

# Memoria Descriptiva de la aplicación

## AltokePe

### 1. Introducción

AltokePe es una aplicación de gestión de pedidos para restaurantes desarrollada bajo una arquitectura cliente-servidor distribuida. El sistema permite coordinar en tiempo real la comunicación entre el módulo recepcionista, las estaciones de cocina y el panel de ranking.

La plataforma optimiza la atención al cliente, reduce los tiempos de espera y mejora la trazabilidad completa de cada pedido desde su registro hasta su entrega final.

Para el desarrollo del sistema se emplearon tecnologías modernas como C++17, Qt 6 para el diseño de interfaces gráficas, comunicación mediante el protocolo TCP/IP y el formato JSON para el intercambio de datos entre los distintos módulos.

El proyecto fue desarrollado siguiendo la metodología ágil SCRUM, estructurado en cinco sprints semanales, lo que permitió realizar entregas incrementales y una adaptación continua a los requerimientos del sistema.



## **2. Arquitectura y Plataforma Tecnológica**

### **a. Componentes**

El sistema AltokePe se encuentra estructurado bajo una arquitectura cliente-servidor, compuesta por los siguientes módulos:

- Servidor Central: encargado de gestionar las conexiones TCP/IP, coordinar la comunicación entre todos los clientes conectados, administrar los estados de los pedidos y sincronizar las notificaciones en tiempo real.
- Cliente Recepcionista: módulo gráfico que permite registrar nuevos pedidos, modificarlos, asignar mesas y generar facturas.
- Clientes de Cocina (Estaciones): interfaces específicas por estación (Snack, Bebidas, Principal, Otros) que reciben los pedidos asignados y permiten actualizar el estado de preparación.
- Cliente Ranking: panel de visualización que muestra el ranking de platos más vendidos, estadísticas de ventas y métricas del rendimiento general del restaurante.

### **b. Tecnologías Utilizadas**

- Lenguaje de programación: C++17
- Framework de interfaces: Qt 6
- Protocolo de comunicación: TCP/IP
- Formato de intercambio de datos: JSON
- Sistema de construcción: CMake
- Control de versiones: Git y GitHub
- Sistemas operativos soportados: Windows y Linux

### c. Diagrama de Arquitectura

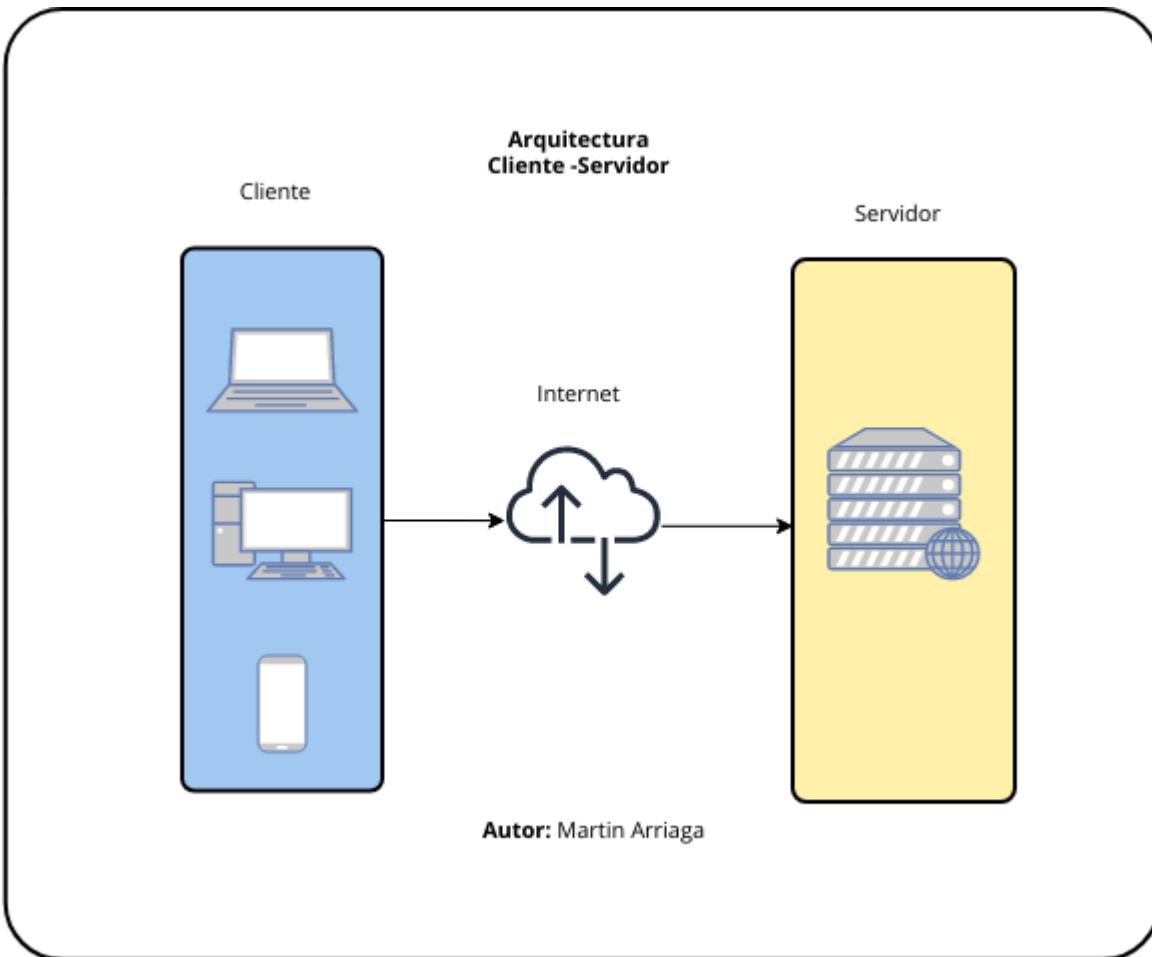


Figura 1: Diagrama de Arquitectura Cliente-Servidor

Fuente:<https://online.visual-paradigm.com/es/community/share/arquitectura-cliente-servidor-v0oazbv65>

## 3. Funcionalidades del Sistema

**a. Módulo recepcionista:** El Módulo Recepcionista permite registrar nuevos pedidos del restaurante, asignarlos a mesas específicas, modificarlos mientras están en curso y generar facturas de manera automática. Además, ofrece la capacidad de recibir notificaciones en tiempo real sobre el estado de preparación de los platos, lo que facilita la coordinación entre la atención al cliente y la cocina.

**b. Módulo cocina:** El módulo de Cocina permite la visualización exclusiva de los pedidos asignados a cada estación, asegurando una

adecuada distribución de tareas entre las áreas especializadas (Snack, Bebidas, Principal y Otros). Los operadores pueden actualizar el estado de los platos conforme avanzan en el proceso de preparación, pasando de pendiente a en preparación y finalmente a listo. Adicionalmente, el sistema genera notificaciones automáticas ante la llegada de nuevos pedidos, facilitando una atención organizada y oportuna.

- c. **Módulo ranking:** El módulo Ranking se encarga de mostrar el ranking de los platos más vendidos del restaurante, presentando estadísticas detalladas según cantidad vendida y nivel de popularidad. Dichos datos se actualizan automáticamente conforme se registran nuevos pedidos, permitiendo analizar el rendimiento comercial del restaurante y detectar tendencias de consumo en tiempo real.
- d. **Funcionalidades del servidor :** El Servidor Central gestiona simultáneamente múltiples conexiones TCP provenientes de los distintos módulos del sistema, coordinando la sincronización de los estados de los pedidos para garantizar la consistencia de la información compartida. Además, se encarga de la persistencia de los datos del menú y de los pedidos mediante archivos en formato JSON, asegurando el respaldo de la información operativa, y mantiene una comunicación bidireccional continua entre todos los clientes conectados para asegurar el correcto funcionamiento integral del sistema.

## 4. Proceso de instalación

## 5. Roles de Usuario y Permisos

- a. **Recepcionista:** El rol de Recepcionista se encarga del registro de pedidos, la gestión de mesas, la generación de facturas y la coordinación directa con el área de cocina. Sus permisos incluyen la creación, modificación y eliminación de pedidos, así como la visualización del estado completo de todos los pedidos del sistema, contando con acceso total al módulo de atención.

- b. Cocinero / Estación:** El rol de Cocinero o Estación es responsable de recibir los pedidos asignados a su área específica, actualizar los estados de preparación de los platos y notificar cuando estos se encuentran listos para su entrega. Entre sus permisos se encuentra la visualización exclusiva de los pedidos correspondientes a su estación y la modificación de los estados de preparación de dichos pedidos.
- c. Chef / Manager:** El Chef o Manager cumple la función de supervisar el trabajo de todas las estaciones de cocina y coordinar los tiempos de preparación de los pedidos para garantizar una operación eficiente. Posee permisos de visualización general sobre todos los pedidos del sistema y la capacidad de modificar los estados de preparación de cualquier estación cuando sea necesario.
- d. Ranking:** El rol de Gerente, a través del módulo Ranking, se dedica al monitoreo del rendimiento general del restaurante y al análisis de las estadísticas de ventas. Sus permisos son de solo lectura, permitiéndole acceder a los rankings de platos y a las métricas del sistema sin posibilidad de realizar modificaciones sobre los pedidos.

## 6. Gestión de Menú y los Pedidos

- a. Gestión del menú:** El sistema administra un menú almacenado en formato JSON que contiene un total de 44 platos (depende del menú proporcionado por el cliente) distribuidos en cuatro categorías: Snack con 16 platos, Bebidas con 11 platos, Principal con 10 platos y Otros con 7 platos. Cada plato cuenta con un identificador único, nombre, costo, tiempo estimado de preparación y estación asignada, lo que permite una organización adecuada de la información gastronómica del restaurante.
- b. Gestión de Pedidos:** El ciclo de vida de un pedido dentro del sistema inicia en el estado Creado, continúa como Pendiente, pasa a En preparación, luego a Listo y finalmente concluye con el estado Entregado. Cada pedido registra información detallada que incluye un

identificador único, la mesa asignada, la lista de platos solicitados con sus respectivas cantidades, el estado actual del pedido, la marca de tiempo correspondiente a su creación, el tiempo estimado de preparación y el costo total.



Figura 2: Ciclo de Vida de un pedido

## 7. Conclusiones

El proyecto AltokePe permitió desarrollar una solución integral para la gestión de pedidos en restaurantes, destacando por su eficiente comunicación en tiempo real entre módulos, su arquitectura modular escalable y la facilidad de uso de sus interfaces gráficas. La aplicación de la metodología SCRUM, estructurada en cinco sprints, favoreció un desarrollo ordenado y adaptable a los requerimientos del proyecto, permitiendo una detección temprana de

problemas y la implementación continua de mejoras. Como líneas de trabajo futuro se contemplan la integración de sistemas de pago electrónico, el desarrollo de una aplicación móvil para clientes, la generación de reportes avanzados, la incorporación de un sistema de control de inventarios y la ampliación del sistema para brindar soporte a múltiples restaurantes.

## **8. Equipo de Desarrollo**

- Ajra Huacso Jeans Anthony
- Luque Condori Luis Guillermo
- Garambel Marin Fernando Miguel
- Mamani Huarsaya Jorge Luis
- Hancco Mullisaca Sergio Danilo
- Mollo Chuquicaña Dolly Yadhira
- Sucle Suca Michael Benjamin