador.
08. Resumen semanal por email con el estado general del inventario
09. Dashboard administrativo: Panel para gestionar inventarios, pedidos y usuarios.
10. Reportes de ventas (más adelante).
11. Sistema de reseñas: Los usuarios pueden calificar y comentar productos.
12. Historial de pedidos: Registro de compras del usuario.
13. Búsqueda de productos con filtro.
14. Banner destacado de productos (más adelante).
15. Pie de página.
16. Landing Page.
17. Galería de inspiración.

Tercer Punto

Análisis de Requerimientos y Clasificación con MoSCoW

1. Preguntas que surgieron:

Gestión de Productos

- ¿Qué tan detallada debe ser la descripción de los productos?
 - Esto para saber qué información se almacenará en la base de datos, permitiendo organizar bien la información.
- ¿El cliente necesita categorías para organizar los productos? ¿Cuáles?
 - Esto dado que afecta cómo los usuarios navegan en la plataforma y no se han definido de manera clara las categorías para el catálogo de producto o la búsqueda con filtros.

Gestión de Pedidos

- ¿Qué ocurre si un cliente quiere modificar un pedido?
 - Esta pregunta es importante dado a que nos permite conocer cómo se manejan las excepciones y así, se evitan problemas.

Notificaciones Automáticas

- ¿Qué tan frecuentes deben ser las notificaciones?
 - Esto porque demasiadas notificaciones podrían molestar al usuario, mientras que muy pocas podrían hacer que no se noten o no sean efectivas.
- ¿El cliente tiene una plantilla específica para los mensajes automáticos, o se debe crear una?

Visualización de Finanzas

- ¿Qué nivel de detalle requieren los reportes?
 - Esto nos permite saber qué tipo de información se desea ver, su detalle y qué tipo de estadísticas son importantes.

Otros:

- ¿Es necesario proporcionar un manual o guía para explicar cómo utilizar el producto?
 - Seguramente sí, porque está relacionado con la documentación.

2. Clasificación MoSCoW:

Recordando, el método MoSCoW clasifica los requerimientos en:

- Must Have (Imprescindibles): Necesarios para que el sistema funcione.
- Should Have (Importantes): Aportan valor significativo pero no son críticos.
- Could Have (Deseables): Mejoran la experiencia pero no son esenciales.
- Won't Have (No incluir ahora): No se implementarán en esta versión.

Requerimientos Clasificados

a. Must Have:

- i. Registro e inicio de sesión.
- ii. Catálogo de productos.
- iii. Carrito de compras.
- iv. Notificaciones automáticas de stock bajo por WhatsApp al administrador.
- v. Dashboard administrativo.
- vi. Historial de pedidos.

b. Should Have:

- i. Calculadora de costo de envío.
- ii. Notificaciones automáticas de pedidos por WhatsApp al cliente.
- iii. Sistema de reseñas.
- iv. Pasarela de pagos.

c. Could Have:

- i. Galería de inspiración.
- ii. Landing Page.
- iii. Resumen semanal por email con estado del inventario.

d. Won't Have

- i. Pie de página personalizado.

- ii. Banner destacado de productos.
- iii. Búsqueda de productos con filtro.
- iv. Reportes de ventas.

3. Estimación

Para el proceso de estimación se utilizará la metodología “Planning Poker” haciendo uso de la herramienta <https://planningpokeronline.com/>.

Se tendrá en cuenta que el proyecto se desarrollará utilizando frameworks modernos como React para el frontend y Node.js con Express para el backend, lo que implica un enfoque modular y eficiente en la creación de funcionalidades. Además, se empleará una base de datos relacional como PostgreSQL, para gestionar la información estructurada de productos, clientes y pedidos, garantizando integridad y escalabilidad en el manejo de datos.

Tras haber analizado los resultados obtenidos en el “Planning Poker” y haber comunicado las razones por las cuales habíamos dado cada una de las votaciones, este fue el consenso al que se llegó.

1. Registro e inicio de sesión:
 - a. **Tiempo:** 2 días.
 - b. **Complejidad:** Baja.
 - c. **Recursos disponibles:** 1 programador.
2. Catálogo de productos:
 - a. **Tiempo:** 3 días.
 - b. **Complejidad:** media.
 - c. **Recursos disponibles:** 2 programadores.
3. Carrito de compras:
 - a. **Tiempo:** 2 días.
 - b. **Complejidad:** baja.
 - c. **Recursos disponibles:** 2 programadores.

4. Notificaciones automáticas de stock bajo por WhatsApp al administrador:
 - a. **Tiempo:** 1 día.
 - b. **Complejidad:** baja.
 - c. **Recursos disponibles:** 1 programador, recursos monetarios (dinero) para el bot.
5. Dashboard administrativo:
 - a. **Tiempo:** 5 días.
 - b. **Complejidad:** media.
 - c. **Recursos disponibles:** 3 programadores.
6. Historial de pedidos:
 - a. **Tiempo:** 2 días.
 - b. **Complejidad:** baja.
 - c. **Recursos disponibles:** 1 programador.
7. Calculadora de costo de envío:
 - a. **Tiempo:** 1 día.
 - b. **Complejidad:** baja.
 - c. **Recursos disponibles:** 1 programador.
8. Notificaciones automáticas de pedidos por WhatsApp al cliente:
 - a. **Tiempo:** 1 día.
 - b. **Complejidad:** baja.
 - c. **Recursos disponibles:** 1 programador, recursos monetarios (dinero) para el bot.
9. Reportes de ventas:
 - a. **Tiempo:** 2 días.
 - b. **Complejidad:** media.
 - c. **Recursos disponibles:** 2 programadores.

10. Sistema de reseñas:

- a. **Tiempo:** 2 días.
- b. **Complejidad:** baja.
- c. **Recursos disponibles:** 2 programadores.

11. Pasarela de pagos:

- a. **Tiempo:** 3 días.
- b. **Complejidad:** media.
- c. **Recursos disponibles:** 2 programadores, una API como WOMPI que permite pagos tanto por PSE como otros medios de pago populares como Nequi.

12. Galería de inspiración:

- a. **Tiempo:** 1 día.
- b. **Complejidad:** baja.
- c. **Recursos disponibles:** 1 programador, librerías para subir imágenes como Multer o Busboy, ambas de Node.js.

13. Banner destacado de productos:

- a. **Tiempo:** 1 día (Si ya se cuentan con los banners).
- b. **Complejidad:** baja.
- c. **Recursos disponibles:** 1 diseñador gráfico para hacer banners llamativos, 1 programador para implementarlos, librerías como React para el carrusel.

14. Landing Page:

- a. **Tiempo:** 2 días.
- b. **Complejidad:** media.
- c. **Recursos disponibles:** 2 programadores, tal vez se pueda usar herramientas como "Yoast SEO" para analizar el SEO si el cliente quiere.

15. Resumen semanal por email con estado del inventario:

- a. **Tiempo:** 1 día.
- b. **Complejidad:** media.

- c. **Recursos disponibles:** 1 programador y herramientas como SendGrid o Mailgun para el envío automático.

16. Pie de página personalizado:

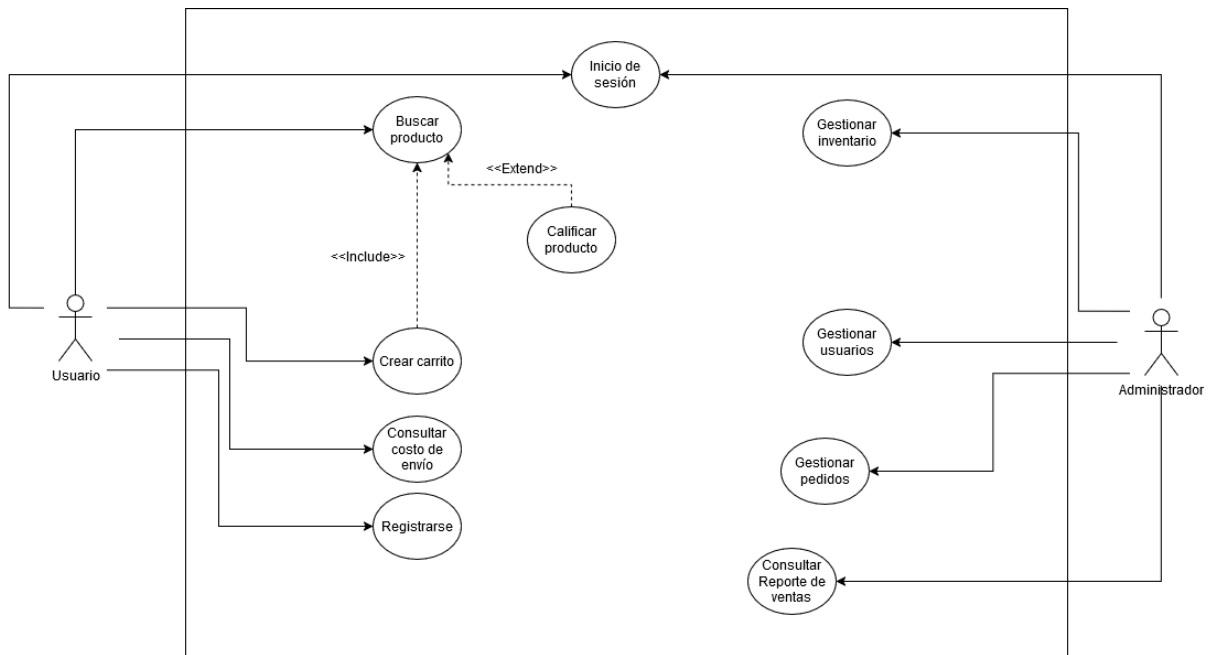
- a. **Tiempo:** 1 día.
- b. **Complejidad:** baja.
- c. **Recursos disponibles:** 1 diseñador gráfico, 1 programador y una herramienta como TailwindCSS que puede que deje personalizar.

17. Búsqueda de productos con filtro:

- a. **Tiempo:** 3 días.
- b. **Complejidad:** media.
- c. **Recursos disponibles:** 2 programadores y herramientas para realizar filtros dinámicos.

Cuarto Punto

Diagrama de Caso de Uso



Caso de Uso 1: Consultar costo de envío

Nombre:	Consultar costo de envío
Objetivo:	Ver cuanto es el estimado de costo de envío dependiendo de la ubicación del cliente
Actores:	Cliente
Flujo Básico:	<ol style="list-style-type: none"> La persona ingresa una dirección al sistema. <ol style="list-style-type: none"> Se ingresa por medio de google maps. Se verifica que la dirección sea válida. <ol style="list-style-type: none"> Dirección dentro de las posibilidades de envío. <ol style="list-style-type: none"> Si no es válida la dirección el sistema pide otra (regresar a paso 1). Se calcula un estimado del costo de envío. Se le muestra al usuario el costo de envío
Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	La información queda visible para ser consultada.

Caso de Uso 2: Creación de carritos

Nombre:	Creación de carritos
Objetivo:	Creación y pago de carritos de productos
Actores:	Cliente
Flujo Básico:	<ol style="list-style-type: none"> 1. Se crea una lista vacía de items a comprar. 2. Se muestra la lista actual. 3. El cliente selecciona añadir producto. 4. El cliente busca un producto. 5. Se añade el producto al carrito. <ol style="list-style-type: none"> a. Se selecciona la cantidad de productos. <ol style="list-style-type: none"> i. Si no hay disponibilidad no deja seleccionar. <p>Se repite este proceso hasta que el cliente desee (regresar a 2).</p> <ol style="list-style-type: none"> 6. Se le da al botón de siguiente. 7. Se muestra la lista de objetos a comprar y total. <ol style="list-style-type: none"> a. Se da la opción de eliminar ítems. 8. El cliente le da al botón de confirmar. <ol style="list-style-type: none"> a. El cliente puede dar al botón atrás (regresar a 2). 9. Se procede a la pasarela de pagos y se realiza el pago. <ol style="list-style-type: none"> a. Si falla volver un menú (regresar a 7). 10. Se añade el pedido a la base de datos.
Puntos de Extensión:	El producto se busca en el catálogo (4).
Precondiciones:	<ul style="list-style-type: none"> • El usuario tiene una conexión a internet. • El usuario ha iniciado sesión.
Postcondiciones:	Se añadió un pedido a la base de datos
Relaciones:	Incluye: Buscar producto

Caso de Uso 3: Buscar producto

Nombre:	Buscar producto
Objetivo:	Encontrar productos del catálogo
Actores:	Cliente
Flujo Básico:	<ol style="list-style-type: none"> 1. El cliente selecciona una categoría. 2. El sistema recupera los productos de dicha categoría. 3. El cliente mira la lista de productos ofrecidos. 4. El cliente selecciona un producto.
Puntos de Extensión:	Con el producto seleccionado puede proceder a calificarlo (3).

Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	Hay un producto seleccionado.
Relaciones:	Incluido en: Creación de carritos.

Caso de Uso 4: Calificar producto

Nombre:	Calificar producto
Objetivo:	Colocar una nota y reseña de un producto dado
Actores:	Cliente
Flujo Básico:	<ol style="list-style-type: none"> Se le pide dar una puntuación al producto. <ol style="list-style-type: none"> Puntuación seleccionable de 1 a 5 estrellas. <ol style="list-style-type: none"> Se pueden seleccionar medias estrellas. Si no se ingresa nada se asume 1. Se le solicita una reseña al usuario. <ol style="list-style-type: none"> Hasta 280 caracteres. Válido si no se ingresa nada. El usuario le da al botón de enviar. Se guarda la reseña en la base de datos. Se agradece al usuario por la reseña.
Precondiciones:	<ul style="list-style-type: none"> El usuario tiene una conexión a internet. Hay un producto seleccionado. El usuario ha iniciado sesión.
Postcondiciones:	Se guardó la reseña en la base de datos.

Caso de Uso 5: Iniciar sesión

Nombre:	Iniciar sesión
Objetivo:	Iniciar sesión en la plataforma
Actores:	Any
Flujo Básico:	<ol style="list-style-type: none"> Se piden datos de autenticación. <ol style="list-style-type: none"> Correo. Contraseña. El usuario le da al botón de ingresar. Se verifican los datos. <ol style="list-style-type: none"> Si no concuerdan los datos de correo y contraseña se vuelven a pedir los datos (regresar a paso 1). Se redirige a la página principal.

Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	El usuario inició sesión.

Caso de Uso 6: Gestionar inventario

Nombre:	Gestionar inventario
Objetivo:	Revisar, añadir o eliminar productos.
Actores:	Administrador
Flujo Básico:	<ol style="list-style-type: none"> 1. Se recupera una lista de los productos ofrecidos. 2. El usuario selecciona un producto. 3. Se muestra información del producto. <ol style="list-style-type: none"> a. Nombre. b. Precio (editable). c. Stock (editable). d. Puntaje de reseñas. 4. Se da clic en guardar información. 5. Se actualiza base de datos. 6. Se regresa a la lista (paso 1).
Flujo Alternativo:	<ol style="list-style-type: none"> 4a. Se da clic en eliminar producto. 5a. Se da confirmó si se quiere eliminar el producto. 6a. Se elimina el producto de los productos disponibles. 7a. Se regresa a la lista (paso 1).
Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	Se modificaron los productos ofrecidos.

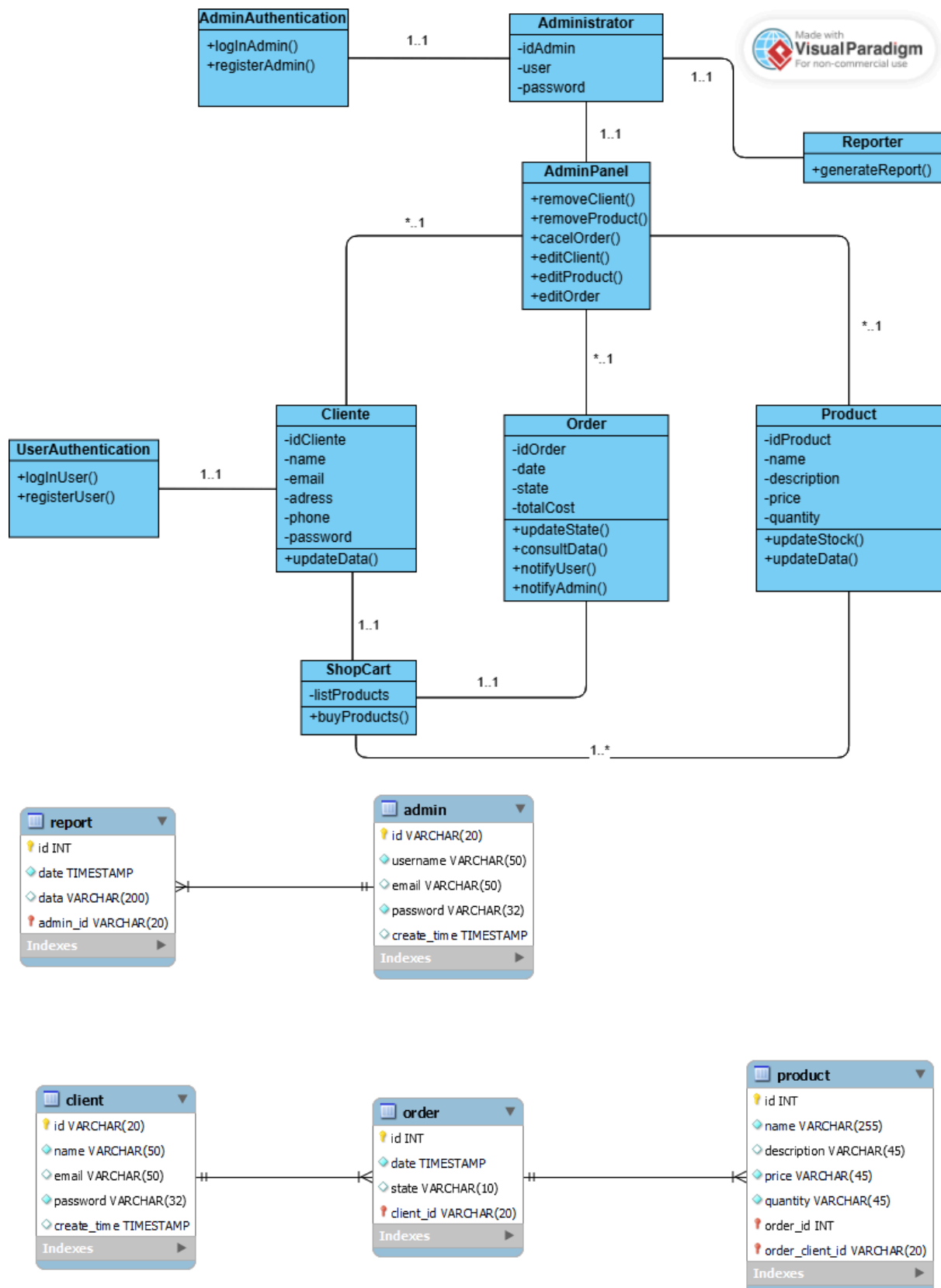
Caso de Uso 7: Gestionar usuarios

Nombre:	Gestionar usuarios
Objetivo:	Revisar o eliminar los usuarios registrados.
Actores:	Administrador
Flujo Básico:	<ol style="list-style-type: none"> 1. Se aplican filtros de búsqueda (opcional). <ol style="list-style-type: none"> a. Nombre. b. Dirección. c. Correo. d. Con pedidos pendientes. 2. Se recupera la lista de la base de datos. <ol style="list-style-type: none"> a. Si no hay filtros se recupera toda. 3. Se selecciona un usuario.

	<ol style="list-style-type: none"> 4. Se muestra la información disponible del usuario. <ol style="list-style-type: none"> a. Nombre. b. Correo. c. Dirección. d. Pedidos. e. Reseñas.
Flujo Alternativo:	<ol style="list-style-type: none"> 5. Se selecciona el botón de eliminar usuario. 6. Se borra la información del usuario de la base de datos. 7. Se regresa a la lista de usuarios (paso 2).
Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	Se muestra información del usuario

Caso de Uso 8: Consultar reporte de ventas

Nombre:	Consultar reporte de ventas
Objetivo:	Consultar información relevante de las ventas
Actores:	Administrador
Flujo Básico:	<ol style="list-style-type: none"> 1. Se selecciona un mes para ver la información. 2. Se recupera información de la base de datos para condensarla. 3. Se muestra la información relevante relativa a los pedidos. <ol style="list-style-type: none"> a. Ingresos del mes. b. Gastos del mes. c. Ganancia del mes. d. Gráfico de barras con productos más vendidos. e. Gráfico de barras de promedio de mejores puntajes de productos.
Precondiciones:	El usuario tiene una conexión a internet.
Postcondiciones:	Se muestra información financiera importante.



Quinto Punto

Historias de Usuario

Historia de Usuario #1

Anexo de Documentos Relacionados:

- No aplica.

Descripción conceptual

Módulo	<i>Módulo de Gestión de Usuarios</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>El sistema permitirá a los usuarios registrarse y acceder mediante un inicio de sesión seguro.</i>

Descripción técnica

Al recibir una solicitud de registro, el sistema valida los datos ingresados, asegurándose de que cumplan con las reglas establecidas (como formato de correo electrónico válido y contraseñas seguras). Los datos son cifrados y almacenados en la base de datos.

Para el inicio de sesión, el sistema verifica las credenciales proporcionadas comparándolas con los datos cifrados almacenados en la base de datos.

Backend

URL	Método	Código html
localhost:8080/api/usuarios	POST: Para el registro de usuarios. POST: Para el inicio de sesión.	201 Created: Registro exitoso. 400 Bad Request: Datos inválidos. 401 Unauthorized: Credenciales incorrectas en inicio de sesión.
Caso de uso técnico Al realizar una solicitud de registro, el sistema debe retornar el código 201 Created y en los datos (data) incluir un objeto con el usuario_id, el nombre y el correo electrónico del		

usuario recién creado, excluyendo la contraseña. En caso de credenciales incorrectas, debe retornar 401 Unauthorized con un mensaje de error adecuado.

Datos de entrada	Datos de salida
Registro de usuarios: <pre>{ "nombre": "María García", "correo": "maria.garcia@prueba.com", "contraseña": "Password1234UNAL" }</pre>	Registro de usuarios: 201: <pre>{ "status": "success", "data": { "usuario_id": 1, "mensaje": "Usuario registrado con éxito" } }</pre> 400: <pre>{ "status": "failed", "message": "El correo ya está en uso o datos inválidos." }</pre>

URL	Método	Código html
localhost:8080/users/1	PATCH	201 Created: Registro exitoso. 400 Bad Request: Datos inválidos.
Caso de uso técnico Al realizar una solicitud de inicio de sesión exitosa, el sistema debe retornar el código 200 OK y en los datos incluir un token de autenticación único (token). En caso de credenciales incorrectas, debe retornar 400 failed con un mensaje de error adecuado.		
Datos de entrada	Datos de salida	
Inicio de Sesión: 200: <pre>{ "correo": "maria.garcia@example.com",</pre>	Inicio de Sesión: 200: <pre>{ "status": "success", "data": {</pre>	

<pre>"contraseña": "Password1234" }</pre>	<pre>"token": "testeyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9" } }</pre> <p>400:</p> <pre>{ "status": "failed", "message": "Credenciales incorrectas." }</pre>
---	---

Frontend

Interacción Esperada:

- Formulario de Registro:
 - El usuario completa un formulario con nombre, correo y contraseña.
 - Al hacer clic en "Registrarse", el sistema valida los datos.
- Inicio de Sesión:
 - El usuario ingresa su correo y contraseña.
 - Al hacer clic en "Iniciar sesión", el sistema valida credenciales y otorga acceso.

Flujo Visual y Eventos:

- **Pantalla de Registro:**
 - Campos requeridos: Nombre, correo, contraseña.
 - Botón "Registrarse" habilitado tras validar datos.
- **Pantalla de Inicio de Sesión:**
 - Campos requeridos: Correo y contraseña.
 - Botón "Iniciar sesión" habilitado tras completar campos.
- **Confirmación:**

- Registro: Mensaje de éxito o error en la pantalla.
- Inicio de Sesión: Redirección al dashboard o mensaje de error.

Historia de Usuario #2

Anexo de Documentos Relacionados:

- No aplica.

Descripción conceptual

Módulo	Gestión de Pedidos
Descripción de la(s) funcionalidad(es) requerida(s):	El sistema debe permitir a los clientes registrar pedidos a través de un formulario en línea, ingresando información como datos del cliente, productos solicitados y cantidades.

Descripción técnica

Al recibir una solicitud de creación de pedido, el sistema valida los datos ingresados y los almacena en la base de datos.

Backend

URL	Método	Código html
localhost:8080/api/pedidos	POST	201 Created 400 Bad request
Caso de uso técnico Al hacer la petición, debe retornar código 201 y en data retorna un objeto con el id de pedido y el estado.		
Datos de entrada 201: <pre>{ "cliente": { "nombre": "Juan Pérez", "contacto": "juan.perez@example.com", "direccion": "Calle Falsa 123" }, "productos": [{"id": 1, "cantidad": 2}, {"id": 3, "cantidad": 1}] }</pre>	Datos de salida 201: <pre>{ "status": "success", "data": { "pedido_id": 101, "estado": "Pendiente" } }</pre>	

400: <pre>{ "cliente": { "nombre": "123_!!!", "contacto": "asdasdfsaf", "direccion": "1010101" }, "productos": [{"id": 1, "cantidad": 2}, {"id": 3, "cantidad": 1}] }</pre>	400: <pre>{ "status": "failed", "message": "Datos inválidos para los campos" }</pre>
---	--

Frontend

Interacción Esperada:

- El cliente selecciona productos y cantidades desde una lista.
- Completa un formulario con sus datos personales.
- Hace clic en el botón "Realizar Pedido".

Flujo Visual y Eventos:

- **Seleccionar productos:** El usuario elige productos y cantidades. Un contador de ítems se actualiza automáticamente.
- **Formulario de pedido:** El usuario ingresa sus datos y confirma el pedido.
- **Confirmación:** Aparece una pantalla de éxito con un resumen del pedido.

Historia de Usuario #3

Anexo de Documentos Relacionados:

- No aplica.

Descripción conceptual

Módulo	<i>Historial de Pedidos</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Registro de compras del usuario.</i>

Descripción técnica

URL	Método	Código html
localhost:8080/orders	GET	200
Caso de uso técnico Al consultar, debe retornar código 200 y en "data" un listado de los pedidos realizados por el usuario.		
Datos de entrada Parámetros opcionales para filtros: fechaInicio (string): Fecha de inicio del rango. fechaFin (string): Fecha de fin del rango. estado (string): Estado del pedido (ej. "Completado", "Pendiente").	Datos de salida <pre> { "status": "success", "data": [{ "id": 1, "fecha": "2024-01-01", "productos": [{ "nombre": "Producto A", "cantidad": 2, "precio": 50.00 }, { "nombre": "Producto B", "cantidad": 1, </pre>	

	<pre> "precio": 25.00 }], "total": 125.00, "estado": "Completado" }, { "id": 2, "fecha": "2024-01-10", "productos": [{ "nombre": "Producto C", "cantidad": 3, "precio": 30.00 }], "total": 90.00, "estado": "Pendiente" }] } </pre>
--	---

Frontend

Interacción esperada:

- El usuario debe acceder al historial de pedidos desde un enlace o botón en la página principal etiquetado como ver historial.
- Al ingresar, se muestra una tabla con los pedidos realizados, incluyendo columnas para fecha, total y estado.
- El usuario puede aplicar filtros para buscar pedidos específicos por rango de fechas, estado o categoría.
- Al hacer clic en un pedido, se despliega una nueva vista con los detalles completos del pedido.

Mockups/Prototipos:

- Página inicial con botón Ver Historial.
- Tabla que muestra los pedidos con opciones de filtro en la parte superior.
- Detalle del pedido mostrado página separada.

FinanceAp

Ver Historial

Fecha Inicio Fecha Fin Estado Categoría

Fecha	Total	Estado
2023-10-21	\$250.00	Completado
2023-10-19	\$120.00	Pendiente
2023-10-18	\$75.00	Cancelado

Order

Filter

Date

Total

Status

Search by Order Number



Order #2023-37-91 details

Order #2023-37-91

Created on 01/09/2023 at 11:30am

View

Email

Edit

Add Payment

Send

Order #2023-37-91

Issued on 01/09/2023

Due by 15/09/2023

From

Cool Company 83/A Main Road
3ND 75 +111 222 333 444

Issued to

Bread & Roll Ltd. 12 Main Road 8DI
92A +111 222 333 444

Type	Description	Quantity	Price
Service	Service Charges	1	\$1,250
Deposit		1	\$3,000
Items	Food Items	1	\$5,550
All payments in USD, tax fee applies.			
			Subtotal \$9,800
			Total \$9,800

Flujo visual y eventos:

- Al hacer clic en "Ver Historial": Mostrar un spinner mientras se cargan los datos.
- Al interactuar con los filtros: Actualizar los resultados de la tabla en tiempo real sin recargar la página.
- Al seleccionar un pedido: Mostrar un modal o una nueva página con el detalle completo.

Historia de Usuario #4

Descripción conceptual

Módulo	Módulo de Gestión de Reseñas
Descripción de la(s) funcionalidad(es) requerida(s):	Permitir que los usuarios califiquen productos con una puntuación de 1 a 5 estrellas, incluyendo la posibilidad de medias estrellas, y que añadan una reseña escrita de hasta 280 caracteres. La funcionalidad debe guardar las reseñas en la base de datos y confirmar al usuario que su reseña fue registrada.

Descripción técnica

Backend

URL	Método	Código html
localhost:8080/api/products/{productId}/reviews	POST	200: Solicitud exitosa, reseña guardada. 404: Producto no encontrado.
Caso de uso técnico Se debe tornar 200 si se guardó con éxito la reseña, en el caso de que el producto que se reseña no existe en la base de datos, se regresa 404.		
Datos de entrada <pre>{ "rating": 4.5, "review": "Excelente calidad, me encantó el diseño." }</pre>	Datos de salida 200: <pre>{ "status": "success", "message": "Reseña guardada exitosamente." }</pre> 404: <pre>{ "status": "error", "message": "El producto con ID {productId} no fue encontrado." }</pre>	

	}
--	---

Frontend

Interacción esperada:

- El usuario selecciona un producto y accede a la sección "Calificar Producto".
- Aparece un formulario con las siguientes opciones:
 - Selección de puntuación mediante estrellas (soporte para medias estrellas, mínimo 1 estrella por defecto).
 - Campo de texto para la reseña (opcional, hasta 280 caracteres).
 - Botón de envío.

Flujo visual y eventos:

1. Al seleccionar una puntuación, las estrellas cambian de color indicando la selección.
2. Si el usuario intenta enviar sin seleccionar una puntuación, aparece un mensaje: "Por favor, califica el producto" (la puntuación por defecto es 1 estrella).
3. Tras el envío exitoso, aparece un mensaje: "Gracias por tu reseña" y el formulario se vacía.

Notas Adicionales:

- La base de datos debe validar que el ID del producto exista antes de guardar la reseña.
- Integrar validaciones front-end para la longitud del texto y rango de puntuación.
- Diseñar un mensaje de error amigable en caso de fallo al enviar la reseña.