

# Armador de figuritas con piezas plásticas (Lego®)

Gabriel González Flores<sup>†</sup>, Sebastián Rodríguez Tencio<sup>†</sup>,  
Juan Aguilar Torres<sup>†</sup>, Andrés Zúñiga Méndez<sup>†</sup>

Contributing authors: [gabriel.gonzalezflores@ucr.ac.cr](mailto:gabriel.gonzalezflores@ucr.ac.cr);  
[sebastian.rodrigueztencio@ucr.ac.cr](mailto:sebastian.rodrigueztencio@ucr.ac.cr); [juan.aguilartorres@ucr.ac.cr](mailto:juan.aguilartorres@ucr.ac.cr);  
[andres.zunigamendez@ucr.ac.cr](mailto:andres.zunigamendez@ucr.ac.cr);

<sup>†</sup>Estos autores contribuyeron igualmente a este trabajo.

## Abstract

En este trabajo se implementa un programa que permite conectar a un cliente con un servidor de piezas, el cual le brinda la información necesaria para poder armar la figura. Y de igual manera, se trabaja en un posible protocolo para las siguientes etapas.

**Keywords:** lego, armador, piezas, protocolo

## 1 Introducción

En el ámbito de la construcción de figuras de Lego, es fundamental tener acceso a un inventario de piezas actualizado y confiable. Con este fin, en el presente trabajo, se ha diseñado un sistema de comunicación entre clientes y servidores que permitirá obtener un listado de piezas disponibles para construir una figura sencilla de Lego.

El sistema se estructura en dos etapas, en la primera los clientes interactuarán directamente con el servidor de piezas para obtener información sobre las figuras disponibles y las piezas necesarias para armarlas. En la segunda etapa, los clientes se comunicarán con servidores intermedios mediante el protocolo HTTP en un puerto TCP determinado.

El servidor de piezas es el encargado de proporcionar el inventario de piezas disponibles y las figuras que se pueden construir con ellas. Este servidor cuenta con una cantidad limitada de piezas, por lo que es importante asegurarse de que las piezas solicitadas estén disponibles.

El objetivo de este sistema es permitir a los clientes obtener información detallada sobre las piezas necesarias para construir una figura específica, incluyendo los nombres

de las piezas tal y como se identifican en el servidor. En caso de que no sea posible completar la figura, el servidor notificará al cliente con un error.

La primera entrega de esta evaluación consiste en la presentación de diversos elementos por parte de cada equipo de trabajo. En primer lugar, se realiza la entrega del cliente, es decir, la implementación del código que permitirá a los usuarios solicitar las piezas necesarias para armar una figura de Lego. Asimismo, se selecciona una figura sencilla y proporcionar el detalle de las piezas necesarias para su construcción. De esta manera, se podrán realizar pruebas y validar que el sistema está funcionando correctamente.

Además, se incluyen casos de prueba que permitan verificar el correcto funcionamiento del sistema en diferentes escenarios. Estos casos de prueba se publicarán en el repositorio y cubrirán diferentes situaciones que puedan presentarse.

Otro elemento importante que se entrega la definición del protocolo de comunicación para adicionar o eliminar servidores de piezas a los servidores intermedios, o viceversa. Esta definición se realiza de una manera clara, para asegurar que los equipos puedan interactuar de manera eficiente y sin errores. Posteriormente un protocolo será elegido para ser seguido por los distintos grupos de trabajo. Por último, se realiza un documento de presentación en el que se explique el funcionamiento del sistema desarrollado, el cuál será presentado en clase.

(Adicionalmente, en un documento también se valorará el trabajo en equipo e individual, y se realizará una evaluación del desempeño de cada miembro del equipo.)

## 2 Objetivo general

Crear un cliente que utilice el protocolo HTTPS para solicitar a un servidor una figura específica y obtener las piezas necesarias para construir el objeto solicitado por el cliente.

## 3 Objetivo específicos

Entre los objetivos específicos se plantean:

1. Diseñar un sistema de solicitud de figuras a través de una plataforma que garantice la seguridad de la transmisión de datos mediante el uso del protocolo HTTPS.
2. Desarrollar un programa capaz de devolver al usuario las piezas necesarias para armar la figura solicitada por este.
3. Brindar la idea de un posible protocolo a utilizar.

## 4 Metodología

En esta etapa se utilizó el lenguaje C++ en el editor de código fuente Visual Studio Code con la extensión de Live Share para lograr trabajar todos los autores del trabajo al mismo tiempo.

Como parte de los Sockets, es decir la petición que realiza un cliente a los servidores de piezas, se programaron con acceso seguro SSL en IPv4 con el fin de aumentar la seguridad entre la comunicación.

Por otro lado, se genera una solicitud HTTPS en la cual se conecta el Socket, que solicita una figura específica, con el servidor de piezas, este después de recibir la solicitud devuelve las piezas necesarias para construir la figura, por lo tanto lo que se realizó fue manipular el HTML que devuelve el servidor de piezas para mostrarle al cliente solo las piezas necesarias para construir la figura que se solicitó.

## 5 Resultados

Mediante el uso de sockets SSL se logró establecer apropiadamente la conexión estable al servidor de piezas. El cliente utiliza exitosamente las solicitudes HTTPS para traer, no sólo la lista de todas las figuras disponibles (que se le mostrarán al usuario), si no también traer las piezas requeridas para construir la figura que el usuario especificó en la línea de comandos, y mostrar en pantalla la cantidad de piezas necesarias para armar dicha figura.

Se plantea un posible protocolo el cuál se detalla de una mejor manera en la sección 5.1

### 5.1 Protocolo

Este sistema de servidores está diseñado para trabajar en conjunto y proporcionar al cliente las piezas necesarias para armar una figura. Para lograr esto, cada servidor de piezas tiene un "struct" que incluye su dirección IP y un arreglo con las figuras que contiene.

Cuando un servidor de piezas es iniciado, envía un mensaje de "broadcast" a todos los servidores intermedios. Este mensaje incluye el "struct" del servidor de piezas, el cual contiene información importante para que los servidores intermedios puedan localizar las piezas necesarias.

Al recibir el mensaje de broadcast, cada servidor intermedio almacena la información en un mapa con la estructura "(figura - ip, ip, ...)". De esta manera, los servidores intermedios tienen acceso a las direcciones IP correspondientes para obtener las piezas necesarias de cada figura.

Una vez que el servidor intermedio ha almacenado la información del servidor de piezas en el mapa, se mantiene en espera de conexiones de clientes. Cuando un cliente realiza una petición para obtener un conjunto de piezas para armar una figura, el servidor intermedio utiliza el mapa previamente creado para acceder a las direcciones IP correspondientes y obtener las piezas necesarias.

Además, se ha previsto la situación en la que un servidor intermedio es iniciado después de algún servidor de piezas. En ese caso, el servidor intermedio enviará un mensaje de "broadcast" a todos los servidores de piezas para solicitar su información. Si algún servidor de piezas está en línea, responderá al mensaje con su "struct" correspondiente.

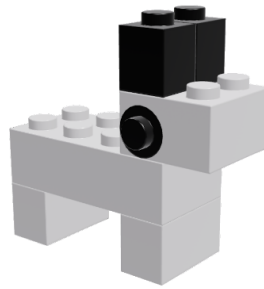
Esta medida asegura que el servidor intermedio tenga acceso a la información necesaria para poder realizar su función en el sistema. De esta manera, se garantiza que todos los servidores, tanto de piezas como intermedios, puedan colaborar y proporcionar al cliente las piezas necesarias para armar las figuras.

El protocolo utilizado para las conexiones entre servidores intermedios y de piezas es UDP, con los servidores intermedios escuchando en el puerto 4444 y los servidores de piezas en el puerto 5555. Por otro lado, la conexión del cliente con el servidor intermedio se realiza mediante el protocolo SSL IPV4, lo que garantiza una comunicación segura y confiable.

En caso de que el servidor intermedio no logre obtener todas las piezas necesarias para armar la figura solicitada por el cliente, éste recibirá un mensaje de error. Esto significa que los servidores intermedios han sido incapaces de localizar alguna de las piezas necesarias para la figura solicitada, es decir llegó al final de las IPs disponibles en el mapa.

En resumen, este sistema de servidores trabaja en conjunto para proporcionar al cliente las piezas necesarias para armar una figura. Los servidores de piezas envían información importante a través de mensajes de broadcast, que son almacenados por los servidores intermedios en un mapa. Luego, los servidores intermedios utilizan este mapa para obtener las piezas necesarias de las direcciones IP correspondientes. Todo el proceso se lleva a cabo mediante protocolos de comunicación seguros y confiables.

## 5.2 Figura



La figura propuesta es la de un dalmata, para la realización de dicha figura solo se requiere de 6 piezas como se detalla a continuación.

Dalmata:

2 brick 1x2 white

1 brick 2x4 white

1 brick 2x2 white

2 brick 1x1 black

Total de piezas: 6

## 6 Conclusión

A partir de los resultados, se pudo llegar a la conclusión de que el programa realizado es capaz de mostrar al cliente un listado de figuras, además el cliente puede escoger la figura que desee y se le devolverá un listado de piezas necesarias para armar dicha

figura. Para esto como se menciona se utiliza SSL en IPv4 con el cuál se puede lograr la conexión con el servidor de piezas, a de más de ser una opción bastante segura.

Por otro lado, se plantea un posible protocolo para las siguientes etapas del proyecto en el cuál se definen varios detalles importantes como por ejemplo el funcionamiento que debería de tener en un futuro el servidor de piezas y el servidor intermedio, además de la manera en la que este último servidor almacenaría la información brindada por el servidor de piezas. Por último, se brinda un ejemplo de una figura y el detalle de piezas respectivo.

## **7 Referencias**

Silberchatz, P. Galvin y G. Gadge, “Operating Systems Concepts”, 9na edición, John Wiley and Sons, 2013.