

Propuesta de protocolo

Se prefiere trabajar bajo el protocolo de HTTP ya que es perfecto para establecer conexiones entre cliente-servidor en la web que es lo que se busca en este proyecto.

Además el HTTP es como el TCP, lo cual es importante porque transmite la información ordenada que en este caso es esencial ya que se quiere ver una figura de caracteres en un archivo de texto

Otra ventaja significativa del uso de HTTP es su compatibilidad con herramientas comunes como navegadores web, lo que facilita tanto el desarrollo como la validación del sistema. Esta compatibilidad también permite que se visualicen los resultados sin requerir herramientas especializadas, lo que simplifica las pruebas durante las primeras etapas del proyecto.

Diseño de la simulación

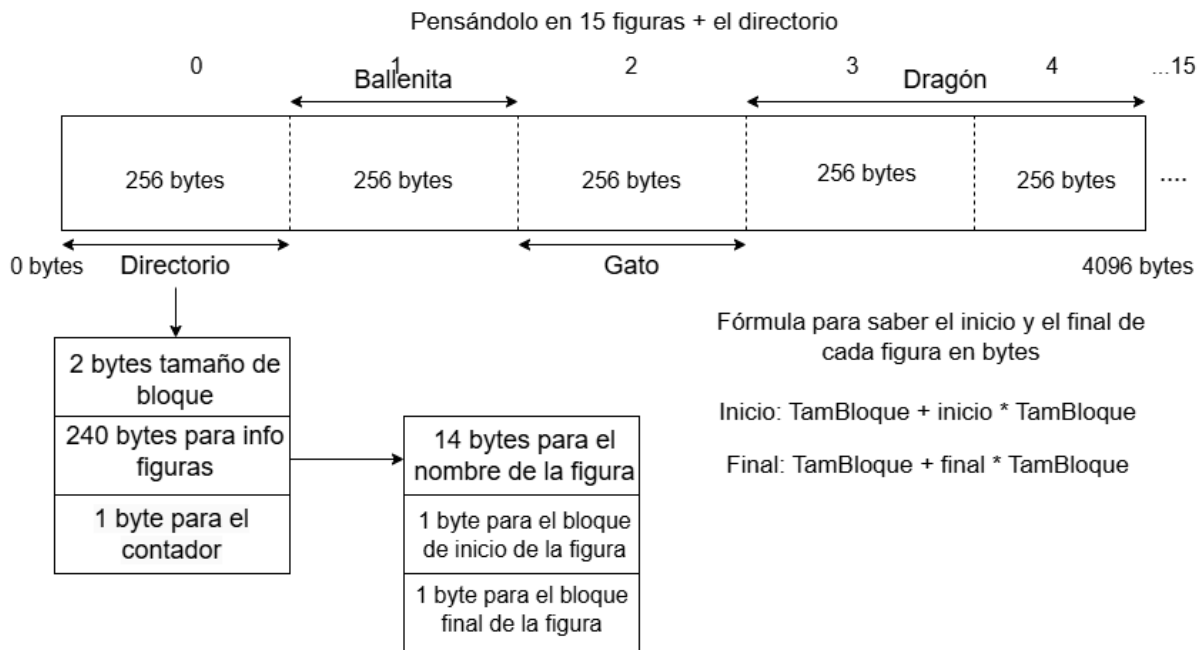
La simulación da la ilusión de trabajar con un servidor que posee las figuras ASCII en archivo de texto en un map y se accede a estas por su tipo de animal que es, además se tiene otra clase que es el intermediario que se le llama tenedor, este gestiona las solicitudes con la clase cliente, el cliente es el que solicita la figura del animal que quiere observar.

Todo esto se implementa en C++, donde se utilizan hilos, uno para la clase servidor y otro para la clase tenedor. Para simular el envío de solicitudes desde clientes, se utiliza una cola de mensajes (queue<Mensaje>) donde cada mensaje representa una solicitud de figura por parte del cliente. Esta cola funciona como un buzón, y está protegida por un pthread_mutex_t que garantiza el acceso sincronizado entre múltiples hilos, evitando condiciones de carrera.

El servidor cuenta con una estructura map<string, string> que simula el sistema de almacenamiento, donde cada figura está identificada por su nombre y su contenido correspondiente. Estas figuras se cargan en memoria desde un archivo de texto al momento de instanciar el servidor, lo cual permite una representación sencilla pero efectiva del sistema de archivos lógico.

Diseño del almacenamiento

Borrador almacenamiento



Para este diseño, se tiene pensado 15 figuras más el directorio, donde el peor caso sería que hayan 15 figuras de 256 bytes cada una, en este diseño, se tienen 16 bloques de 256 bytes, lo que daría un sistema de archivos de 4096 bytes, donde el directorio sería solo de un bloque de 256 bytes, el cual habría 2 bytes del tamaño de cada bloque, 240 bytes para información de las figuras y 1 byte para guardar un contador que vaya de 0 a 15 para representar cada bloque.

La información de las figuras en el directorio tendría 14 bytes para el nombre de cada figura, 1 byte para representar el inicio de la figura y 1 byte para representar el final de la figura, lo cual esta información multiplicado por las 15 figuras que se tienen pensado, darían los 240 bytes en total para la información de las figuras en el directorio. Por otro lado, a la hora de querer acceder a los bytes del inicio y final de una figura en específico, se podría realizar con las fórmulas dadas, donde toma el inicio o el final de la figura como contador de 1 byte (0 a 15) y lo multiplica por el tamaño del bloque, además de sumar también el tamaño del bloque para saltar el directorio.

Diseño de red

Se diseña una topología de red en Cisco Packet Tracer para representar la conexión entre los servidores de dibujos y el servidor-tenedor dentro de una red privada.

La simulación consiste en crear un entorno de red de capa 2, conformado por un switch y tres dispositivos finales (representados mediante PCs), los cuales simulan el servidor-tenedor y dos servidores de dibujos.

Cada dispositivo fue configurado con una dirección IPv4 perteneciente al rango 172.16.123.0/24, la cual es la red correspondiente a la infraestructura privada definida en el enunciado del proyecto. Esta asignación permite la comunicación entre todas las PCs a través del switch central.

Con este diseño se establece una base sobre la cual se implementarán los protocolos de comunicación entre los diferentes componentes del sistema de archivos.

