

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

ESCUELA DE CIENCIAS Y SISTEMAS

LABORATORIO ORGANIZACIÓN DE LENGUAJES Y COMPILADORES

SECCION: B

**PROYECTO 2**  
**SYSCOMPILER**  
**MANUAL TECNICO**

CARLOS ROBERTO RANGEL CASTILLO

201907636

# OBJETIVOS

## Objetivo General

Aplicar los conocimientos sobre la fase de análisis léxico y sintáctico de un compilador para la realización de un intérprete sencillo, con las funcionalidades principales para que sea funcional.

## Objetivos Específicos

1. Reforzar los conocimientos de análisis léxico y sintáctico para la creación de un lenguaje de programación.
2. Aplicar los conceptos de compiladores para implementar el proceso de interpretación de código de alto nivel.
3. Aplicar los conceptos de compiladores para analizar un lenguaje de programación y producir las salidas esperadas.
4. Aplicar la teoría de compiladores para la creación de soluciones de software.
5. Aplicar conceptos de contenedores para generar aplicaciones livianas.
6. Conocer más acerca de Docker y Docker-Compose.
7. Generar aplicaciones utilizando arquitecturas Cliente-Servidor.

## **DESCRIPCION GENERAL**

El curso de Organización de Lenguajes y Compiladores 1, ha puesto en marcha un nuevo proyecto, requerido por la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, que consiste en crear un lenguaje de programación para que los estudiantes, del curso de Introducción a la Programación y Computación 1, aprendan a programar y tener conocimiento de todas las generalidades de un lenguaje de programación. Cabe destacar, que este lenguaje será utilizado para generar sus primeras prácticas de laboratorio del curso antes mencionado.

Por lo tanto, a usted, que es estudiante del curso de Compiladores 1, se le encomienda realizar el proyecto llamado SysCompiler, dado sus altos conocimientos en temas de análisis léxico, sintáctico y semántico.

## ARQUITECTURA GENERAL DEL PROYECTO

Hoy en día, se ha dado gran importancia al uso de tecnologías de contenedores, lo que otorga como ventajas; rapidez en el despliegue y facilidad de mantenimiento en un servidor.

Para el presente proyecto, se le solicita manejar una arquitectura Cliente-Servidor, con el objetivo de que pueda separar los servicios administrados por el intérprete, de la aplicación cliente que se mostrará al usuario final.

De la misma manera, se solicita el uso de contenedores de Docker (de manera no obligatoria), además del uso de docker-compose para administrar y desplegar fácilmente los contenedores a través de un archivo de especificaciones .YAML.

A continuación, se encuentra el diagrama de la arquitectura a implementar en un entorno de Docker.

### Arquitectura Cliente-Servidor sin Docker

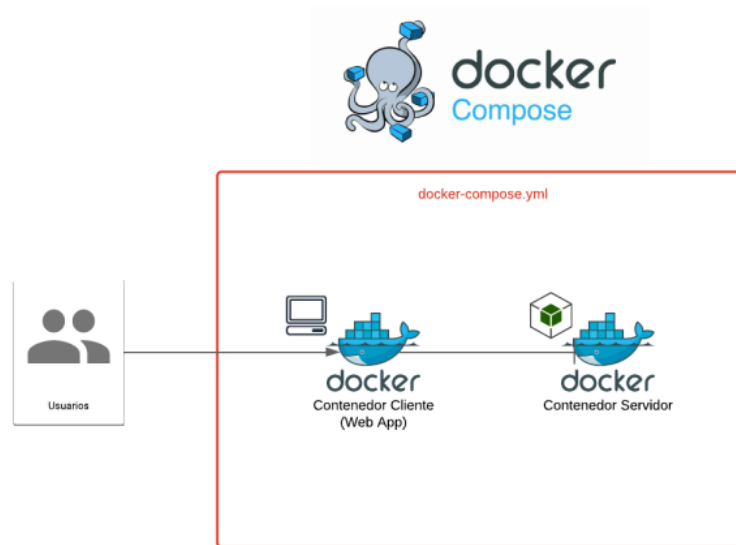
**Figura I:** Comunicación Cliente-Servidor sin Docker.



Fuente: Anónimo, tomado del enunciado del proyecto dos.

## Arquitectura Cliente-Servidor con Docker

**Figura II:** Comunicación Cliente-Servidor con Docker.



Fuente: Anónimo, tomado del enunciado del proyecto dos.