

Actividad 5 – Propuesta de Aplicativo con Arquitectura

Estefanía Toro Bañol

Arquitectura de Software / Corporación Universitaria Iberoamericana

Ingeniería de Software

Joaquín Sánchez

Octubre 2024

Gestión de Inventario usando la Arquitectura Cliente - Servidor

Para esta actividad se propone realizar un software aplicando una de las arquitecturas vistas en el transcurso del curso, por lo tanto, el sistema que se propone para esta ocasión es la de gestión de inventario para una venta de vehículos, la cual permita añadir, actualizar, eliminar y consultar vehículos en el sistema. La arquitectura elegida para desarrollar este sistema es la Arquitectura cliente/servidor, uno de los más conocidos, el cual está distribuido por dos componentes: el proveedor y el consumidor, lo que significa que el proveedor es el servidor que brinda una variedad de servicios los cuales son consumidos por el cliente.

Con esta arquitectura pueden existir varios clientes para solo un servidor el cual permite a los clientes conectarse a través de TCP (Protocolo de control de transmisión) permitiendo una comunicación continua y bidireccional, para que el cliente pueda enviar y recibir los datos del servidor y viceversa.

La estructura del proyecto consta de dos partes, las cuales son Cliente y Servidor, cada uno de forma independiente la cual permite centralizar la información y la separación de responsabilidades, el servidor será el único que tendrá acceso a los datos y servirá solo a los clientes.

Ventajas de usar La Arquitectura Cliente-Servidor

- **Divide Responsabilidades:** Permite implementar la lógica de negocio de forma separada del cliente
- **Fácil mantenimiento:** Cualquier cambio o actualización en el sistema, solo se necesita realizarse en el servidor
- **Escalabilidad:** Se pueden añadir más clientes fácilmente, los cuales podrán conectarse al servidor que permiten acceder a los recursos compartidos
- **Centralización de recursos:** Todos los datos y la lógica de la aplicación se gestionan en un único servidor, facilitando la seguridad, mantenimiento y administración ya que está centralizado en un solo lugar

Desventajas

- **Dependencia del Servidor:** El sistema depende completamente de la disponibilidad del servidor, por tanto, si el servidor falla o se desconecta, todos los clientes perderán acceso a la aplicación
- **Costos de Infraestructura:** Mantener un servidor centralizado implica costos asociados como hosting, mantenimiento, actualización, y seguridad del servidor, puesto que a medida que crece el número de clientes, se podría necesitar de una infraestructura más robusta
- **Carga en la red:** A medida que aumente los clientes que se conectan al servidor, el tráfico de la red aumenta, lo que puede provocar una disminución en el rendimiento

Herramientas utilizadas

- **Visual Studio Code:** Herramienta que me permite escribir y depurar código para el servidor y cliente

- **Python 3.11.9:** Lenguaje de programación, ya que tiende a ser muy versátil y que además tiene librerías que permiten la creación de aplicaciones para cliente – Servidor
- **Terminal o Consola:** La cual me permite ejecutar los scripts y me permite probar el servidor y el cliente
- **Librerías de Python como:**
 - **Socket:** Permite crear conexiones de red TCP entre el cliente y el servidor, la cual me permite enviar y recibir datos entre ambos
 - **Json:** Se utiliza para enviar y recibir información en formato JSON entre el cliente y Servidor, lo que facilita la interpretación de los mensajes

Documentos Técnicos

1. Requisitos Funcionales

Son las funcionalidades del sistema y son los que definen el funcionamiento de la misma desde el punto de vista del Cliente las cuales son:

- Visualizar los productos del Inventario
- Permite añadir nuevos productos
- Permite actualizar los productos existentes
- Permite eliminar los productos del inventario

2. Arquitectura Cliente Servidor

Explica la estructura de la aplicación, donde un servidor maneja el inventario y el cliente puede conectarse al servidor para realizar operaciones, usando Sockets TCP la cual permite la interacción a través de una conexión establecida entre ambos