

Evidencia GA6-220501123-AA1-EV01

Informe Arquitectura y Tecnologías

Aprendiz: Juan Carlos Lopez Moreno

Instructor: Alvaro Esteban Betancourt Matoma

Área Técnica

Servicio Nacional de Aprendizaje – SENA

Técnico en Programación de Aplicaciones y Servicios Para la Nube

Código: 3186263

Diciembre 2025

1. Introducción

El presente informe técnico tiene como objetivo describir y justificar la arquitectura de software y las tecnologías seleccionadas para el desarrollo del "Sistema de Gestión y Venta de Café Orgánico - Finca Miraflores". Este proyecto busca digitalizar la presencia comercial de la finca mediante un escaparate digital, una tienda en línea y un sistema de gestión administrativa. La selección tecnológica se fundamenta en criterios de escalabilidad, rendimiento, SEO y mantenibilidad, alineándose con las tendencias modernas del desarrollo web. A través de este documento, se analizan los conceptos teóricos fundamentales como la arquitectura Cliente-Servidor y la Programación Orientada a Objetos, para luego detallar cómo se implementan específicamente mediante el uso de Next.js, TypeScript y Supabase en la solución propuesta.

2. Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. En el contexto del desarrollo web moderno, esta arquitectura ha evolucionado significativamente. El cliente, típicamente un navegador web, se encarga de la presentación y la interacción con el usuario, mientras que el servidor gestiona la lógica de negocio, el acceso a bases de datos y la seguridad. Para el proyecto de Finca Miraflores, esta arquitectura se implementa de manera híbrida gracias al framework Next.js. Este permite que ciertas partes de la aplicación se rendericen en el servidor (Server-Side Rendering) para entregar contenido optimizado y rápido al cliente, mientras que otras interacciones ocurren directamente en el navegador, ofreciendo una experiencia de usuario fluida y dinámica.

3. Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios (SOA) es un estilo de diseño de software donde los servicios se proporcionan a los otros componentes a través de un protocolo de comunicación a través de una red. Aunque el proyecto Finca Miraflores se define inicialmente como un "monolito modular" para facilitar el desarrollo por un solo programador, incorpora principios de SOA mediante el uso de servicios externos (Software as a Service). Por ejemplo, la autenticación, la base de datos y el almacenamiento de archivos son gestionados por Supabase como servicios independientes, y el procesamiento de pagos se delega a Stripe. Esta aproximación permite que el sistema base se centre en la lógica de negocio específica de la venta de café, mientras consume servicios robustos y especializados para funciones críticas de infraestructura.

4. Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es el lenguaje estándar para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Su importancia radica en ofrecer una forma estandarizada de escribir los "planos" de un sistema, incluyendo conceptos conceptuales como procesos de negocio y funciones del sistema, así como conceptos concretos como clases escritas en un lenguaje específico, esquemas de bases de datos y componentes de software reutilizables. En el desarrollo de este proyecto, UML es fundamental para diagramar la estructura de las clases del dominio, los flujos de interacción del usuario en el proceso de compra y la arquitectura de despliegue, asegurando que la lógica del negocio esté claramente definida antes de la implementación del código.

5. Programación Orientada a Objetos (POO)

La Programación Orientada a Objetos (POO) es un paradigma de programación que utiliza "objetos" para diseñar aplicaciones y programas informáticos. Se basa en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. En el contexto del proyecto Finca Miraflores, la POO se manifiesta en la estructura del código TypeScript. Las entidades del negocio, como "Producto", "Pedido" o "Usuario", se modelan como interfaces y tipos que definen sus propiedades y comportamientos. Los servicios del sistema (ProductsService, OrdersService) actúan como clases que encapsulan la lógica necesaria para manipular estos objetos, promoviendo un código organizado, reutilizable y fácil de mantener.

6. Modelo Entidad-Relación

El modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Este modelo es la base para el diseño de la base de datos relacional del proyecto. En el sistema de Finca Miraflores, se han identificado entidades clave como "Perfiles" (usuarios extendidos), "Productos" (inventario de café), "Pedidos" (transacciones) e "Ítems de Pedido" (detalles de compra). La correcta definición de las relaciones entre estas tablas, como la relación uno a muchos entre usuarios y pedidos, o muchos a muchos entre pedidos y productos, es crucial para garantizar la integridad referencial y la eficiencia de las consultas en la base de datos PostgreSQL utilizada.

7. Tecnologías Seleccionadas para el Proyecto

A continuación, se describen las tecnologías específicas seleccionadas para la construcción del software, reemplazando las opciones genéricas por las herramientas modernas definidas en la arquitectura del proyecto.

7.1 ¿Qué es TypeScript? (Lenguaje de Programación)

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto estricto sintáctico de JavaScript y añade tipado estático opcional al lenguaje. Para el proyecto Finca Miraflores, se ha seleccionado TypeScript en lugar de JavaScript o Java debido a su capacidad para detectar errores en tiempo de compilación, lo que aumenta significativamente la robustez del código. El tipado estático permite definir "contratos" claros para las estructuras de datos (como la forma exacta de un objeto "Café"), mejorando la experiencia de desarrollo con autocompletado inteligente y facilitando el mantenimiento a largo plazo del sistema de comercio electrónico.

7.2 ¿Qué es PostgreSQL? (Base de Datos)

PostgreSQL es un sistema de gestión de bases de datos relacional de objetos potente y de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que le ha ganado una sólida reputación de fiabilidad, integridad de datos y corrección. En este proyecto, PostgreSQL actúa como el motor de persistencia principal, alojado dentro de la plataforma Supabase. Se eligió sobre otras opciones como MySQL debido a su estricto cumplimiento de los estándares SQL, su soporte avanzado para tipos de datos complejos (como JSONB) y su robustez para manejar transacciones financieras críticas, asegurando que ningún pedido se pierda o se registre incorrectamente.

7.3 ¿Qué es Supabase? (Plataforma de Backend)

Supabase es una alternativa de código abierto a Google Firebase que proporciona todas las herramientas de backend necesarias para construir un producto. En lugar de utilizar una herramienta de administración tradicional como MySQL Workbench, el proyecto utiliza el panel de control (Dashboard) de Supabase para gestionar la base de datos PostgreSQL. Supabase no solo ofrece una interfaz gráfica para administrar tablas y relaciones, sino que también integra autenticación de usuarios, almacenamiento de archivos (para las imágenes de los productos) y APIs generadas automáticamente. Esto reduce drásticamente el tiempo de desarrollo al eliminar la necesidad de construir y mantener un servidor backend tradicional desde cero.

7.4 ¿Qué es Visual Studio Code? (Entorno de Desarrollo)

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Para el desarrollo del sistema Finca Miraflores, VS Code es la herramienta fundamental debido a su excelente integración con TypeScript y el ecosistema de React. Gracias a sus extensiones, permite visualizar errores en tiempo real, formatear el código automáticamente y gestionar el control de versiones sin salir del editor, optimizando el flujo de trabajo del desarrollador.

7.5 ¿Qué es Git? (Control de Versiones)

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. En este proyecto, Git es esencial para llevar un

registro histórico de todos los cambios realizados en el código fuente. Permite crear "ramas" para trabajar en nuevas características (como el carrito de compras o la integración de pagos) sin afectar la versión principal estable del software, facilitando la experimentación y la corrección de errores de manera segura.

7.6 ¿Qué es GitHub? (Repositorio Remoto)

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Es propiedad de Microsoft y ofrece funcionalidad de alojamiento de código, control de versiones y gestión de incidencias. Para Finca Miraflores, GitHub sirve como el repositorio central en la nube donde se almacena el código fuente del proyecto. Además de ser una copia de seguridad segura, se integra directamente con la plataforma de despliegue Vercel, permitiendo que cada vez que se actualiza el código en GitHub, la página web pública se actualice automáticamente mediante un proceso de Integración Continua y Despliegue Continuo (CI/CD).

7.7 ¿Qué es Node.js? (Entorno de Ejecución)

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Permite a los desarrolladores ejecutar JavaScript en el servidor. Aunque el proyecto utiliza Next.js (un framework), Node.js es la tecnología subyacente que permite que las herramientas de desarrollo funcionen. Se utiliza para gestionar las dependencias del proyecto (librerías externas como React, Stripe o Tailwind) a través de npm y para ejecutar el servidor local durante la etapa de desarrollo. Además, Node.js potencia las "Server Actions" de Next.js, permitiendo ejecutar lógica segura del lado del servidor, como el procesamiento de pagos secretos, utilizando el mismo lenguaje que en el frontend.

8. Conclusiones

La selección de la arquitectura y tecnologías para el sistema "Finca Miraflores" responde a la necesidad de crear una plataforma moderna, segura y escalable con recursos de desarrollo optimizados. La implementación de una arquitectura basada en Next.js y el patrón de Monolito Modular permite un equilibrio ideal entre rendimiento (gracias al renderizado del lado del servidor) y facilidad de mantenimiento. El uso de TypeScript garantiza un código robusto y menos propenso a errores, vital para un sistema transaccional.

Asimismo, la decisión de utilizar Supabase sobre una infraestructura tradicional de servidor reduce la complejidad operativa, delegando la gestión de la base de datos PostgreSQL y la seguridad a expertos, permitiendo al desarrollador centrarse en la lógica de negocio y la experiencia de usuario. En conjunto, este stack tecnológico no solo cumple con los requerimientos funcionales de venta y gestión, sino que asegura que la aplicación sea sostenible y evolutiva en el tiempo, proporcionando una base sólida para el crecimiento digital de la Finca Miraflores.