**Arquitectura de software**

**Proyecto: Sistema de gestión de delivery ChickenGo**

Arquitecto de Software: Collachagua Poma Airton Wilson

Versión 2.0

Lima, mayo del 2023

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 22/04 | 1.0 | Arquitectura de software | Collachagua Poma Airton Wilson |
| 27/05 | 2.0 | Capa 2 (Servidor) añadida | Collachagua Poma Airton Wilson |
|  |  |  |  |

**Introducción:**

Una arquitectura de software es el diseño estructural y organizacional de un sistema de software. Define cómo se organizan y se relacionan entre sí los componentes del software, así como las reglas y los principios que rigen su desarrollo. La arquitectura de software es fundamental para el éxito de cualquier proyecto de software, ya que una buena arquitectura permite crear sistemas más eficientes, escalables, mantenibles y adaptables a medida que las necesidades cambian con el tiempo.

1. **Descripción general del sistema**  
   Chicken Go es una plataforma que permitirá al restaurante a gestionar y monitorear el proceso de entrega de sus productos. El sistema de delivery incluye varias funciones y características las cuales facilitarán el proceso, también proporciona a que los usuarios puedan realizar pedidos, seguimiento al estado de sus pedidos y la calificación del servicio para que así sientan la facilidad de comprar desde casa, siendo así nuestra prioridad la satisfacción del cliente al darle un buen servicio.
2. **Tecnologías aplicadas**

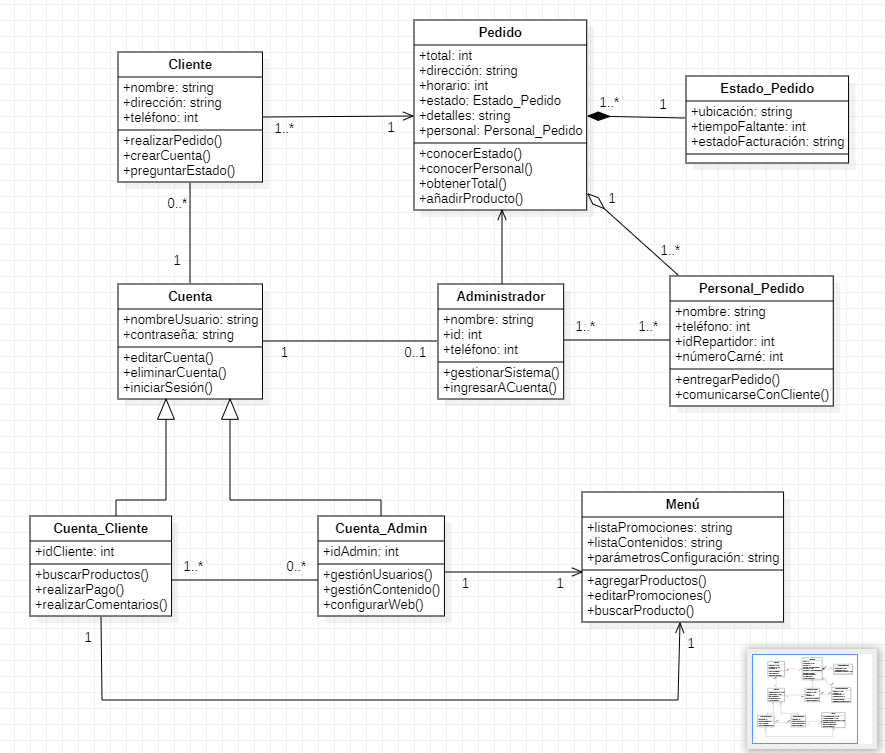
* Lenguaje de Desarrollo: JavaScript, Css,Html.
* Entorno de desarrollo: Visual Studio
* Sistema gestor de base de datos: MySQL
* Modelado del sistema: UML versión 2.0 StarUML
* Sistema de Control de Versiones: Git
* Gestor de Repositorios: Github Desktop

1. **Requisitos del sistema**

* Visualizar carta en línea:
* Agregar platos al carrito:
* Realizar el pago:
* Panel de administración
* Indicar dirección y horario de entrega
* Registro del cliente:
* Visualizar promociones:
* Escribir comentarios:

1. **Diagrama de clases (UML):**

Un diagrama UML (Lenguaje de Modelado Unificado) es una herramienta gráfica utilizada para visualizar, especificar, construir y documentar sistemas de software, en este caso se uso el diagrama de clases donde muestra las clases y relaciones entre ellas que existen en el sistema, así como las funcionalidades que tienen.



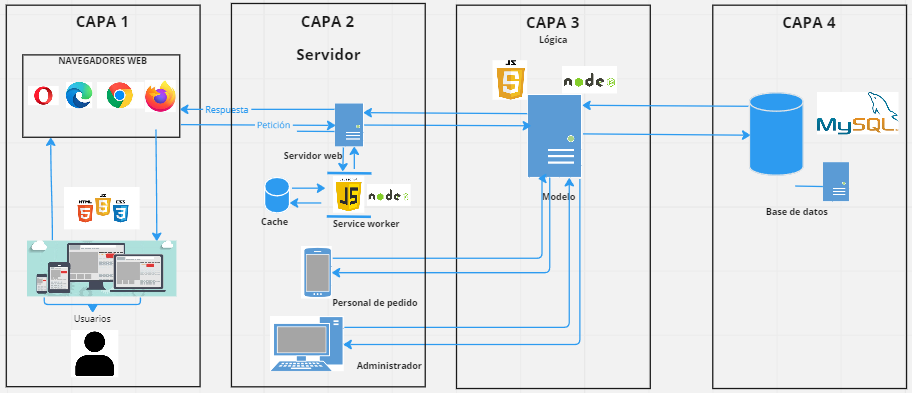
1. **Aplicación de arquitectura:**

El uso de capas es una técnica comúnmente utilizada en el diseño de arquitecturas de software para separar las diferentes responsabilidades del sistema en capas lógicas y físicas. Cada capa se encarga de una parte específica de la funcionalidad del sistema y se comunica con otras capas a través de interfaces bien definidas.  
  
Capa de presentación (capa 1): Esta capa proporciona una interfaz de usuario para que los clientes puedan interactuar con el sistema. Incluye una página web que permite visualizar el menú, agregar platos al carrito, realizar el pago, indicar la dirección y horario de entrega, visualizar promociones y escribir comentarios, en general, se encarga de la presentación visual de la información al usuario y de la interacción con el mismo.

Capa de presentación (capa 2): Esta capa se encarga de interactuar con la base de datos para almacenar y recuperar la información necesaria para el funcionamiento del sistema.   
La capa del servidor actúa como el intermediario entre el cliente y los servicios de backend, procesando solicitudes, aplicando la lógica de negocio, interactuando con servicios externos y generando respuestas adecuadas.

Capa de aplicación (capa 3): Esta capa es responsable de procesar las solicitudes realizadas por los clientes en la capa de presentación. Incluye servicios para procesar órdenes, procesar pagos, gestionar la dirección y el horario de entrega, gestionar el registro del cliente, visualizar el estado del pedido y visualizar las promociones, Esta capa es responsable de la lógica de negocio del sistema, es decir, de implementar los requisitos funcionales del mismo. Es aquí donde se implementan los algoritmos y reglas de negocio necesarios para procesar la información y producir resultados.

Capa de datos (capa 4): Esta capa se encarga de almacenar y gestionar los datos del sistema, como los detalles del menú, los detalles del pedido, la información del cliente, las promociones y la facturación. La capa de persistencia de datos será implementada en una base de datos MySQL, En general, se encarga de la gestión de la base de datos y la interacción con otros sistemas de almacenamiento de datos.



1. **Descripción detallada de los componentes**

Interfaz de usuario: Esta es la parte del sistema que los usuarios finales, es decir, los clientes, utilizan para realizar pedidos. La interfaz de usuario de la página web debe permitir a los clientes navegar por el menú del restaurante, realizar pedidos y realizar pagos.

Gestión de pedidos: Una vez que se realiza un pedido, el sistema de gestión de pedidos del software recibe la información y la procesa automáticamente. Este componente también incluye la capacidad de asignar los pedidos a los repartidores y realizar un seguimiento de los mismos.

Gestión de inventario: El software de delivery para un restaurante debe mantener un registro actualizado de los ingredientes y suministros disponibles. De esta manera, el sistema puede alertar a los gerentes del restaurante cuando ciertos ingredientes se estén agotando o cuando sea necesario realizar un pedido de suministros.

Gestión de entregas: La gestión de entregas implica la asignación de los repartidores a los pedidos, así como el seguimiento en tiempo real de la ubicación de los mismos. Esta función también incluye la capacidad de proporcionar actualizaciones a los clientes sobre el estado de su entrega, como la hora estimada de llegada.

Gestión de pagos: El software debe ser capaz de procesar pagos de forma segura y eficiente. Esto puede incluir la integración con proveedores de servicios de pago en línea, así como la capacidad de gestionar reembolsos y cancelaciones.