



UNIVERSIDAD DE TALCA
INGENIERÍA CIVIL EN COMPUTACIÓN

Proyecto 2

Sistema Distribuido de Gestión de Estaciones de Combustible

Entregable 1: Diseño de Arquitectura

Curso: Sistemas Distribuidos
Profesor: Rodrigo Pavez (rpavez@utalca.cl)

Integrantes:

Pablo Correa
Benjamín Navarro
Sebastián Retamal

28 de Octubre de 2025

1. Introducción

El presente documento describe el diseño arquitectónico de un sistema distribuido de tres niveles para la gestión integral de una empresa de combustibles. Este sistema permite la administración centralizada de precios, la recolección de información de transacciones desde múltiples estaciones de servicio distribuidas geográficamente, y la operación autónoma de cada distribuidor en caso de pérdida de conectividad con la casa matriz.

1.1. Contexto del Problema

Una empresa de combustibles opera múltiples estaciones de servicio distribuidas geográficamente. Cada estación cuenta con cuatro surtidores multifunción que dispensan los cinco tipos de combustible que comercializa la empresa: Gasolina 93, Gasolina 95, Gasolina 97, Diesel y Kerosene. Cada surtidor es capaz de despachar cualquiera de estos cinco combustibles según la selección del cliente, similar a las bombas modernas de estaciones de servicio reales.

La empresa enfrenta los siguientes desafíos operacionales:

- **Control centralizado de precios:** La casa matriz necesita definir precios base para cada uno de los cinco tipos de combustible, que se propaguen automáticamente a todas las estaciones de servicio a nivel nacional.
- **Autonomía operacional:** Cada estación debe poder aplicar su propio factor de utilidad sobre los precios base recibidos, permitiendo estrategias de pricing locales adaptadas a condiciones de mercado específicas.
- **Registro exhaustivo de transacciones:** Todas las ventas deben quedar registradas con información detallada incluyendo tipo de combustible dispensado, litros, precio aplicado y timestamp de la operación.
- **Resiliencia ante fallas:** El sistema debe continuar operando localmente en cada estación ante pérdidas de conectividad con la casa matriz, y sincronizar automáticamente al recuperar la conexión.
- **Actualización inteligente de precios:** Los surtidores no pueden cambiar precios durante una transacción activa, requiriendo un mecanismo de encolamiento de actualizaciones pendientes.
- **Redundancia de datos:** Garantizar la integridad y disponibilidad de la información mediante estrategias de respaldo y replicación.

1.2. Elementos de la Solución

Arquitectura Distribuida de Tres Niveles El sistema se estructura en una jerarquía de tres niveles claramente definidos: Casa Matriz (nivel superior de gestión y control), Distribuidores (estaciones de servicio intermediarias), y Surtidores (dispositivos multifunción de dispensado de combustible). Esta jerarquía permite una separación clara de responsabilidades y facilita la escalabilidad del sistema.

Comunicación Basada en TCP Se utiliza el protocolo TCP para la comunicación crítica entre componentes del sistema, garantizando la entrega confiable y ordenada de mensajes de actualización de precios y sincronización de transacciones. La comunicación se realiza mediante sockets nativos de Python, proporcionando control total sobre el protocolo de comunicación.

Interfaces Web con Flask Cada componente del sistema expone una interfaz web desarrollada con Flask, permitiendo la gestión, monitoreo y simulación de operaciones desde navegadores web estándar. Las interfaces utilizan HTML, CSS y JavaScript para proporcionar experiencias de usuario interactivas.

Persistencia con SQLite Los distribuidores utilizan SQLite como sistema de gestión de bases de datos embebido, almacenando transacciones, configuraciones de surtidores y precios actuales para los cinco tipos de combustible. SQLite fue seleccionado por su simplicidad, confiabilidad y capacidad de operar sin servidor dedicado.

Redundancia Dual Se implementa una estrategia de respaldo que combina backups periódicos de la base de datos (cada 5 minutos) con un log continuo de transacciones en formato texto, permitiendo la recuperación ante fallas y la detección de inconsistencias.

Contenedorización con Docker Todo el sistema se despliega mediante contenedores Docker orquestados con Docker Compose, facilitando el despliegue, la portabilidad entre entornos, y la simulación de una infraestructura distribuida en un único host para propósitos de desarrollo y pruebas.

Formato de Mensajes JSON Todos los mensajes intercambiados entre componentes utilizan formato JSON, proporcionando un estándar de serialización legible, ampliamente soportado y fácil de depurar.

2. Diseño de la Arquitectura

2.1. Elementos de la Arquitectura y sus Roles

La arquitectura del sistema se compone de tres niveles jerárquicos, cada uno con componentes específicos que desempeñan roles claramente definidos. La Figura 1 muestra la estructura física completa del sistema.

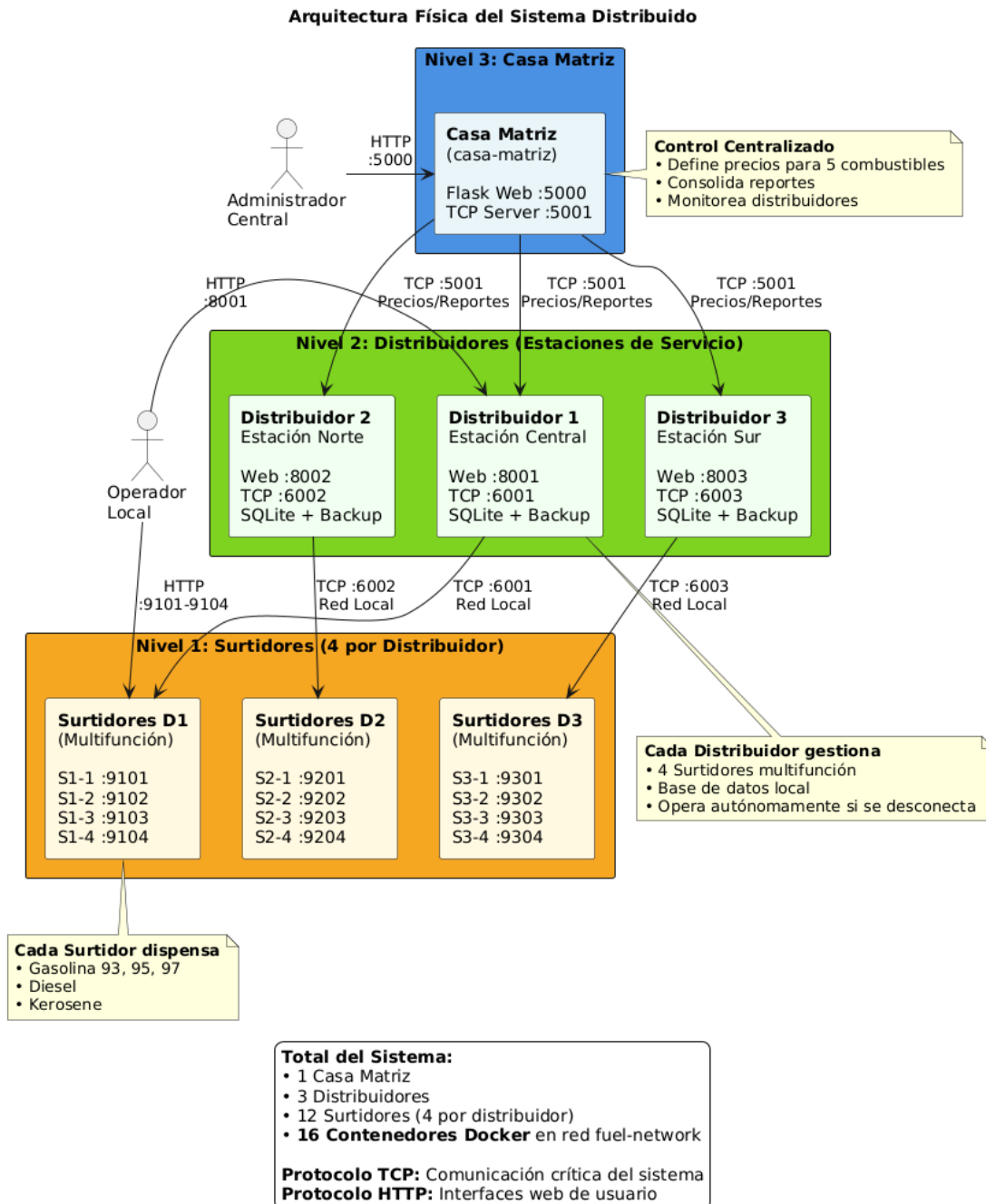


Figura 1: Arquitectura física del sistema distribuido de gestión de combustibles

2.1.1. Nivel 3: Casa Matriz

Descripción General La Casa Matriz constituye el nodo central del sistema y representa el nivel más alto de la jerarquía. Actúa como el único punto de definición de políticas de precios corporativos para los cinco tipos de combustible y como receptor de información consolidada de todas las estaciones de servicio de la red.

Componentes

- **Servidor TCP (Puerto 5001):** Servidor que escucha conexiones entrantes de los distribuidores. Mantiene un registro en memoria de todos los distribuidores conectados activamente, detecta desconexiones mediante timeouts y gestiona la comunicación bidireccional para envío de comandos y recepción de reportes.
- **Servidor Web Flask (Puerto 5000):** Aplicación web que proporciona una interfaz de administración completa. Permite la definición centralizada de precios para cada uno de los cinco tipos de combustible (93, 95, 97, Diesel, Kerosene), visualización en tiempo real del estado de conexión de distribuidores, generación de reportes consolidados de ventas y exportación de datos en formatos estándar (CSV, PDF).
- **Gestor de Distribuidores:** Módulo de software que mantiene el estado de cada distribuidor conectado, incluyendo información de última conexión, número de surtidores activos, y métricas agregadas de ventas por tipo de combustible. Implementa lógica de heartbeat para detección proactiva de fallas de conectividad.
- **Generador de Reportes:** Componente que consulta información a todos los distribuidores conectados, consolida datos de transacciones y genera reportes ejecutivos con métricas de negocio segmentadas por tipo de combustible (volumen total vendido, número de transacciones, ingresos totales, distribución geográfica de ventas).

Roles e Impacto

1. **Control de Precios Corporativos:** Define los precios base de los cinco tipos de combustible a nivel nacional, asegurando consistencia en la estrategia de pricing de la empresa.
2. **Punto Central de Comunicación:** Actúa como hub de comunicación con todos los distribuidores, simplificando la topología de red y facilitando la administración centralizada.
3. **Consolidación de Información:** Agrega datos de todas las estaciones para proporcionar visibilidad completa del negocio a nivel ejecutivo, con desagregación por tipo de combustible.
4. **Monitoreo de Red:** Supervisa la salud de la red de distribuidores, detectando problemas de conectividad y generando alertas operacionales.

El impacto de este nivel es crítico: sin la Casa Matriz operacional, no es posible actualizar precios de forma centralizada ni obtener reportes consolidados. Sin embargo, gracias a la autonomía de los distribuidores, las operaciones locales pueden continuar.

2.1.2. Nivel 2: Distribuidores (Estaciones de Servicio)

Descripción General Los Distribuidores representan las estaciones de servicio físicas distribuidas geográficamente. Funcionan como nodos intermediarios entre la Casa Matriz y los surtidores locales, con capacidad de operación completamente autónoma durante períodos de desconexión. Cada distribuidor gestiona cuatro surtidores multifunción.

Componentes

- **Cliente TCP:** Cliente que establece y mantiene conexión persistente con la Casa Matriz (puerto 5001). Recibe actualizaciones de precios corporativos para los cinco tipos de combustible y envía reportes periódicos de transacciones locales. Implementa reconexión automática con backoff exponencial ante fallas.
- **Servidor TCP Local (Puerto 6000):** Servidor que gestiona conexiones de los cuatro surtidores en la red local de la estación. Propaga actualizaciones de precios de los cinco combustibles a todos los surtidores y recibe notificaciones de transacciones completadas.
- **Servidor Web Flask (Puerto 8000):** Interfaz web local para monitoreo y gestión de la estación. Muestra estado en tiempo real de los cuatro surtidores, permite visualización de transacciones locales, y proporciona controles para operaciones manuales de respaldo y sincronización.
- **Base de Datos SQLite:** Almacena toda la información crítica del distribuidor:
 - Tabla **surtidores**: Configuración y estado actual de cada uno de los cuatro surtidores (ID, precios actuales de los 5 combustibles, litros totales dispensados por tipo, número de cargas por tipo, estado operacional).
 - Tabla **transacciones**: Registro completo de todas las ventas (ID transacción, ID surtidor, tipo de combustible, timestamp, litros, precio por litro, total en pesos, flag de sincronización con Casa Matriz).
 - Tabla **precios_historico**: Auditoría de cambios de precio para cada tipo de combustible con timestamp y origen (casa matriz o local).
- **Sistema de Respaldo Dual:**
 - *Backup de Base de Datos*: Copia completa del archivo SQLite cada 5 minutos en volumen Docker persistente.
 - *Log de Transacciones*: Archivo append-only que registra todas las operaciones en formato JSON, permitiendo reconstrucción del estado del sistema y auditoría completa.
- **Gestor de Cola de Precios:** Módulo que maneja actualizaciones de precios pendientes para surtidores que están actualmente dispensando combustible. Cuando un surtidor cambia a estado LIBRE, aplica automáticamente cualquier actualización en cola para los cinco tipos de combustible.

- **Monitor de Conectividad:** Componente que detecta pérdidas de conexión con la Casa Matriz mediante timeouts y fallas de heartbeat. Activa automáticamente el modo de operación local autónoma y gestiona el proceso de reconexión y sincronización.

Roles e Impacto

1. **Intermediación Inteligente:** Recibe precios corporativos para los cinco combustibles y los transforma aplicando factores de utilidad locales antes de propagarlos a los cuatro surtidores.
2. **Persistencia de Datos:** Garantiza que todas las transacciones de todos los tipos de combustible queden registradas incluso durante desconexiones con la Casa Matriz.
3. **Gestión de Surtidores:** Coordina las operaciones de cuatro surtidores multifunción, gestionando actualizaciones de precio simultáneas para los cinco combustibles y recolección de datos de transacciones.
4. **Operación Autónoma:** Continúa operando de forma independiente durante fallas de conectividad, asegurando continuidad del negocio.
5. **Sincronización:** Reconcilia datos locales con la Casa Matriz al recuperar conectividad, enviando todas las transacciones acumuladas durante el período de desconexión.

2.1.3. Nivel 1: Surtidores

Descripción General Los Surtidores representan los dispositivos físicos terminales que interactúan directamente con los clientes para dispensar combustible. Cada surtidor es multifunción, capaz de despachar cualquiera de los cinco tipos de combustible que comercializa la empresa (Gasolina 93, 95, 97, Diesel y Kerosene), similar a las bombas modernas de estaciones de servicio que cuentan con múltiples mangueras para diferentes productos. Cada distribuidor cuenta con cuatro surtidores de este tipo.

Componentes

- **Cliente TCP:** Mantiene conexión permanente con su distribuidor asignado. Envía notificaciones de cambio de estado (`LIBRE` ↔ `EN_OPERACION`) y datos de transacciones completadas especificando el tipo de combustible dispensado.
- **Servidor Web Flask (Puerto 9000):** Interfaz web simple para simulación de operaciones de dispensado y visualización del estado actual. Permite simular cargas de cualquiera de los cinco tipos de combustible, especificando el tipo y litros a dispensar.
- **Tabla de Precios:** Almacena los precios actuales de los cinco tipos de combustible:
 - Gasolina 93
 - Gasolina 95
 - Gasolina 97
 - Diesel
 - Kerosene
- **Máquina de Estados:** Controla el ciclo de vida operacional del surtidor:
 - *Estado LIBRE:* Surtidor disponible para nuevas transacciones y para recibir actualizaciones de precio de cualquier combustible.
 - *Estado EN_OPERACION:* Surtidor dispensando combustible activamente. Rechaza actualizaciones de precio que son encoladas por el distribuidor.
- **Buffer de Actualización:** Memoria temporal para almacenar actualizaciones de precios recibidas durante operaciones. Al finalizar la transacción, todos los precios almacenados en buffer se aplican automáticamente.
- **Generador de Transacciones:** Simula el proceso de dispensado de combustible. El cliente selecciona el tipo de combustible deseado, se calcula el monto total basado en litros dispensados y precio vigente de ese combustible, genera timestamp, y crea el objeto de transacción para envío al distribuidor.

- **Contadores Acumulativos por Tipo de Combustible:** Mantiene estadísticas diarias segmentadas:
 - Litros totales dispensados por cada tipo de combustible
 - Número de cargas realizadas por cada tipo
 - Monto total recaudado por cada tipo

Roles e Impacto

1. **Punto de Venta Terminal Multifunción:** Interfaz directa con clientes finales, ejecutando transacciones de venta de cualquiera de los cinco tipos de combustible.
2. **Recolección de Datos Segmentados:** Genera información primaria de transacciones especificando el tipo de combustible dispensado, que fluye hacia arriba en la jerarquía para consolidación y análisis.
3. **Aplicación de Precios Diferenciados:** Utiliza el precio vigente correspondiente al tipo de combustible seleccionado en cada transacción, garantizando que el precio no cambie durante una operación activa.
4. **Consistencia Operacional:** Mantiene la integridad de cada transacción mediante su máquina de estados, previniendo condiciones de carrera y estados inconsistentes independientemente del combustible dispensado.
5. **Flexibilidad Operacional:** Permite a los clientes seleccionar cualquier tipo de combustible disponible, maximizando la utilización del activo y mejorando la experiencia del cliente.

2.2. Elementos de Comunicación entre Niveles

2.2.1. Protocolos de Comunicación

Comunicación TCP (Sistema Distribuido Crítico) **Alcance:** Comunicación entre componentes del sistema distribuido.

Conexiones:

- Casa Matriz \leftrightarrow Distribuidor: Puerto 5001
- Distribuidor \leftrightarrow Surtidor: Puerto 6000

Características Técnicas:

- Conexiones persistentes mantenidas activamente durante toda la operación del sistema.
- Garantía de entrega ordenada mediante el protocolo TCP/IP.

Justificación: TCP fue seleccionado para la comunicación crítica porque garantiza que los mensajes de actualización de precio (para los cinco combustibles) y las transacciones nunca se pierdan, lleguen en el orden correcto, y se detecten inmediatamente fallas de conectividad. Estas garantías son fundamentales para la integridad del sistema.

Comunicación HTTP/REST (Interfaces de Usuario) **Alcance:** Interacción entre usuarios y el sistema a través de navegadores web.

Características Técnicas:

- Protocolo HTTP/1.1 stateless.
- Endpoints RESTful para consultas (GET) y comandos (POST).
- Respuestas en formato JSON para fácil procesamiento en el frontend.
- Actualización de datos mediante polling cada 5 segundos (JavaScript `setInterval`).
- Códigos de estado HTTP estándar para indicar éxito o error de operaciones.

Justificación: HTTP/REST es el estándar para aplicaciones web modernas, proporciona compatibilidad universal con navegadores, y no requiere instalación de software adicional en las máquinas cliente.

2.3. Arquitectura Física y Vínculos

2.3.1. Topología de Red

El sistema utiliza Docker Compose para crear una red virtual privada denominada `fuel-network` que interconecta todos los componentes. Esta red simula una infraestructura corporativa con las siguientes características:

- **Red Bridge:** Tipo de red Docker que permite comunicación entre contenedores mediante nombres de servicio.
- **DNS Interno:** Docker proporciona resolución automática de nombres, permitiendo que los contenedores se referencien entre sí usando nombres de servicio en lugar de IPs (ej: `casa-matriz`, `distribuidor-1`).
- **Aislamiento:** La red es privada y aislada del host, simulando un entorno de producción real.
- **Mapeo de Puertos:** Puertos internos de contenedores se mapean a puertos del host para acceso desde navegadores web.

2.3.2. Vínculos de Comunicación

La Figura 1 muestra los vínculos entre componentes. Los vínculos se clasifican en dos categorías:

Vínculos TCP (Comunicación del Sistema)

- **Casa Matriz - Distribuidor 1:** TCP persistente, puerto 5001
- **Casa Matriz - Distribuidor 2:** TCP persistente, puerto 5001
- **Casa Matriz - Distribuidor 3:** TCP persistente, puerto 5001
- **Distribuidor 1 - Surtidores 1.1-1.4:** TCP persistente, puerto 6001
- **Distribuidor 2 - Surtidores 2.1-2.4:** TCP persistente, puerto 6002
- **Distribuidor 3 - Surtidores 3.1-3.4:** TCP persistente, puerto 6003

Vínculos HTTP (Interfaces de Usuario)

- **Navegador - Casa Matriz:** HTTP, `localhost:5000`
- **Navegador - Distribuidor 1:** HTTP, `localhost:8001`
- **Navegador - Distribuidor 2:** HTTP, `localhost:8002`
- **Navegador - Distribuidor 3:** HTTP, `localhost:8003`
- **Navegador - Surtidores:** HTTP, `localhost:9101-9304`

2.3.3. Flujo de Datos Típico

Escenario 1: Actualización de Precios

1. Administrador ingresa nuevos precios para los cinco combustibles en interfaz web de Casa Matriz (HTTP POST a `localhost:5000`).
2. Casa Matriz envía mensaje JSON con los cinco precios a todos los distribuidores conectados vía TCP.
3. Cada distribuidor recibe los precios, aplica su factor de utilidad local a cada uno, y actualiza su base de datos.
4. Cada distribuidor propaga los cinco precios ajustados a sus cuatro surtidores vía TCP.
5. Surtidores en estado LIBRE actualizan inmediatamente los cinco precios; surtidores EN_OPERACION encolan las actualizaciones.
6. Cada distribuidor envía confirmación a Casa Matriz con detalle de surtidores actualizados y pendientes.
7. Casa Matriz actualiza su dashboard mostrando el estado de propagación de los precios.

Escenario 2: Registro de Transacción Multifunción

1. Cliente selecciona Gasolina 95 en Surtidor 1-1 (simulado desde interfaz web `localhost:9101`).
2. Surtidor cambia a estado EN_OPERACION y notifica al distribuidor.
3. Se simula dispensado de 45 litros de Gasolina 95 al precio vigente (\$1580/litro).
4. Surtidor calcula total (\$71,100) y genera objeto de transacción especificando tipo combustible "95" con timestamp.
5. Surtidor cambia a estado LIBRE y envía transacción al Distribuidor 1 vía TCP.
6. Distribuidor 1 inserta transacción en tabla `transacciones` de SQLite, registrando el tipo de combustible.
7. Distribuidor 1 actualiza contadores acumulativos de Gasolina 95 para el surtidor.
8. Distribuidor 1 escribe transacción en log append-only para respaldo.
9. Si hay conexión con Casa Matriz, distribuidor eventualmente sincroniza la transacción.
10. Administrador puede visualizar la transacción en interfaz web del Distribuidor 1 (`localhost:8001`), con detalle del tipo de combustible dispensado.