
ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (SRS)

para

SISTEMA DE RETAIL OMNICANAL

Versión 1.0

**Mauricio Alfonso Donado Canedo - T00064022,
Mauro José Prasca Bohorquez - T00065353,
Paula Andrea Márquez Orlando - T00064084**

11 de octubre de 2024

Índice general

1	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Acrónimos y Abreviaturas	4
1.4	Referencias	4
2	Descripción General	5
2.1	Características del Producto	5
2.2	Clases de Usuarios y Características	5
2.3	Limitaciones de Diseño y Aplicación	5
2.4	Documentación del Usuario	5
2.5	Supuestos y Dependencias	5
3	Características del Sistema	6
3.1	Descripción y Prioridad	6
3.2	Secuencias de Estímulo/Respuesta	6
3.3	Requisitos Funcionales	6
4	Requisitos de la Interfaz	7
4.1	Interfaces de Usuario	7
4.2	Interfaces de Hardware	7
4.3	Interfaces de Software	7
4.4	Interfaces de Comunicación	7
5	Otros Requisitos No Funcionales	8
5.1	Requisitos de Desempeño	8
5.2	Requisitos de Seguridad	8
5.3	Atributos de Calidad del Software	8
6	Otros Requerimientos	9
7	Requerimientos Adicionales del Sistema	10
7.1	API Unificada para Gestión de Clientes	10
7.2	Sincronización de Inventario en Tiempo Real	10
7.3	API de Pedidos Omnicanal	10
7.4	Generación de Reportes Consolidada	10

8	Requerimientos del Usuario	11
8.1	Experiencia de Usuario Consistente	11
8.2	Acceso Personalizado	11
8.3	Interfaz de Usuario Intuitiva	11
9	Requerimientos del Sistema	12
9.1	Rendimiento y Escalabilidad	12
9.2	Disponibilidad	12
9.3	Mantenimiento	12
10	Diagramas	13
11	Diagramas de Procesos	14
12	Diagrama de Clases	15
12.1	Clase User	15
12.2	Clase Administrator	15
12.3	Clase Customer	15
12.4	Clase Inventory	15
12.5	Clase Product	15
12.6	Clase Order	16
12.7	Clase Location	16
12.8	Clase Supplier	16
12.9	Diagrama de clase	17
12.10	Diagrama de Clases	17
12.11	Diagrama de Casos de Uso	17
12.12	Diagrama proceso de gestión de inventario	18
12.13	Diagrama gestión de análisis de datos	19
12.14	Diagrama proceso de gestión de catálogo	20
12.15	Diagrama proceso de gestión de pedidos	21
12.16	Diagrama máquinas de estados de catálogos/productos	22
12.17	Diagrama máquinas de estados de análisis de datos	24
12.18	Diagrama máquinas de estados de gestión de pedidos	25
12.19	Diagrama de secuencias de datos	26
12.20	Diagrama de secuencias de gestión de pedidos	26
12.21	Diagrama de secuencias gestión de inventario	27
13	Conclusión	28

1 Introducción

1.1. Propósito

Este documento especifica los requerimientos funcionales y no funcionales del sistema de integración omnicanal para una compañía de retail. Su objetivo es proporcionar una descripción detallada del sistema que permita la integración de varios canales (en línea, tiendas físicas) para ofrecer una experiencia de compra unificada y optimizada.

1.2. Alcance

El sistema propuesto permitirá a la compañía de retail gestionar de manera centralizada los datos de inventario, clientes y productos. Esto incluye la integración de sistemas heredados en una arquitectura basada en microservicios que expone APIs para diversos servicios de negocio, permitiendo una experiencia consistente en todos los puntos de contacto.

1.3. Definiciones, Acrónimos y Abreviaturas

- **API:** Interfaz de Programación de Aplicaciones (Application Programming Interface).
- **Omnicanal:** Experiencia de cliente que abarca múltiples canales de venta (en línea, en tienda, móvil).
- **SLA:** Acuerdo de Nivel de Servicio (Service Level Agreement).

1.4. Referencias

Este documento sigue las recomendaciones de la norma IEEE 29148 para la documentación de requerimientos de software.

2 Descripción General

2.1. Características del Producto

El sistema es una solución de software que consolida los datos de clientes, productos e inventarios, permitiendo una integración fluida entre múltiples sistemas heredados. La arquitectura se basará en microservicios, con un enfoque API-first, facilitando la interoperabilidad y escalabilidad.

2.2. Clases de Usuarios y Características

Los usuarios del sistema incluyen empleados del área de ventas, clientes que interactúan con las tiendas físicas y en línea, y administradores encargados de la gestión del inventario y los datos del cliente.

2.3. Limitaciones de Diseño y Aplicación

El sistema debe cumplir con normativas de protección de datos. Además, debe ser compatible con los sistemas heredados existentes, utilizando un enfoque gradual de migración.

2.4. Documentación del Usuario

La documentación debe ser fácil de usar e incluir guías detalladas para administradores y empleados de ventas, junto con guías rápidas para clientes finales.

2.5. Supuestos y Dependencias

El sistema debe suponer que todas las tiendas físicas ya están conectadas a internet y que los dispositivos móviles de los clientes tienen acceso constante a la red.

3 Características del Sistema

3.1. Descripción y Prioridad

El sistema priorizará la integración de sistemas de inventario en tiempo real y la gestión centralizada de pedidos.

3.2. Secuencias de Estímulo/Respuesta

Las APIs deben permitir respuestas en tiempo real, asegurando la sincronización inmediata de datos de inventario y pedidos entre todos los canales.

3.3. Requisitos Funcionales

El sistema debe ofrecer las siguientes funcionalidades clave:

- API unificada para la gestión de clientes y perfiles.
- Sincronización de inventarios en tiempo real.
- Gestión de pedidos omnicanal.
- Generación de reportes sobre ventas e inventario.

4 Requisitos de la Interfaz

4.1. Interfaces de Usuario

El sistema debe ofrecer una interfaz web y móvil para los clientes, y una consola de administración para los empleados.

4.2. Interfaces de Hardware

El sistema debe interactuar con terminales de puntos de venta (POS) en tiendas físicas y dispositivos móviles de los clientes.

4.3. Interfaces de Software

El sistema debe interoperar con los sistemas de inventario existentes y otros sistemas heredados.

4.4. Interfaces de Comunicación

El sistema debe utilizar protocolos seguros para la transmisión de datos, como HTTPS y OAuth 2.0 para la autenticación.

5 Otros Requisitos No Funcionales

5.1. Requisitos de Desempeño

El sistema debe garantizar un tiempo de respuesta inferior a 200 ms para las solicitudes de API comunes.

5.2. Requisitos de Seguridad

El acceso a las APIs debe estar autenticado mediante OAuth 2.0, y la información sensible debe ser encriptada tanto en tránsito como en reposo.

5.3. Atributos de Calidad del Software

El software debe ser escalable, permitiendo la integración de nuevos puntos de venta o servicios sin afectar la disponibilidad.

6 Otros Requerimientos

El sistema debe estar preparado para cumplir con normativas de protección de datos, y debe ser fácilmente adaptable a nuevas regulaciones locales.

7 Requerimientos Adicionales del Sistema

7.1. API Unificada para Gestión de Clientes

- El sistema debe ofrecer una API para gestionar la información de los clientes, incluyendo la creación, actualización y consulta de perfiles.
- La API debe consolidar datos de clientes desde sistemas heredados y proporcionar acceso en tiempo real a estos datos.

7.2. Sincronización de Inventario en Tiempo Real

- El sistema debe permitir la consulta de niveles de inventario en tiempo real en todas las tiendas y canales en línea.
- Las actualizaciones de inventario deben ser reflejadas instantáneamente a nivel global.

7.3. API de Pedidos Omnicanal

- El sistema debe gestionar pedidos realizados en línea y en tienda, permitiendo la recogida en tienda o el envío a domicilio.
- Las promociones y descuentos deben aplicarse de manera coherente a través de todos los canales.

7.4. Generación de Reportes Consolidada

- El sistema debe ofrecer herramientas para generar reportes sobre ventas, comportamiento de clientes e inventario.
- Los datos deben ser accesibles en tiempo real, permitiendo la toma de decisiones estratégicas.

8 Requerimientos del Usuario

8.1. Experiencia de Usuario Consistente

- Los usuarios deben experimentar un servicio consistente en todos los puntos de contacto (Sitio web, aplicación móvil).
- Las promociones y precios deben ser coherentes entre canales.

8.2. Acceso Personalizado

- Los usuarios deben poder acceder a su historial de compras, preferencias y recibir recomendaciones basadas en datos.
- Los usuarios deben poder administrar sus perfiles y preferencias desde cualquier dispositivo.

8.3. Interfaz de Usuario Intuitiva

- La interfaz debe ser fácil de usar y entender, permitiendo a los clientes encontrar productos rápidamente y realizar pedidos sin complicaciones.

9 Requerimientos del Sistema

9.1. Rendimiento y Escalabilidad

- El sistema debe manejar muchas solicitudes concurrentes sin degradación del rendimiento.
- Debe ser escalable horizontalmente para soportar el incremento de la demanda durante eventos o temporadas altas.

9.2. Disponibilidad

- El sistema debe tener una disponibilidad del 99.9% para cumplir con los SLA acordados.
- Se deben implementar estrategias de recuperación ante desastres.

9.3. Mantenimiento

- El sistema debe permitir el mantenimiento sin interrumpir la operación normal.
- Debe haber redundancia en todos los servicios críticos.

10 Diagramas

A continuación se presentan los diagramas:

11 Diagramas de Procesos

Interacción General

El diagrama muestra el flujo de interacción entre las capas Administrador, Frontend, API y Base de Datos para la gestión de productos dentro de un sistema de inventario. El administrador es el usuario que tiene acceso para realizar diversas acciones relacionadas con los productos.

Administrador

El administrador ingresa al sistema desde el dashboard y puede gestionar productos a través de varias acciones. Entre ellas, se incluye la posibilidad de *agregar productos, editar productos y eliminar productos.

Acciones Disponibles en el Frontend

- Agrupar por categoría: El administrador puede ver y seleccionar productos organizados por categorías.
- Seleccionar producto: Una vez que se selecciona un producto, el administrador puede realizar una serie de acciones, como aumentar o disminuir el stock.
- Acción inválida: En caso de que el administrador intente realizar una acción no permitida o incorrecta, el sistema generará un error y detendrá el proceso.

Gestión de Productos

- Agregar producto: Permite crear un nuevo producto en la base de datos.
- Editar producto: El administrador puede modificar la información existente de un producto.
- Eliminar producto: Opción para eliminar un producto del sistema.
- Aumentar/Disminuir stock: En caso de gestión de stock, el sistema verifica si la cantidad es mayor o menor que cero antes de actualizar los datos.

API y Base de Datos

Las acciones del administrador se comunican con la API, que a su vez interactúa con la Base de Datos para recuperar o actualizar la información de los productos. La base de datos responde con el producto solicitado o con un error si la operación no es válida.

12 Diagrama de Clases

12.1. Clase User

Esta clase abstracta define los atributos `userID`, `name` y `email`, junto con los métodos `login` y `logout`. Su propósito es servir como base para las clases hijas `Administrator` y `Customer`, que heredarán su funcionalidad.

12.2. Clase Administrator

Esta clase, que hereda de la clase `User`, define los atributos `AdminID` y `privileges`, así como los métodos `manageUser` y `updateSettings`. Su función es permitir la gestión de usuarios y la actualización de configuraciones específicas de la administración, facilitando así el control y la supervisión dentro del sistema.

12.3. Clase Customer

Esta clase, que hereda de la clase `User`, define los atributos `customerID`, `phone` y `preferences`, así como los métodos `createProfile`, `updateProfile` y `getProfile`. Su propósito es gestionar la información del cliente, permitiendo la creación, actualización y recuperación de su perfil dentro del sistema.

12.4. Clase Inventory

Esta clase define los atributos `inventoryID`, `productID`, `locationID` y `quantity`. Además, cuenta con los métodos `updateStock` y `checkStock`. El método `updateStock` permite modificar la cantidad de productos en el inventario, mientras que `checkStock` facilita la verificación de las existencias disponibles de un producto específico. El propósito de esta clase es gestionar el inventario de productos, asegurando un control eficiente sobre las cantidades y las ubicaciones de los artículos dentro del sistema.

12.5. Clase Product

Esta clase define los atributos `productID`, `name`, `description`, `price` e `inventoryStatus`. Además, incluye los métodos `getProductDetails`, `updateProduct` y `checkAvailability`. El método `getProductDetails` proporciona información completa sobre el producto, mientras que `updateProduct` permite modificar los detalles del producto, como el precio o

la descripción. Por último, `checkAvailability` verifica si el producto está disponible en el inventario. El propósito de esta clase es gestionar la información relacionada con los productos, asegurando un seguimiento eficaz de sus características y disponibilidad dentro del sistema.

12.6. Clase Order

Esta clase define los atributos `orderId`, `customerID`, `orderDate`, `status`, `totalAmount` y `items`, donde `items` es una lista que almacena los productos incluidos en el pedido. Además, incluye los métodos `placeOrder` y `updateOrderStatus`. El método `placeOrder` permite realizar un nuevo pedido, registrando todos los detalles necesarios, mientras que `updateOrderStatus` se encarga de modificar el estado del pedido, reflejando su progreso a lo largo del proceso de cumplimiento. El propósito de esta clase es gestionar el proceso de pedidos, asegurando un seguimiento adecuado de la información del cliente, el estado del pedido y el total de la compra dentro del sistema.”

12.7. Clase Location

Esta clase define los atributos `locationID`, `name`, `address` y `capacity`. También incluye los métodos `getLocationDetails` y `updateLocation`. El método `getLocationDetails` proporciona información detallada sobre la ubicación, incluyendo su nombre, dirección y capacidad, mientras que `updateLocation` permite modificar estos detalles según sea necesario. El propósito de esta clase es gestionar la información relacionada con las ubicaciones, asegurando un seguimiento preciso de sus características y facilitando la actualización de datos en el sistema.

12.8. Clase Supplier

Esta clase define los atributos `name`, `available`, `email`, `address`, `rating` y `products`, donde `products` es una lista de objetos de tipo `Product` que el proveedor ofrece. Además, incluye los métodos `provideProduct`, `getSupplierDetails` y `getSuppliedProducts`. El método `provideProduct(productID, quantity)` permite al proveedor suministrar una cantidad específica de un producto determinado, mientras que `getSupplierDetails` proporciona información completa sobre el proveedor, incluyendo su nombre, dirección y calificación. Por último, `getSuppliedProducts` devuelve una lista de los productos que el proveedor ofrece, estableciendo así una relación directa con la clase `Product`. El propósito de esta clase es gestionar la información de los proveedores y sus productos, facilitando la interacción entre el sistema y los proveedores y asegurando un flujo eficiente de suministros.

12.9. Diagrama de clase

12.10. Diagrama de Clases

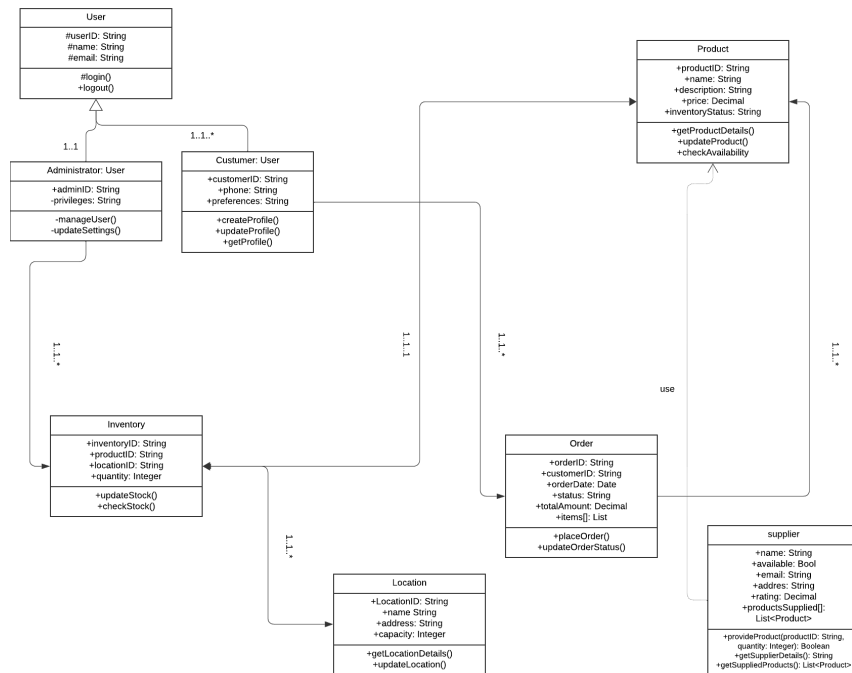


Figura 12.1: Diagrama de Clases del

12.11. Diagrama de Casos de Uso

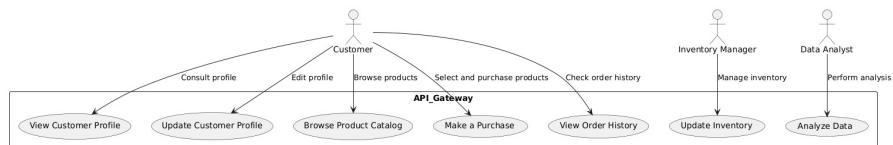


Figura 12.2: Diagrama de Casos de Uso

12.12. Diagrama proceso de gestión de inventario

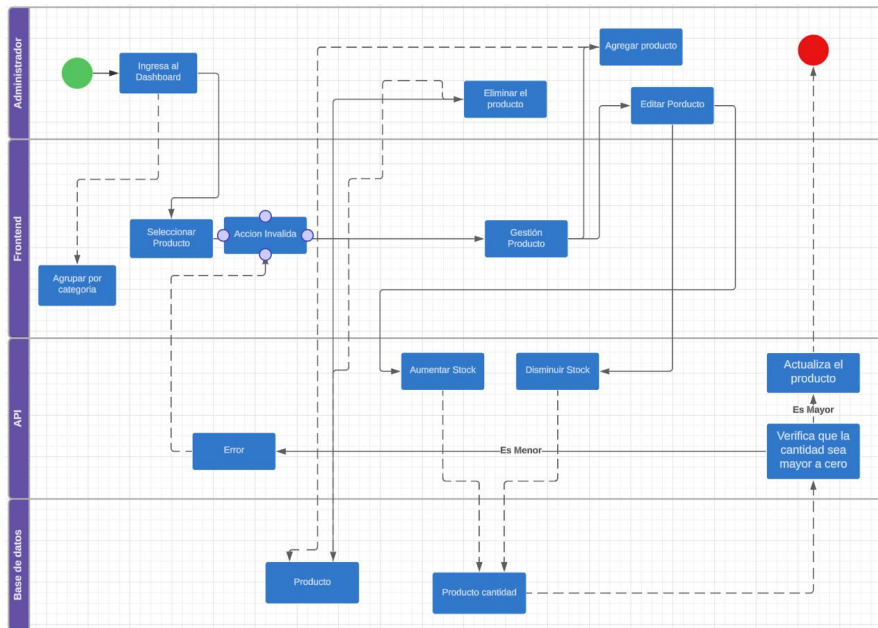


Figura 12.3: Diagrama de Proceso de Gestión de Inventario

12.13. Diagrama gestión de análisis de datos

Use Case ID:			
Use Case Name:	Order management		
Created By:	Mauricio Donado	Last Updated By:	Mauricio Donado
Date Created:	09/10/2024	Date Last Updated:	10/10/2024
Actors:	Data Analyst		
Description:	Data Analytics Management allows administrators or analysts to access and process the data generated by the system.		
Trigger:	An analyst or administrator requests the generation of a report or initiates a specific query from the analysis platform. An automated system can schedule periodic data analysis, such as a monthly sales report.		
Preconditions:	<ul style="list-style-type: none"> -The user must be authenticated with a role that allows access to the data analysis tool. -Relevant data must be available and up-to-date in the storage system. -Analysis tools and scripts must be properly configured and running. 		
Postconditions:	<ul style="list-style-type: none"> -The requested report or analysis has been successfully generated and can be accessed by the user. -The analysis results are stored for future reference or exported in the appropriate format. -Key insights derived from the analysis can be shared or presented to other users or teams. 		
Normal Flow:	<ul style="list-style-type: none"> -The user accesses the data analysis system from the administration panel. -Selects the type of analysis to be performed. -The user configures the analysis parameters. -The system processes the request, accessing the stored data and applying the necessary filters and calculations. -The system generates a report with relevant graphs and tables. -The user views the results and has the option to export or share them. 		
Alternative Flows:	<ul style="list-style-type: none"> -If the requested analysis requires additional data not available in the current system, the user can configure a new data source or request a database update. -The user can save the analysis parameters used to generate recurring reports automatically, instead of setting them manually each time. 		
Exceptions:	<ul style="list-style-type: none"> -Database failure: If the system cannot access the data, the system should notify the user and attempt a reconnection. -Incorrect query: If the user requests an analysis with parameters that do not generate valid results, the system should return an error or warning and allow the parameters to be corrected. -Excessive load: If the requested analysis generates too high a load on the servers, the system should optimize the process or notify the user of possible delays. 		
Priority:	High: Data analytics management is critical to strategic decision making. The company depends on this functionality to monitor its performance and adjust its operations effectively. Accuracy and reliability of analytics are critical.		

Figura 12.4: Diagrama de Gestión de Análisis de Datos

12.14. Diagrama proceso de gestión de catálogo

Use Case ID:			
Use Case Name:	catalog management		
Created By:	Mauricio Donado	Last Updated By:	Mauricio Donado
Date Created:	09/10/2024	Date Last Updated:	10/10/2024
Actors:	Customer, Inventory Manager		
Description:	allows the customer to review the product catalog and the inventory manager to update the product catalog		
Trigger:	A system administrator accesses the "Catalog Management" option from the system's main menu. Or an automated system requires updating the catalog at a specific time.		
Preconditions:	-The user must be logged in with a role that grants administrator or catalog manager permission. -The system must be connected to the database and the database must be available.		
Postconditions:	-The catalog has been correctly updated with new products, categories or descriptions. -Changes in the catalog should be reflected in the points of sale and be available to customers in real time.		
Normal Flow:	-The administrator accesses the "Catalog Management" section. -He/she selects the option to add, modify or delete a product or category. -The system displays a form where the administrator enters or modifies the data. -The administrator confirms the changes. -The system validates the data entered. -The system saves the changes in the database and displays confirmation of success.		
Alternative Flows:	-If the administrator selects a non-existing product or category, the system may offer the option to create a new one or revise the entry. -The administrator can cancel the operation at any time before confirming the changes, which would lead to a return to the main screen without making any changes.		
Exceptions:	-If the connection to the database fails during the operation, the system should notify the user with an error message and offer to retry the operation. -If the data entered is invalid (for example, an incorrect format for the price), the system should display an error message detailing the fields to be corrected.		
Priority:	High: It is crucial that the catalog management system runs smoothly as it directly impacts product visibility and customer experience on the sales platform.		

Figura 12.5: Diagrama de Proceso de Gestión de Catálogo

12.15. Diagrama proceso de gestión de pedidos

Use Case ID:			
Use Case Name:	Order management		
Created By:	Paula Marquez	Last Updated By:	Paula Marquez
Date Created:	09/10/2024	Date Last Updated:	10/10/2024
Actors:	Customer, Order System, Inventory Manager		
Description:	allows the user to view the order and place order history, the place order system and the inventory manager to update the order status		
Trigger:	A customer confirms the purchase on the platform. An administrator or order manager accesses the system to review or update an order.		
Preconditions:	The customer must be logged in or have provided the necessary information to complete the purchase. The selected products must be available in the catalog and in stock. The payment system must be active and available.		
Postconditions:	The order is successfully registered in the system and the customer receives order confirmation. The product inventory is updated to reflect the quantity on hand after the order is placed. The order status is set to "Processing" or "Confirmed" and is ready for shipment or preparation.		
Normal Flow:	<ul style="list-style-type: none"> -The customer selects the products and proceeds to the shopping cart. -The customer chooses a shipping option and selects the payment method. -The system calculates the total costs, including taxes and shipping. -The customer confirms the order. -The system verifies the availability of the products in stock. -The system processes the payment. -The system generates a unique order number and displays the order confirmation to the customer. -The system sends a notification to the customer with the order details and a summary of the purchase. 		
Alternative Flows:	<ul style="list-style-type: none"> -If the customer does not have sufficient credit or the payment method fails, the system offers the customer the option to retry with another payment method or correct the billing details. -If the customer decides to change the shipping address or products before confirming, he/she can edit the order at any time before finalizing it. 		
Exceptions:	<ul style="list-style-type: none"> -Failed payment: If the transaction cannot be completed due to problems with the payment gateway or the data provided, the system must notify the customer and indicate the steps to follow. -Out of stock: If any of the products are not available after order confirmation, the system should offer options such as removing the product from the order, notifying the customer of the delay, or canceling the entire order. -Connection error: If a connection failure or server error occurs during order confirmation, the system should keep the shopping cart active and allow retrying the operation without data loss. 		
Priority:	High: The order management process is essential to the business, as it directly impacts sales and customer satisfaction. It must operate without interruptions and with high reliability.		

Figura 12.6: Diagrama de Proceso de Gestión de Pedidos

12.16. Diagrama máquinas de estados de catálogos/productos

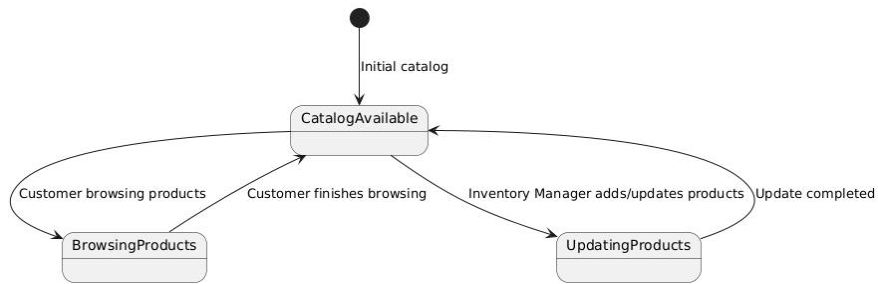
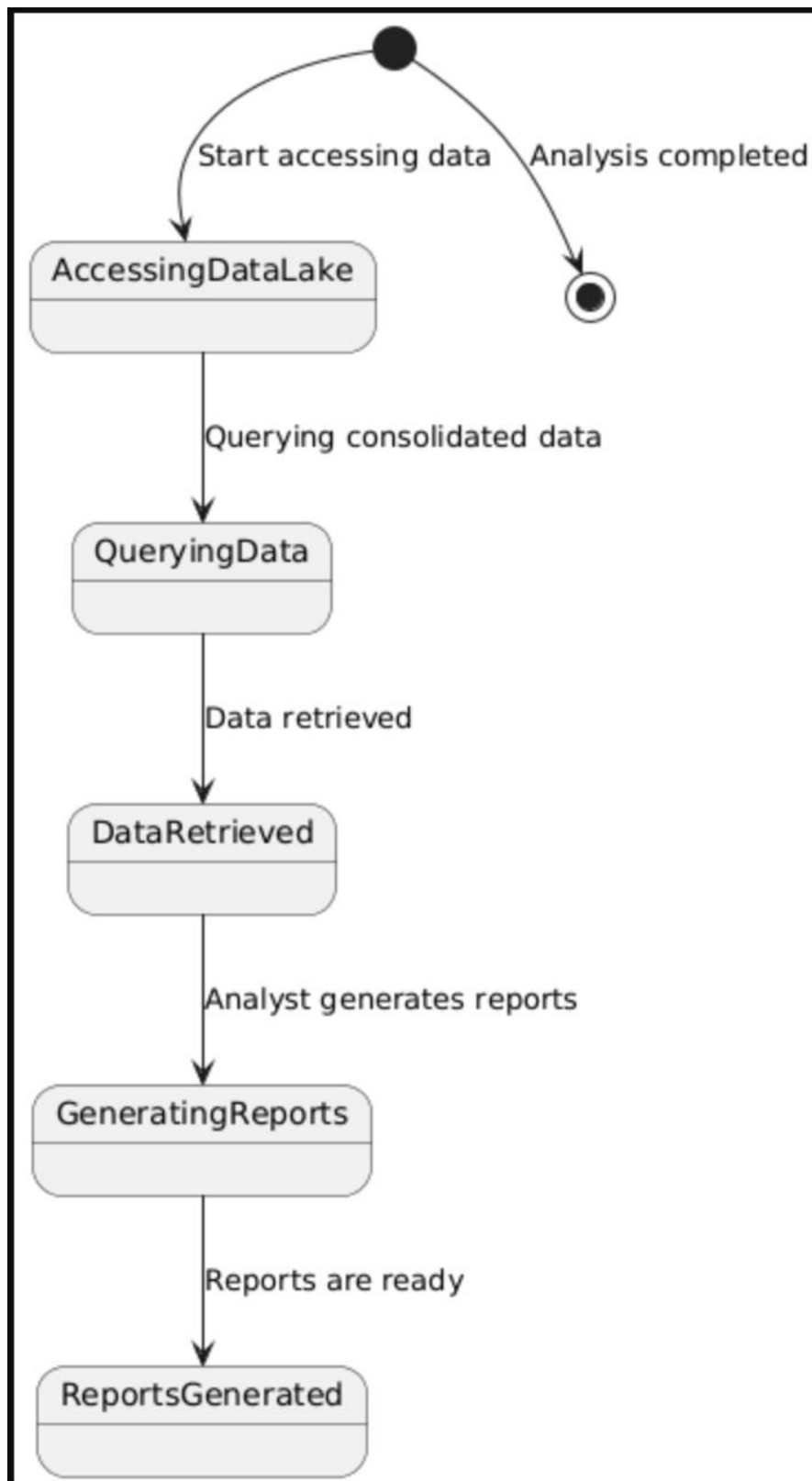


Figura 12.7: Diagrama de Máquina de Estados de Catálogos/Productos

12.17. Diagrama máquinas de estados de análisis de datos



12.18. Diagrama máquinas de estados de gestión de pedidos

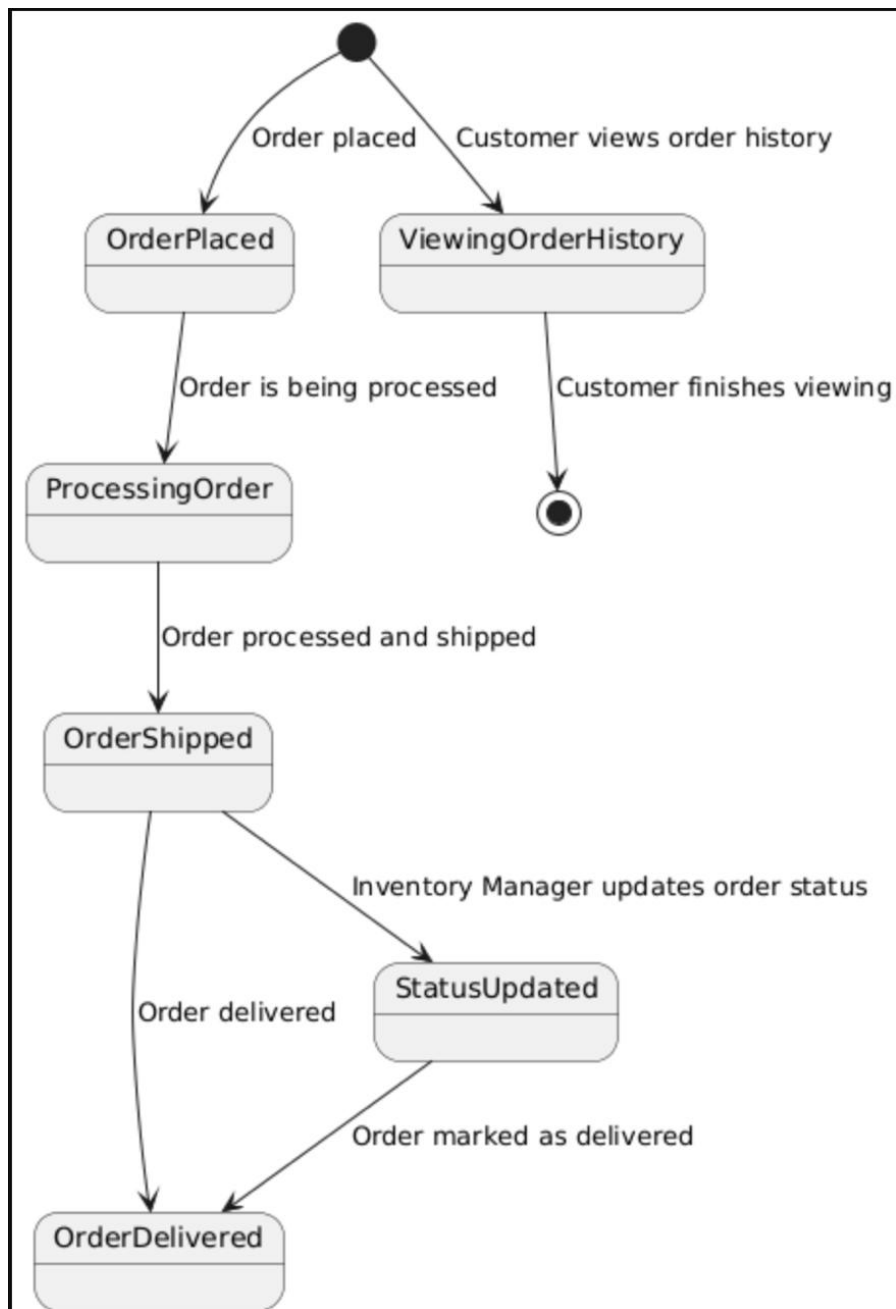


Figura 12.9: Diagrama de Máquina de Estados de Gestión de Pedidos

12.19. Diagrama de secuencias de datos

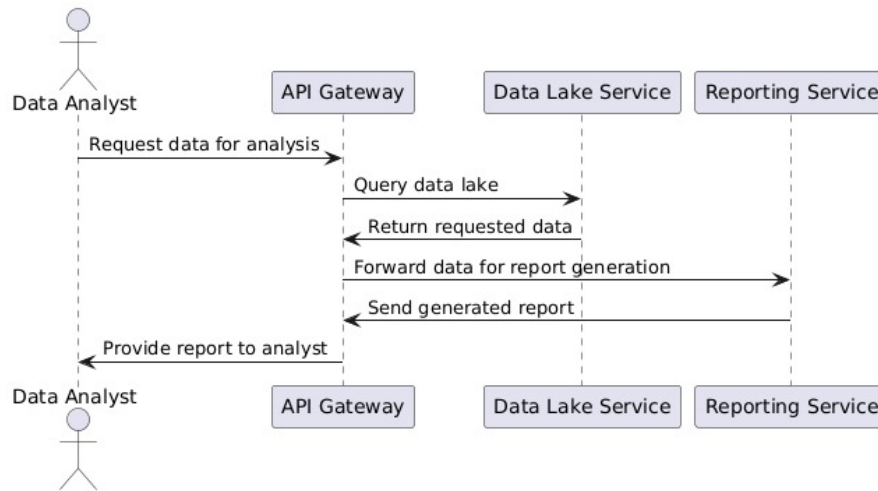


Figura 12.10: Diagrama de Secuencias de Datos

12.20. Diagrama de secuencias de gestión de pedidos

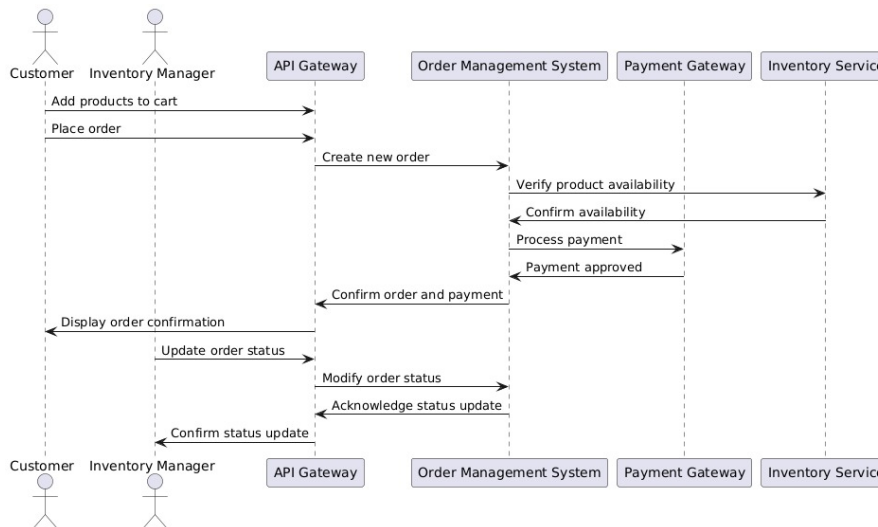


Figura 12.11: Diagrama de Secuencias de Gestión de Pedidos

12.21. Diagrama de secuencias gestión de inventario

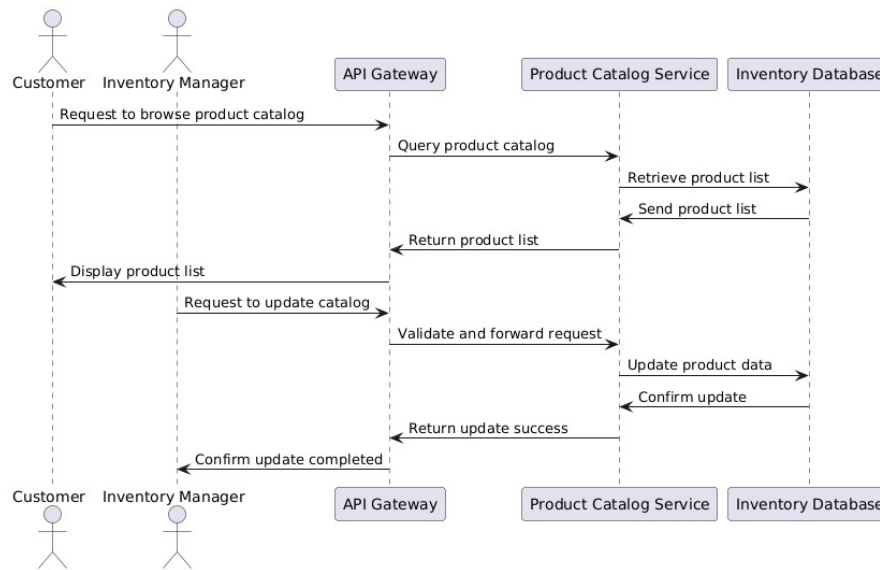


Figura 12.12: Diagrama de Secuencias de Gestión de Inventario del

13 Conclusión

El propuesto ofrece una solución integral para unificar la experiencia de clientes, integrar sistemas heredados y facilitar la toma de decisiones basada en datos en tiempo real.