Documento Arquitectura de Software

Plataforma web

“La Picá de la Chabelita”

**Identificación de Documento**

| **Identificación** | DAS |
| --- | --- |
| **Proyecto** | Plataforma Web "La Pica de la Chabelita |
| **Versión** | 1.0 |

| **Documento mantenido por** | Pamela Aldana  Jenifer López |
| --- | --- |
| **Fecha de última revisión** | 20-09-2024 |
| **Fecha de próxima revisión** | 23-09-2024 |

| **Documento aprobado por** | Jenifer Lopez (Scrum Máster) |
| --- | --- |
| **Fecha de última aprobación** | 23-09-2024 |

|  |  |  |  |
| --- | --- | --- | --- |

**Contenido**

[**Introducción** 4](#_heading=h.gjdgxs)

[**Contexto del Problema** 4](#_heading=h.30j0zll)

[**Alcance** 4](#_heading=h.1fob9te)

[**Arquitectura de Software** 5](#_heading=h.3znysh7)

[**Objetivos y Restricciones de la Arquitectura** 6](#_heading=h.2et92p0)

[**Objetivos de la Arquitectura** 6](#_heading=h.tyjcwt)

[**Restricciones de la Arquitectura** 6](#_heading=h.3dy6vkm)

[**Arquitectura Lógica** 7](#_heading=h.1t3h5sf)

[**Parte Estructural** 7](#_heading=h.4d34og8)

[**Parte Dinámica** 8](#_heading=h.2s8eyo1)

[**Arquitectura de Procesos** 9](#_heading=h.17dp8vu)

[**Arquitectura de Desarrollo** 20](#_heading=h.3rdcrjn)

[**Arquitectura Física** 20](#_heading=h.26in1rg)

[**Escenarios** 22](#_heading=h.lnxbz9)

[**Referencias** 29](#_heading=h.35nkun2)

# 

# **Introducción**

El objetivo de este documento es proporcionar una descripción detallada de la arquitectura del software para la Plataforma Web "La Pica de la Chabelita", un sistema que busca optimizar la gestión interna y las ventas del restaurante mediante una solución integral que automatice procesos, mejore la experiencia del usuario y administre eficientemente los recursos.

# **Contexto del Problema**

La gestión actual del restaurante La Pica de la Chabelita presenta problemas significativos como la mala administración de ventas, falta de organización y carencia de automatización en la gestión del stock. Esta situación ha ocasionado estancamiento en el crecimiento del negocio. El objetivo de la plataforma es brindar una solución tecnológica que permita gestionar eficientemente el restaurante, optimizar los procesos de negocio y mejorar la experiencia de los clientes a través de una plataforma web accesible.

# **Alcance**

El sistema permitirá al restaurante gestionar ventas, stock, y pedidos de forma automatizada. Los clientes podrán realizar pedidos en línea, ver el menú disponible, y seguir el estado de sus órdenes. El administrador tendrá acceso a herramientas de gestión del inventario y control de usuarios.

# **Arquitectura de Software**

La arquitectura de la plataforma web está basada en el modelo **4+1**. A continuación, se describen las vistas incluidas:

* **Vista Lógica:** Representa la estructura lógica del sistema con diagramas de clases y secuencias.
* **Vista de Procesos:** Describe la ejecución del sistema, relaciones entre procesos y su sincronización.
* **Vista de Despliegue:** Muestra la topología física e infraestructura donde será desplegado el sistema.
* **Vista Física:** Detalla la estructura de nodos y servidores implicados en el despliegue.
* **Vista de Casos de Uso y Escenarios de Calidad:** Define los actores, principales casos de uso.

# **Objetivos y Restricciones de la Arquitectura**

## **Objetivos de la Arquitectura**

* **Desempeño:** La plataforma debe ofrecer tiempos de respuesta rápidos y manejar múltiples transacciones concurrentes.
* **Tolerancia a Fallos:** El sistema deberá estar diseñado para continuar funcionando ante fallos menores.
* **Seguridad:** Implementar controles estrictos de seguridad, como autenticación y cifrado de datos.
* **Escalabilidad:** El sistema debe ser fácilmente ampliable, permitiendo la integración de nuevas funcionalidades en el futuro.
* **Operatividad:** La plataforma debe ser sencilla de utilizar tanto para administradores como para usuarios finales.

## **Restricciones de la Arquitectura**

* **Plazo de desarrollo:** El proyecto debe estar listo en un plazo de tres meses.
* **Infraestructura limitada:** El servidor inicial debe ser capaz de manejar tanto la base de datos como la aplicación web en un solo entorno.
* **Compatibilidad tecnológica:** El sistema se desarrollará en Django, con una base de datos MySQL y desplegado en Apache.

# 

# **Arquitectura Lógica**

## **Parte Estructural**

El siguiente diagrama de clases representa los principales módulos y sus interacciones. Los módulos incluyen:

* **Módulo de Usuarios:** Gestión de roles y autenticación.
* **Módulo de Pedidos:** Administración de pedidos, estados y transacciones.
* **Módulo de Inventario:** Control y actualización del stock en tiempo real.

**Diagrama de Clases:**

Diagrama

Descripción generada automáticamente

## **Parte Dinámica**

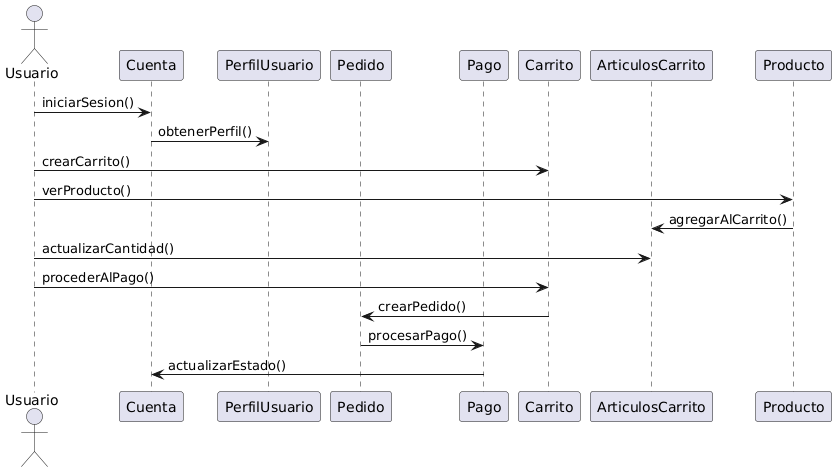
El diagrama de secuencia muestra la interacción dinámica entre los usuarios y la plataforma al momento de realizar un pedido en línea.

**Diagrama de Secuencia:**

Imagen que contiene Calendario

Descripción generada automáticamente

**Diagrama de Comunicación:**

****

# **Arquitectura de Procesos**

Para el sistema de ventas del restaurante "La Picá de la Chabelita", se han identificado y representado a través de las historias de usuario los diagramas de actividades que tiene cada una de ellas. Estos diagramas permiten visualizar cómo tanto los clientes como los administradores interactúan con el sistema de ventas a través de la plataforma. Se utiliza la herramienta Bizagi Modeler para armar las vistas de los procesos de cada una.

**Diagrama de Actividades:**

* 1. **Inicio de sesión (Cliente/ Administrador)**

Diagrama

Descripción generada automáticamente

* 1. **Hacer pedidos en línea (Cliente)**

Diagrama

Descripción generada automáticamente

* 1. **CRUD de pedidos (Administrador)**

Diagrama

Descripción generada automáticamente

* 1. **Panel administrador (Administrador)**

Diagrama

Descripción generada automáticamente

* 1. **CRUD de Administrador (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **Gestión de menú (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **Visualización de ventas (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **CRUD de pagos (Administrador)**

**Diagrama

Descripción generada automáticamente con confianza media**

* 1. **Pagos en línea (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Registro de pagos (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **Reservas en línea (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Confirmación de reserva (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Enviar beneficios por correo (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **Sección de servicios (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Sección de información del restaurante (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Perfil del cliente (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Historial de pedidos (Cliente)**

**Diagrama

Descripción generada automáticamente**

* 1. **Control de stock (Administrador)**

**Diagrama

Descripción generada automáticamente**

* 1. **CRUD de inventario (Administrador)**

**Diagrama

Descripción generada automáticamente**

# 

# **Arquitectura de Despliegue**

La vista de desarrollo detalla cómo está estructurado el código fuente y los componentes. Se destacan:

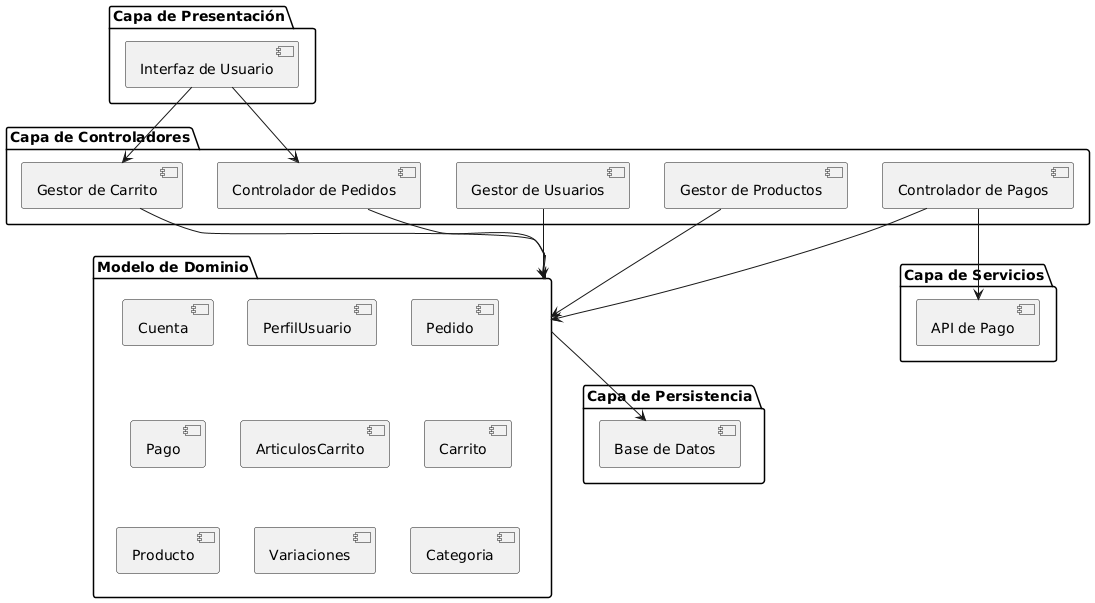
* **Controladores:** Módulos que gestionan la lógica de negocio.
* **Vistas:** Interfaces y plantillas web presentadas al usuario.
* **Modelos:** Interacción con la base de datos.

**Diagrama de Componentes:**

Diagrama

Descripción generada automáticamente

**Diagrama de paquetes:**

****

# **Arquitectura Física**

El sistema se despliega sobre un servidor web Apache, con la base de datos MySQL y la plataforma Django. La infraestructura incluye:

* **Cliente Web:** Navegadores que acceden a la plataforma.
* **Servidor Web:** Procesa peticiones y responde con los datos solicitados.
* **Base de Datos:** Almacena información de pedidos, usuarios, y stock.

**Diagrama de Despliegue:**

Diagrama

Descripción generada automáticamente

# **Vista de Escenarios +1**

**Modelo de Casos de Uso**

Para el sistema de ventas del restaurante "La Picá de la Chabelita", se han identificado diversos casos de uso que cubren las funcionalidades clave tanto para los clientes como para los administradores. Estos casos de uso incluyen acciones como iniciar sesión, realizar pedidos, gestionar el menú y administrar el inventario, entre otras. La documentación de dichos casos se ha realizado utilizando la herramienta CodeToUML, la cual facilita la creación de diagramas de casos de uso claros y bien estructurados. Estos diagramas permiten visualizar de manera efectiva las interacciones entre los actores y el sistema, proporcionando una base sólida para el desarrollo e implementación del software.

**Explicaciones de Relaciones**

**Include**: Indica que un caso de uso incluye siempre el comportamiento de otro caso de uso como parte de su funcionalidad.

**Extend**: Muestra que un caso de uso puede opcionalmente extender el comportamiento de otro caso de uso.

**Diagramas de Casos de Uso:**

1. **Iniciar sesión (Cliente y Administrador)**

Diagrama

Descripción generada automáticamente

1. **Realizar pedido (Cliente)**

**Diagrama

Descripción generada automáticamente**

**3. Realizar pago (Cliente)**

**Diagrama

Descripción generada automáticamente**

**4. Hacer reserva (Cliente)**

**Diagrama

Descripción generada automáticamente**

**5. Ver y actualizar perfil (Cliente)**

**Diagrama

Descripción generada automáticamente**

**6. Consultar historial de pedidos (Cliente)**

**Diagrama

Descripción generada automáticamente**

**7. Consultar servicios (Cliente)Diagrama

Descripción generada automáticamente**

**8**. **Consultar información del restaurante (Cliente)Diagrama

Descripción generada automáticamente**

**9. Gestionar pedidos (Administrador)**Diagrama

Descripción generada automáticamente

**10. Acceder al panel administrativo (Administrador)**

**Diagrama

Descripción generada automáticamente**

**11. Gestionar perfiles de administradores (Administrador)** Diagrama

Descripción generada automáticamente

**12.** **Administrar menú (Administrador)** Diagrama

Descripción generada automáticamente

**13. Consultar ventas (Administrador)** Diagrama

Descripción generada automáticamente

**14. Gestionar métodos de pago (Administrador)** Diagrama

Descripción generada automáticamente

**15. Emitir recibos de pagos (Administrador)**

Diagrama

Descripción generada automáticamente

**16. Gestionar stock (Administrador)**

Diagrama

Descripción generada automáticamente

**17. Gestionar inventario (Administrador)** Diagrama

Descripción generada automáticamente

# **Referencias**

* ISO/IEC 25010 Calidad del Producto Software.
* Documentación de Django (v4.x)
* Bootstrap 5.3
* Documentación API de Transbank (Webpay)