

Arquitectura de software

Nombres de los participantes del proyecto

Sebastián Urrego Cárdenas y Duber Muñoz

1. Conceptos Fundamentales de la Arquitectura del Software

Definición:

El proyecto busca crear un sistema adaptable que optimice la gestión de inventarios para pequeñas y medianas empresas. Con un enfoque en la precisión, eficiencia y facilidad de uso, nuestro sistema proporcionará las herramientas necesarias para mejorar las operaciones diarias y facilitar la toma de decisiones informadas basadas en datos actualizados.

Calidad del Software:

La **escalabilidad** se debe optimizar para que el sistema pueda manejar un crecimiento en la cantidad de datos y usuarios sin comprometer su desempeño. La **mantenibilidad** del software se incrementa mediante un código claro, bien estructurado facilitando futuras actualizaciones y correcciones.

Comunicación Eficaz:

Hay que asegurar que tanto los dos desarrolladores del proyecto compartan una comprensión clara y coherente.

Toma de Decisiones Informadas:

Apoyar la adopción de decisiones de diseño fundamentadas y alineadas con los objetivos del proyecto.

Decisiones de Diseño Coherentes: Promover la coherencia y el conocimiento en las decisiones de diseño para asegurar que sean bien fundamentadas.

2. Principios y Patrones de Arquitectura

Principios de Diseño:

Modularidad :El sistema se dividirá en partes específicas para hacer más entendible su manejo para modificar los componentes con nuevas ideas y implementaciones que necesita el usuario.

-Encapsulamiento: Cada componente (Como botones) tendrán su propia estructura para así facilitar el mantenimiento fácil de entender ,modificar y ampliar.

-Cohesión: Tenemos partes específicas para guardar información importante del usuario es mucho más fácil entender cómo funciona el proyecto y hacerle cambios sin confundirte.

-Patrones de Arquitectura

Arquitectura de Microservicios por que puede ser beneficioso por varias razones, especialmente si el proyecto tiene el potencial de crecer o necesita ser muy flexible y su escalabilidad independiente puede crecer según la demanda de temporadas altas y podrá ser escalado sin necesidad de escalar todo el sistema, sin interferir unos con otros además el equipo de desarrollo puedes desplegar actualizaciones o mejoras para un microservicio sin afectar a los demás.Si el servicio que nosotros brindamos experimenta problemas , el resto del sistema, como la actualización de inventarios y la gestión de ventas, continuará funcionando normalmente, Puedes probar el microservicio que maneja las alertas de inventario sin tener que probar el sistema completo

4. importancia de los artefactos de Diseño .

Asegurar que el sistema sea altamente eficiente, fácil de comprender y mantener, y capaz de adaptarse a las futuras necesidades del cliente.

Organización Eficiente de la Información:

Importancia: Los artefactos de diseño, como los diagramas de base de datos y diagramas de clases, ayudan a organizar la estructura del inventario, mostrando cómo se relacionan los productos, categorías, proveedores y otros elementos. Esto asegura que toda la información esté bien estructurada y que las interacciones entre diferentes partes del sistema sean claras.

Facilita la Gestión y Actualización del Inventario:

Importancia: Con artefactos de diseño detallados, los desarrolladores pueden entender fácilmente cómo se gestionan las entradas y salidas de productos, los niveles de stock, y las alertas de reposición.

Documentación : Se Documentara toda la fase de diseño del software para que así pueda ser utilizada en diferentes o para futuros desarrolladores o un equipo de soporte para mantener el sistema de inventario (**StockEase**) funcionando correctamente para resolver futuras implementaciones.

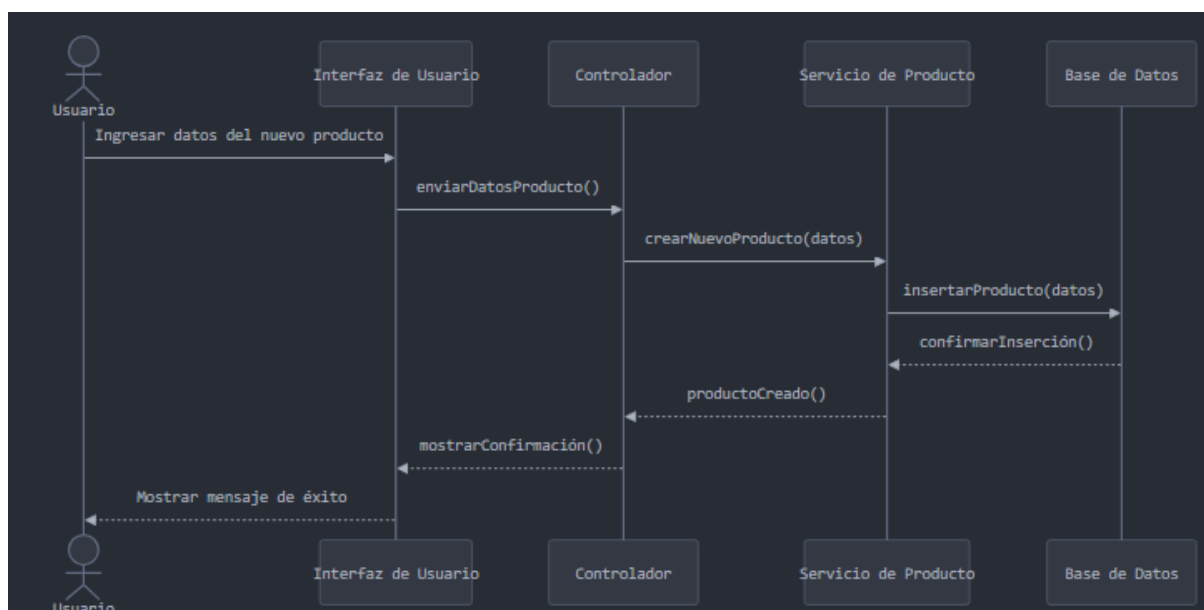
Mantenimiento y evolución: Si el sistema calcula incorrectamente los niveles de stock debido a un error en el código, el mantenimiento correctivo nuestro equipo se encargara de solucionar ese problema para que el inventario refleje los datos correctos asi asegurando queel sistema siga funcionando sin interrupciones.

Nos enfocaremos mucho en la interfaz de usuario para que así sea amigable a la vista y se ajuste a los usuarios finales mejorando su experiencia.

Toma de decisiones

-La toma de decisiones será crucial y analizada para asegurar un sistema eficiente, seguro y escalable que se adapte a las necesidades del usuario, desde la elección de las herramientas y tecnologías adecuadas para desarrollar el sistema de inventario,

Diagrama de Secuencia



El Usuario inicia el proceso ingresando los datos del nuevo producto en la Interfaz de Usuario (UI).

La UI envía estos datos al Controlador mediante el método `enviarDatosProducto()`.

El Controlador procesa la solicitud y llama al Servicio de Producto con el método `crearNuevoProducto(datos)`.

El Servicio de Producto se encarga de la lógica de negocio y prepara los datos para su almacenamiento. Luego, llama a la Base de Datos para insertar el nuevo producto usando `insertarProducto(datos)`.

La Base de Datos procesa la inserción y envía una confirmación de vuelta al Servicio de Producto.

El Servicio de Producto notifica al Controlador que el producto ha sido creado exitosamente.

El Controlador indica a la UI que muestre una confirmación.

Finalmente, la UI muestra un mensaje de éxito al Usuario.

Diagrama de componentes

Nota: Se utilizó el siguiente software para así mantener el proceso de datos de UML en movimiento : <https://www.plantuml.com>

-Este es el código con el cual pueden ver el diagrama de componentes:

@startuml

actor Vendedor

participant "Interfaz de Usuario" as UI

participant "Controlador de Ventas" as SC

participant "Controlador de Inventario" as IC

participant "Controlador de Productos" as PC

participant "Base de Datos" as DB

Vendedor -> UI: Iniciar venta

UI -> SC: `crearNuevaVenta()`

SC -> DB: `iniciarTransacción()`

loop Para cada producto

 Vendedor -> UI: Agregar producto a la venta

 UI -> SC: `agregarProducto(id, cantidad)`

```

SC -> PC: obtenerDetallesProducto(id)
PC -> DB: consultarProducto(id)
DB --> PC: detallesProducto
PC --> SC: detallesProducto
SC -> IC: verificarDisponibilidad(id, cantidad)
IC -> DB: consultarInventario(id)
DB --> IC: cantidadDisponible
IC --> SC: disponibilidad
alt Producto disponible
    SC -> IC: actualizarInventario(id, cantidad)
    IC -> DB: reducirInventario(id, cantidad)
    DB --> IC: confirmaci3nActualizaci3n
    IC --> SC: actualizaci3nExitosa
    SC -> UI: productoAgregado()
else Producto no disponible
    SC -> UI: errorProductoNoDisponible()
end
end

```

```

Vendedor -> UI: Finalizar venta
UI -> SC: finalizarVenta()
SC -> DB: guardarVenta()
DB --> SC: confirmaci3nVenta
SC -> DB: cometerTransacci3n()
SC -> UI: ventaCompletada()
UI --> Vendedor: Mostrar resumen de venta
@endum1

```

<http://hilite.me> utilizamos este software para imprimir esta informaci3n en formato de c3digo

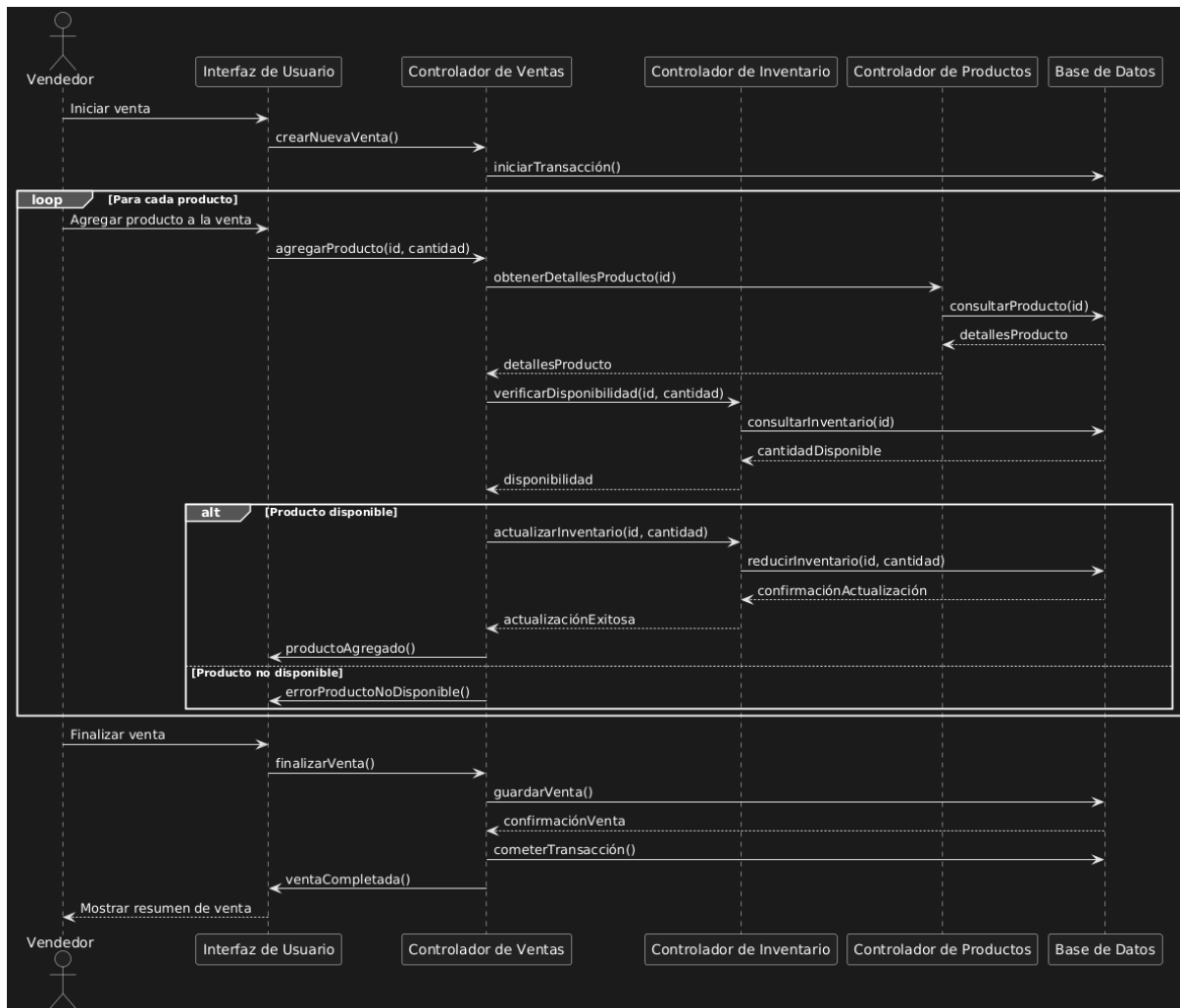


Diagrama de despliegue

Este diagrama de despliegue UML que representa la arquitectura física de un sistema de inventario.

Red Local:

Cliente PC: Computadoras de escritorio o laptops utilizadas por los empleados para acceder al sistema.

Cliente Móvil: Dispositivos móviles (tablets, smartphones) para acceso al sistema en movimiento.

Impresora de Recibos: Para imprimir recibos, etiquetas, etc.

Servidor de Aplicaciones:

Servidor Web: Maneja las solicitudes HTTP/HTTPS de los clientes y sirve la interfaz de usuario.

Servidor de Aplicación: Contiene la lógica de negocio del sistema de inventario.

Servidor de Base de Datos:

Base de Datos: Almacena todos los datos del sistema (productos, inventario, ventas, etc.).

Servicios en la Nube:

Servicio de Respaldo: Proporciona copias de seguridad de los datos del sistema.

Servicio de Reportes: Genera informes y análisis avanzados.

Las conexiones entre los componentes muestran cómo se comunican:

Los clientes (PC y móvil) se conectan al servidor web a través de HTTP/HTTPS.

El servidor web se comunica con el servidor de aplicación.

El servidor de aplicación interactúa con la base de datos y los servicios en la nube.

El servidor de aplicación también se comunica con la impresora de recibos para generar documentos físicos.

