

# **Protocolo Intragrupal**

## **Introducción**

Este documento describe el protocolo de comunicación para el curso CI-0123 Grupo 6, que implementa un sistema de almacenamiento y gestión de figuras distribuido entre múltiples servidores.

## **Arquitectura del Sistema**

El sistema está compuesto por tres componentes principales:

### **Cliente (CLI)**

- Componente que inicia las peticiones al sistema.
- Se conecta al Tenedor mediante TCP en el puerto 8080.
- Envía comandos para gestionar figuras.

### **Tenedor (T)**

- Componente intermediario que actúa como coordinador.
- Puertos TCP:
  - Puerto 8080: Escucha conexiones de clientes.
  - Puerto 8081: Se conecta al servidor de figuras.
- Puertos UDP:
  - Puerto 4321: Para señalización con servidores.
- Funciones:
  - Mantiene registro de qué figuras están en qué servidor.
  - Distribuye peticiones a los servidores correspondientes.
  - Decide arbitrariamente en qué servidor almacenar nuevas figuras.

### **Servidor (S)**

- Componente de almacenamiento de figuras.
- Puertos TCP:
  - Puerto 8081: Escucha por solicitudes del tenedor.
  - Puerto 8082: Se conecta al caché.
- Puertos UDP:
  - Puerto 1234: Para señalización UDP.
- Almacena y gestiona las figuras asignadas.

### **Cache (C)**

- Componente intermedio que reenvía figuras ya procesadas sin necesidad de volverlas a consultar al servidor de figuras
- Puertos TCP:
  - Puerto 8082: Escucha por conexiones del servidor de figuras
- Funciones:
  - Almacena algunas figuras pedidas recientemente y las reenvía el mismo si dispone de ellas, de no tener una figura se la solicita al servidor y de ser posible la almacena.
  - Si el servidor se cierra, este recibirá una respuesta por la conexión TCP que le indica que debe cerrar las conexiones y morir

## Comunicación

### TCP

- Cliente ↔ Tenedor: Puerto 8080.
- Tenedor ↔ Servido: Puerto 8081.
- Servidor ↔ Caché: Puerto 8082.

### UDP

- Tenedor: Puerto 4321.
- Servidor: Puerto 1234 (Fork) y Puerto 8765 (Caché).
- (Opcional) Caché: Puerto 5678.
- Propósito: Notificación de estado (encendido/apagado) entre Tenedor y Servidores.

## Formato de Mensajes

Todos los mensajes siguen el siguiente formato:

COMANDO /BEGIN/ [parámetros] /END/

Reglas del formato: Los comandos se envían en mayúscula. Los parámetros se delimitan con /BEGIN/ y /END/. Los parámetros son cadenas de texto (strings) separados por el carácter /.

## Comandos

### LIST

**Propósito:** Obtener la lista de figuras disponibles en el sistema.

**Formato:**

LIST /BEGIN/ /END/

**Comportamiento:** El Tenedor consulta su registro de figuras, devuelve la lista completa de figuras disponibles en todos los servidores.

## GET

**Propósito:** Obtener una figura específica.

**Formato:**

GET /BEGIN/ nombre\_figura /END/

**Parámetros:**

nombre\_figura: Nombre de la figura a obtener

**Comportamiento:** El Tenedor consulta en qué servidor está la figura, envía la petición al servidor correspondiente y retorna el contenido de la figura al cliente.

## ADD

**Propósito:** Agregar una nueva figura al sistema.

**Formato:**

ADD /BEGIN/ nombre\_figura / tamaño /END/

**Parámetros:**

nombre\_figura: Nombre de la nueva figura

tamaño: Tamaño en bytes de la figura

contenido\_figura: Datos de la figura

**Comportamiento:** El Tenedor decide arbitrariamente en qué servidor almacenar la figura, envía la figura al servidor seleccionado y actualiza su registro de distribución de figuras.

## DELETE

**Propósito:** Eliminar una figura del sistema.

**Formato:**

DELETE /BEGIN/ nombre\_figura /END/

**Parámetros:**

nombre\_figura: Nombre de la figura a eliminar

**Comportamiento:** El Tenedor identifica en qué servidor está la figura, envía comando de eliminación al servidor correspondiente y actualiza su registro removiendo la figura.

## CONNECT

**Propósito:** Establecer conexión entre componentes.

**Formato:**

CONNECT /BEGIN/ ip /END/

**Parámetros:**

ip: IP de donde se pide la conexión

**Comportamiento:** Se envía por UDP para establecer conexión TCP entre Tenedor-Servidor. Inicia sesión de comunicación.

QUIT

**Propósito:** Cerrar conexión entre componentes.

**Formato:**

QUIT /BEGIN/ ip /END/

**Parámetros:**

ip: IP de donde se pide la conexión

**Comportamiento:** Notifica del apagado del componente para cerrar la conexión.

### **Manejo de Errores**

**Código:** 200 - OK

**Significado:** Operación exitosa.

**Casos de uso:**

- Comando ejecutado correctamente.
- Figura encontrada y retornada.
- Figura agregada exitosamente.
- Figura eliminada correctamente.

**Código:** 400 - Bad Request

**Significado:** Error en la petición del cliente.

**Casos comunes:**

- Formato de comando incorrecto.
- Parámetros faltantes o inválidos.
- Comando no reconocido.
- Sintaxis incorrecta en delimitadores.

**Código:** 500 - Internal Server Error

**Significado:** Error interno del sistema.

**Casos comunes:**

- Error de comunicación entre Tenedor y Servidor.
- Fallo en el almacenamiento de figuras.
- Servidor no disponible.
- Error en la distribución de datos.

## **Señalización de Estado**

### **Notificación de Encendido:**

- Los Servidores notifican al Tenedor vía UDP cuando se inician.
- El Tenedor registra la disponibilidad del servidor.
- Se establece conexión TCP para operaciones.

### **Notificación de Apagado:**

- Los Servidores notifican al Tenedor vía UDP cuando se van a apagar.
- El Tenedor marca el servidor como no disponible.
- Se cierran las conexiones TCP activas.

## **Consideraciones de Implementación**

### **Distribución de Carga:**

- El Tenedor decide arbitrariamente el servidor de destino para nuevas figuras.
- No se implementa balanceamiento de carga específico.

### **Tolerancia a Fallos:**

- El sistema UDP permite detectar servidores caídos.
- El Tenedor debe manejar servidores no disponibles.
- Se requiere mecanismo de recuperación para figuras en servidores caídos.

### **Concurrencia:**

- Múltiples clientes pueden conectarse simultáneamente al Tenedor.
- El Tenedor debe manejar conexiones concurrentes.
- Los servidores deben soportar múltiples operaciones simultáneas.