Programación Orientada a Objetos

Trabajo Integrador

Ingeniería en Mecatrónica – Facultad de Ingeniería – Ciclo Lectivo 2024

Profesores

Esp. Ing. César Omar Aranda

Ing. Facundo Martín

Alumnos

Calderón Dal Pozzo, Joaquín. Legajo N° 13839

De Miguel Funes, Juan Marcos. Legajo N° 13673

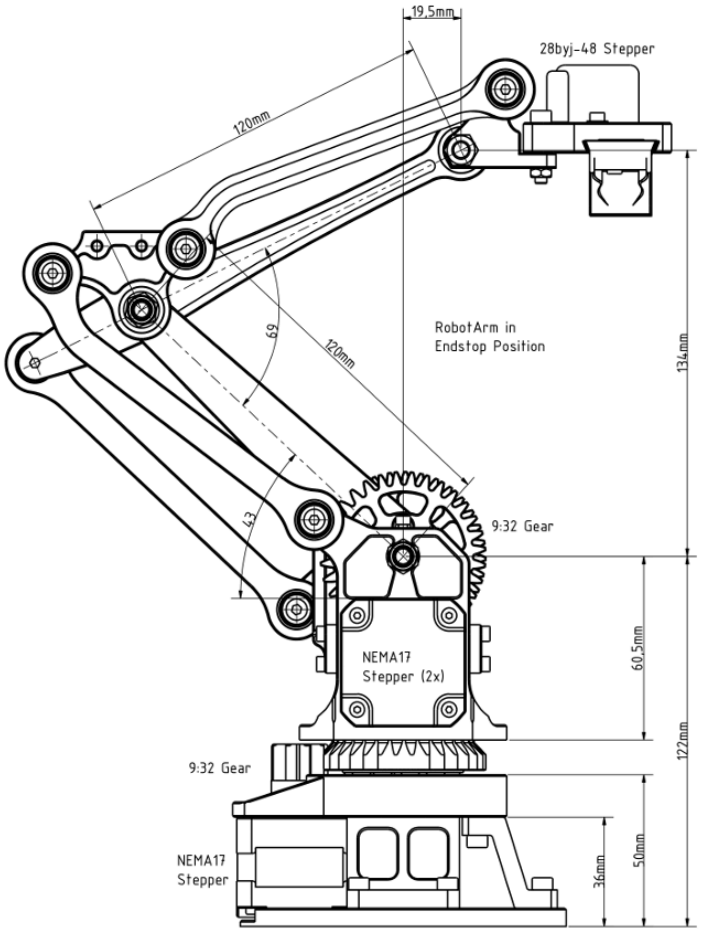
Huertas Coppo, Juan Francisco. Legajo N° 12620

Rayes Cano, Julián Andrés. Legajo N° 13256

Índice

Introducción

Este informe tiene por objetivo detallar un sistema bajo el paradigma de la Orientacion a Objetos para el control remoto de un robot de tres grados de libertad, implementado en los lenguajes Python y C++. Se tiene por objetivo aplicar los conceptos fundamentales del paradigma de la Programación Orientada a Objetos (POO) para la creación de una aplicación que permita manipular precisamente el robot en comandos de tipo G-Code. El robot anteriormente mencionado puede verse representado geométricamente en la Fig. 1.



El proyecto abarca el diseño de un servidor en Python y de un cliente en C++. La lógica de control del robot se encuentra implementada en el servidor, y éste proporciona una interfaz de acceso remoto RPC (Remote Procedure Call). Por otro lado, el cliente, desarrollado en C++ proporciona una interfaz de usuario que permite enviar órdenes al robot y monitorear su estado.

La estructura del proyecto sigue una arquitectura de tres capas: modelo, vista y controlador (MVC), repartiendo la lógica en componentes que gestionan datos, interfaz de usuario y flujo de control

respectivamente. Esto permite que la solución se pueda adaptar a futuras implementaciones con interfaces gráficas o ampliar su capacidad con agregados externos sin afectar la estabilidad del funcionamiento básico.

Además del control del robot, se incluye un sistema de registro de actividades, donde cada acción realizada es almacenada en un archivo CSV, permitiendo la trazabilidad de operaciones, así como un archivo de usuarios y contraseñas que asegura que sólo personal autorizado pueda controlar el robot.

Las principales funcionalidades abarcan desde la activación y desactivación de motores y del efector final hasta la ejecución de movimientos en modo manual y automático. En modo manual, el usuario envía comandos y en modo automático se cargan y ejecutan archivos en G-Code con las secuencias de trabajo, permitiendo automatizar tareas repetitivas.

A lo largo de este trabajo se detallará la aplicación de la solución escalable y adaptable para el control remoto de un robot.

Cuerpo

Conceptos

Arquitectura Cliente-Servidor

Este proyecto se basa en el modelo cliente-servidor, una arquitectura en la que el cliente solicita servicios al servidor, quien los ejecuta y responde. En este caso, el cliente, codificado en lenguaje C++ se conecta al servidor, codificado en lenguaje Python mediante XML-RPC para enviar comandos que controlen al robot

La importancia de esta arquitectura radica en la separación de responsabilidades, donde el cliente maneja la interfaz del usuario cliente y el servidor gestiona la lógica de control y comunicación con el hardware del robot.

Comunicación con XML-RPC

XML-RPC es un protocolo que utiliza XML para codificar llamadas a procedimientos y se comunica a través de HTTP. En este proyecto, el cliente, codificado en C++, llama métodos remotos del servidor en Python mediante XML-RPC para enviar los comandos.

El uso de este protocolo facilita la interoperabilidad entre ambas partes del proyecto, codificadas en distintos lenguajes.

Múltiples hilos

En este trabajo, el servidor implementa múltiples hilos (multithreading) para manejar la comunicación con el cliente y gestionar las operaciones del robot en paralelo. El servidor puede escuchar comandos mientras mantiene otros procesos activos, esencial para un funcionamiento fluido.

Comunicación Serial con el Hardware

La comunicación serial consiste en un método de transmisión de datos en que los mensajes, en bits, se envían secuencialmente por un sólo canal, lo que es común en microcontroladores.

Existe una clase "Controlador", la cual se comunica con el robot a través de un puerto serial, haciendo uso de la librería pySerial, para la comunicación e intercambio de información referente a cambios de posición, estado y accionamientos del robot.

Patrones de diseño y modularidad

El diseño del sistema sigue un patrón modularizado, con clases específicas para las tareas requeridas en cuanto a interfaces de usuario, comunicaciones, conexiones entre cliente y servidor, etc. La modularidad mencionada ofrece escalabilidad y mantenimiento en el sistema.

En cuanto a patrones de diseño se incluye el uso del Patrón Controlador para las clases del cliente y servidor y el Patrón Singleton, mediante el cual se restringe el servidor a una sola instancia.

Manejo de excepciones y Registro de actividades

Se gestionan los errores que pudieran tener lugar, lo que es esencial en aplicaciones, permitiendo conocer la ubicación de las falencias de código y hacer un correcto seguimiento y mantenimiento.

Para el registro de actividades existe una clase llamada "Log_de_trabajo" la cual registra las actividades del robot en un archivo CSV, incluyendo tanto las operaciones exitosas como las fallidas. Esto permite rastrear y diagnosticar errores, proporcionando herramientas adicionales para la seguridad y el mantenimiento.

Recursos Visuales y Esquemas

Se utiliza un diagrama de clases para representar la estructura y relaciones entre clases, un diagrama de secuencia para ilustrar un procedimiento completo tanto de cliente como de servidor y un diagrama de actividad para detallar el proceso desde el inicio hasta el final del programa.

Componentes de Software

Librería XmlRpc

Esta biblioteca de C++ se utiliza para implementar el protocolo XML-RPC, permitiendo asi ejecución remota de métodos del servidor desde el cliente. Esta librería utiliza XML para codificar las llamadas a métodos y HTTP para la comunicación. En este proyecto, es el puente de comunicación entre el cliente y el servidor.

Librería PySerial

Esta librería facilita la comunicación serial entre una aplicación de Python y dispositivos conectados en puertos serie. En este proyecto, se utiliza esta librería para la comunicación con el microcontrolador de la placa Arduino, enviando comandos y recibiendo información sobre el estado del robot.

Librería PyGcode

Esta herramienta se utiliza para trabajar con el G-code en Python, que corresponde al lenguaje que interpreta el robot.

Se utiliza para parsear, manipular y generar G-code. Permite leer líneas de G-code y descomponerlas, crear nuevas líneas de G-code

Descripción

Diagrama de Clases

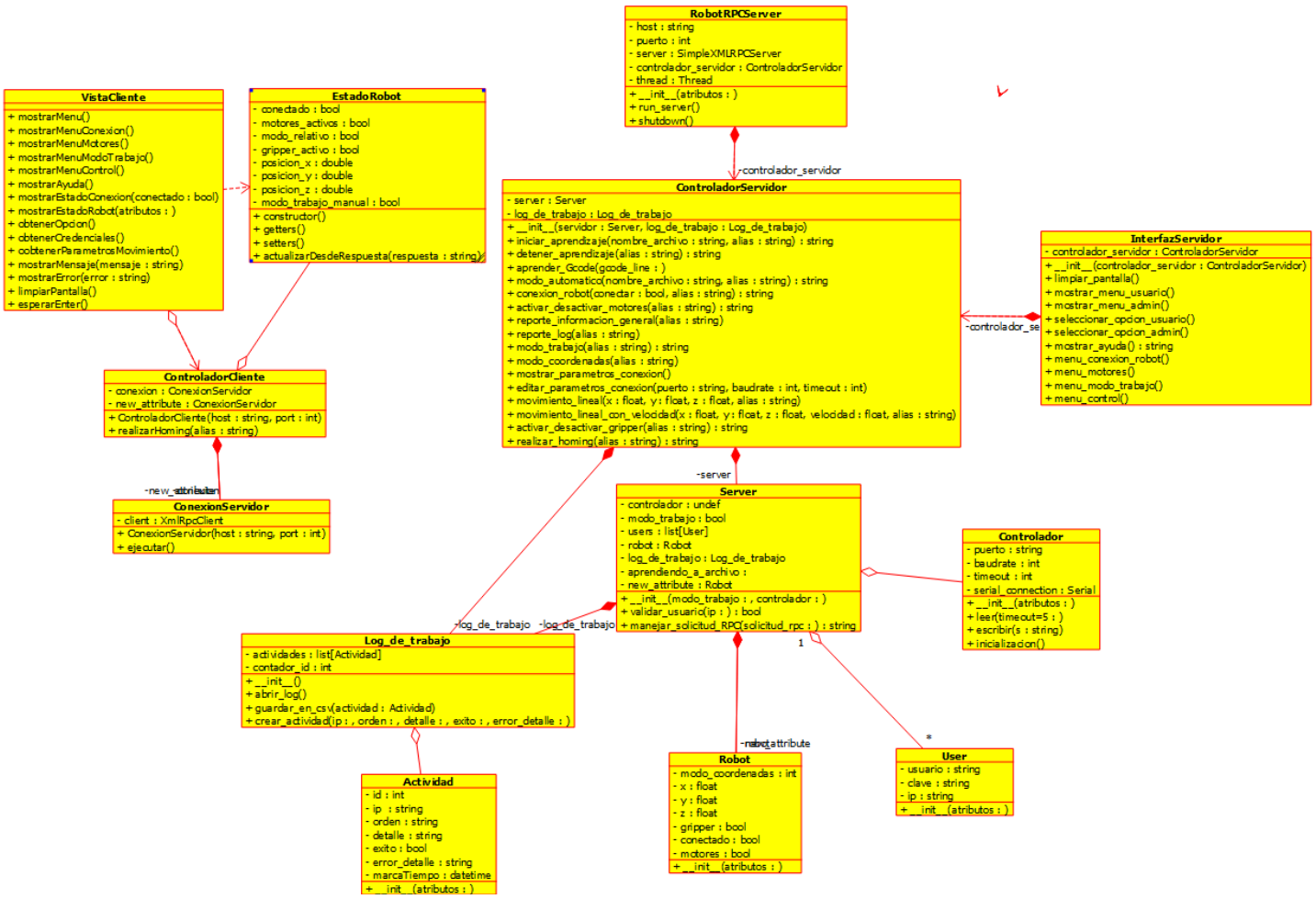


Diagrama de Secuencias

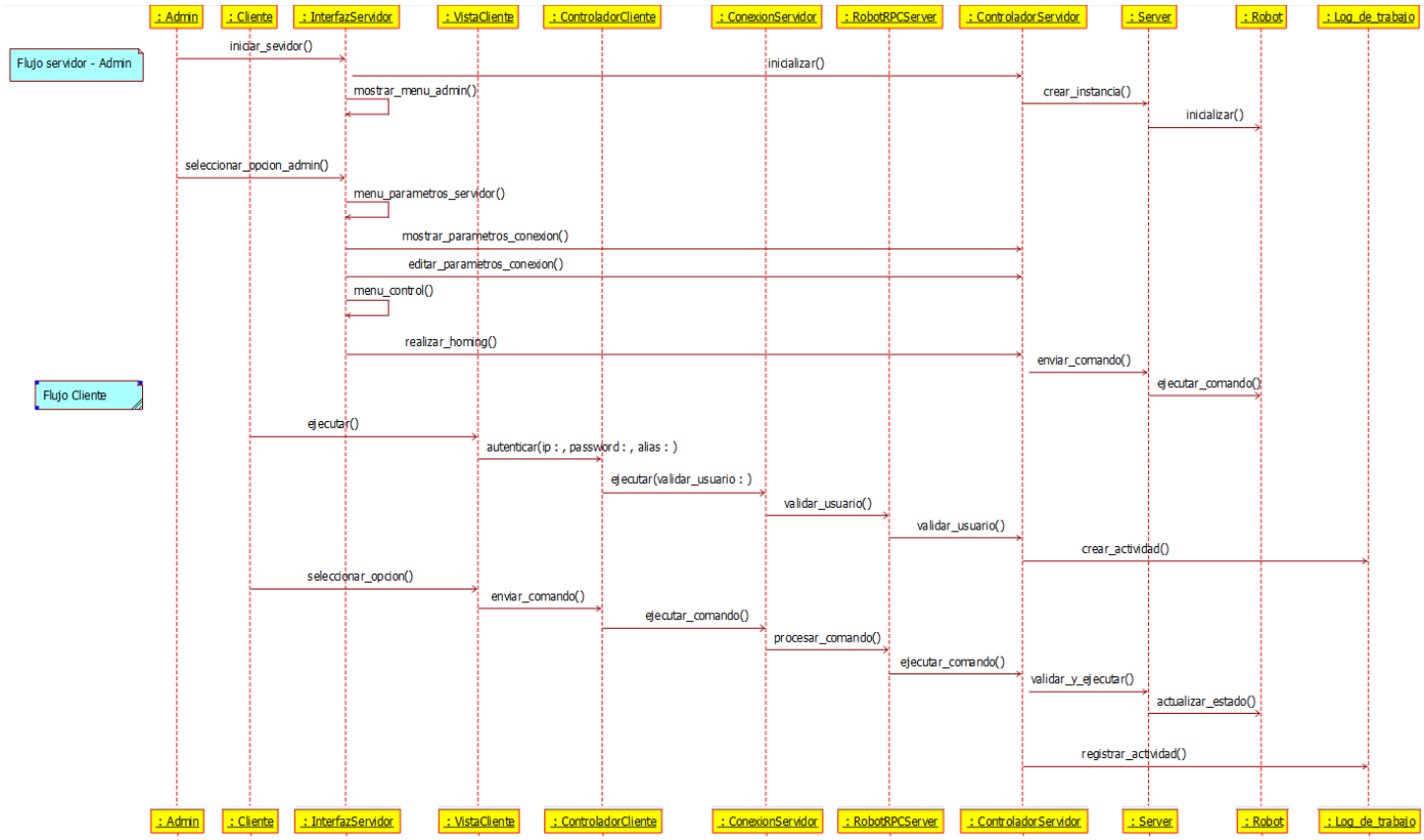
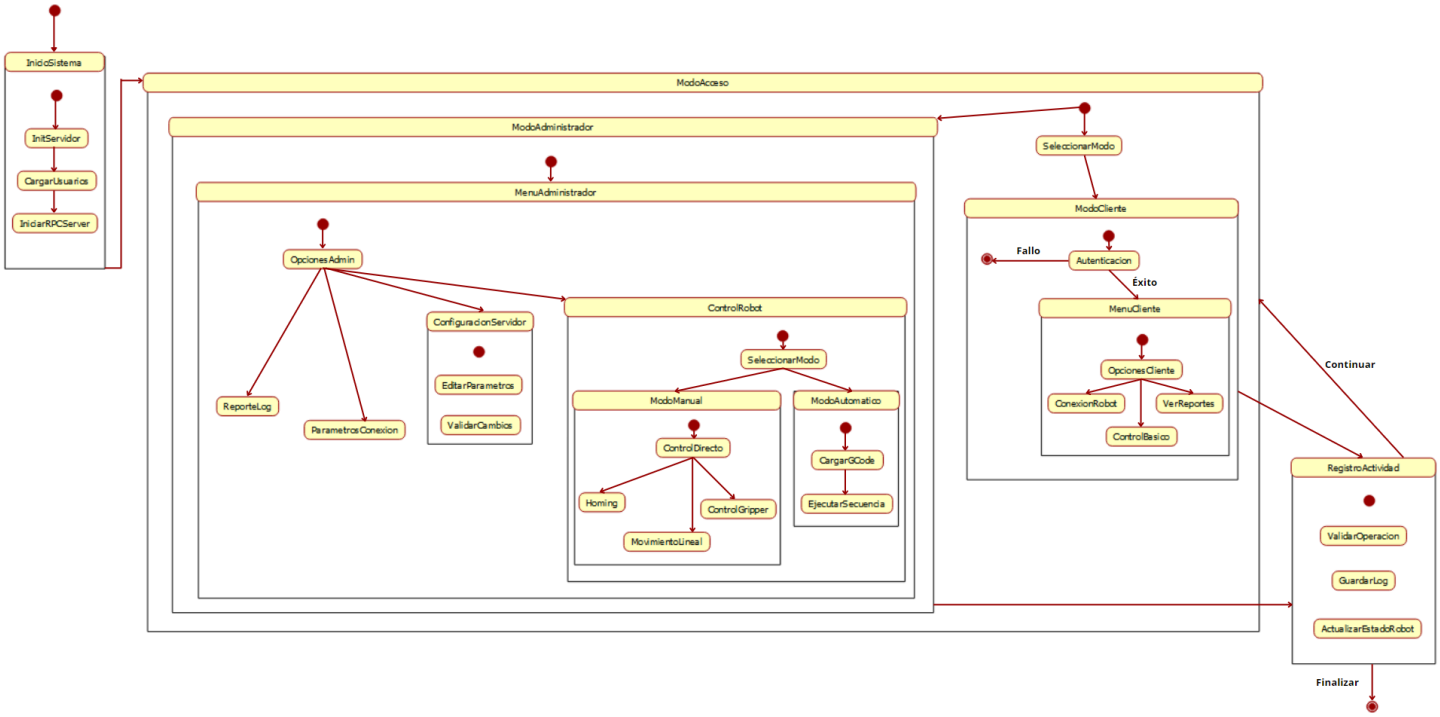


Diagrama de Actividad



Desarrollo

Capturas de pantalla

Módulo y reporte de Prueba Unitaria

Conclusiones y comentarios

Comentarios

Ventajas y desventajas

Extensiones

Referencias

Apuntes de cátedra: (XML-RPC), Persistencia, Manejo de excepciones

https://www.geeksforgeeks.org/client-server-model/

https://docs.python.org/3/library/threading.html

https://medium.com/@pysquad/explore-pyserial-serial-communication-libraries-e79b32a6dfe7

https://refactoring.guru/design-patterns/creational-patterns