**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**“Universidad del Perú. Decana de América”**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**E.P. INGENIERÍA DE SOFTWARE**



**“Documento de la Arquitectura de Software”**

**Empresa: AETHER TECH**

**Grupo 2**

**Docente:**

Wong Portillo, Lenis Rossi

**Integrantes:**

* Davalos Benito Rodrigo
* Limachi Sarmiento, Jose Luis
* Meza Torres, Joseph Omar
* Panta Labán, Leonardo Justo
* Uscamayta Sanchez, Gabriel Omar
* Villanueva Aguirre, Cesar Alexander

**2025 - I**

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| **10/5/25** | **1.0** | **índice** | **Davalos** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[**Introducción 4**](#_q66z52o1xr9h)

[Propósito 4](#_47tm89pvlxj7)

[Alcance 4](#_d29fwopy7cmw)

[Definicion, siglas y abreviaturas 4](#_bj3vm5qro60a)

[Vista Global 4](#_h5zcwqdz2hq6)

[**Macro Arquitectura 4**](#_jlnzjb5wajhh)

[Modelo Multicapa 4](#_1cel6w5azoh)

[**Vista Física 5**](#_tr0nzt7aqjyb)

[**Vista funcional o lógica 5**](#_w5pu8gwptpin)

[**Vista de Despliegue 5**](#_danksmb8a20j)

[**Vista de procesos 5**](#_h2p6ap8p48ck)

# 

# Introducción

## Propósito

El propósito de este documento es definir y dar a conocer la Arquitectura de Software para el desarrollo del proyecto "Sistema Gestor de Almacenes para tiendas Cencosud (SGATC)”. Se utilizarán las vistas de proceso, lógica y física, para describir la estructura y funcionalidad del sistema, así como su distribución física, interacción y descripción general.

Este documento está dirigido al equipo de desarrollo, con el fin de proporcionar las bases arquitecturales y una visión global del diseño para llevar a cabo el proyecto.

## Alcance

Este documento describe el alcance del sistema "Gestión de Almacenes para Cencosud", el cual será implementado para mejorar los procesos logísticos y operativos de los supermercados Wong y Metro. El sistema permitirá realizar un control eficiente del inventario, automatizar los pedidos de reabastecimiento y optimizar la distribución de productos en los almacenes. El sistema será accesible a través de una plataforma web, diseñada para ser utilizada en dispositivos con navegación estándar por los operadores logísticos, encargados de almacén y personal administrativo.

## Definicion, siglas y abreviaturas

* DAS: Documento que describe la arquitectura de software.
* SGATC: Sistema gestor de Almacenes de Tiendas Cencosud
* API: Interfaz que permite a diferentes aplicaciones comunicarse entre sí.
* HTTP: Protocolo utilizado para transferir datos en la web.
* IIS (Internet Information Services): Servidor web utilizado para alojar aplicaciones y sitios web.
* DBMS: Sistema que administra bases de datos.
* SQL (Structured Query Language): Lenguaje utilizado para interactuar con bases de datos.
* GUI (Graphical User Interface): Interfaz visual que permite interactuar con un programa.
* React: Biblioteca de JavaScript para crear single page applications.
* Responsive Design: Diseño que se adapta a diferentes dispositivos y tamaños de pantalla.
* Singleton: Patrón de diseño que garantiza una única instancia de una clase.
* Modelo Multicapa (Modelo n-tier): Patrón de arquitectura de software que separa la lógica de negocio de la interfaz del usuario en capas.

## Vista Global

Se presentará una macro arquitectura del sistema "SGATC", con especificaciones de las vistas utilizadas en el proyecto, metas y restricciones arquitectónicas, así como vistas con diagramas para una comprensión completa del sistema.

# Macro Arquitectura

## Modelo Multicapa

El modelo multicapa, organiza el sistema en componentes diferenciados para separar la lógica de negocio de la interfaz de usuario. Esta estructura se compone de:

* Capa de Presentación/UI: Contiene la interfaz de usuario y los componentes del Front-End.
* Capa de Lógica de Negocio: Gestiona procesos, reglas de negocio, controladores, middlewares y el enrutamiento.
* Capa de Acceso a Datos: Se encarga de la gestión de datos, la interacción con bases de datos, ORM y APIs.

La principal ventaja de esta arquitectura es su separación, tanto lógica como física, de las funcionalidades. Cada capa puede operar en sistemas independientes, con su propio sistema operativo y plataforma de servidor (como servidores web, de aplicaciones o de bases de datos), elegidos según sus necesidades específicas. Además, al ejecutarse en hardware o servidores virtuales dedicados, cada nivel puede optimizarse y personalizarse sin afectar a los demás. Esto mejora la mantenibilidad, escalabilidad y flexibilidad del sistema, ya que cada capa tiene una función definida, facilitando la integración de nuevas funcionalidades y futuras adaptaciones.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Atributos de calidad “Observables”** | | | | | | | |
| Atributo | Descripción | | | Tácticas /Patrón de  Arquitectura | | Aplicación | |
| Seguridad | Garantizar que el acceso a los recursos y funcionalidades esté restringido según los roles asignados a los usuarios autorizados. | | | Control de Acceso Basado en Roles: Asignación de permisos concretos al usuario para poder comprar o vender productos. | | Aplicación al momento de autenticación de usuarios y accesos a funcionalidades específicas. | |
| Disponibilidad | Capacidad del sistema para estar operativo y accesible para los usuarios en todo momento. | | | Balanceo de Carga: Distribución equitativa de la carga de trabajo entre varios servidores para evitar sobrecargas. | | Implementación en la infraestructura de servidores de SGATC. | |
| Usabilidad | Facilidad de uso del sistema para los usuarios finales. | | | Diseño Centrado en el Usuario (UCD): Integración de procesos de diseño y pruebas centrados en la experiencia del usuario. | | Desarrollo de interfaces intuitivas, para el manejo de operaciones de compra y venta. | |
| **Atributos de calidad “No observables”** | | | | | | | |
| Atributo | | | Descripción | Tácticas /Patrón de  Arquitectura | | Aplicación | |
| Modificabilidad | | | Facilidad para realizar cambios en el sistema, como agregar nuevas funcionalidades o adaptarse a requisitos cambiantes. | Arquitectura Multicapa: Descomposición del sistema en servicios independientes y autocontenidos. | | División de la lógica de negocios, gestión de recursos y acceso a datos en microservicios. | |
| Mantenibilidad | | | Capacidad de realizar mantenimiento y mejoras al sistema de manera eficiente y económica. | Separación de capas: División clara de las capas de presentación, lógica de negocios y acceso a datos para facilitar las actualizaciones. | | Implementación de patrones de diseño y estándares de codificación que favorezcan la legibilidad, cohesión y reutilización del código. | |
| Portabilidad | | | Adaptabilidad del sistema para funcionar en diferentes entornos. | Diseño Responsivo: Desarrollo de interfaces de usuario que se ajustan automáticamente. | | Implementación de técnicas de diseño web responsivo para una experiencia de usuario consistente y accesible. | |

# Vista Física

Descripción:

El sistema permite a los usuarios acceder aSGATC desde dispositivos como computadoras, laptops o móviles, interactuando con una página web alojada en un servidor remoto.

Elementos físicos:

Usuarios:

Personas que interactúan con la plataforma desde cualquier dispositivo con navegador.

Dispositivos electrónicos:

Equipos con conexión a internet: PCs, laptops, tablets o smartphones.

Servidor de aplicaciones:

Aloja la página web de EzCommerce (frontend en React y backend en Node.js).

Gestionado por un proveedor externo para garantizar disponibilidad y seguridad.

Base de datos:

Almacena toda la información del sistema (productos, clientes, transacciones).

Usa MySQL como gestor de bases de datos relacional.

Diagrama

El contenido generado por IA puede ser incorrecto.

# Vista funcional o lógica

Descripción:

Configuración de red física donde se despliega el sistema, mostrando los nodos clave y su interacción.

Componentes:

Nodo de Acceso a la Aplicación Web:

Descripción: Dispositivos del usuario (PC, laptop, móvil) con navegador web.

Protocolo: HTTP/2 (peticiones seguras).

Interacción: Envía solicitudes al servidor web (frontend en React).

Nodo del Servidor Web:

Descripción: Servidor que aloja:

Frontend: Aplicación React (interfaz de usuario).

Backend: API en Node.js (lógica de negocio, autenticación).

Funcionalidad: Separa capas (presentación, lógica) para mejor mantenibilidad.

Nodo del Servidor de Base de Datos:

Descripción: Servidor dedicado que gestiona datos con MySQL (DBMS).

Protocolo: Conexión segura al backend mediante SSH o TLS.

Funcionalidad: Almacena y administra datos (ej: productos,acciones).

Lógica de Negocio y Backend:

Descripción: APIs en Node.js que intermedián entre el frontend y la base de datos.

Interacción:

Recibe peticiones HTTP del frontend.

Ejecuta operaciones en MySQL mediante consultas

# Vista de Despliegue