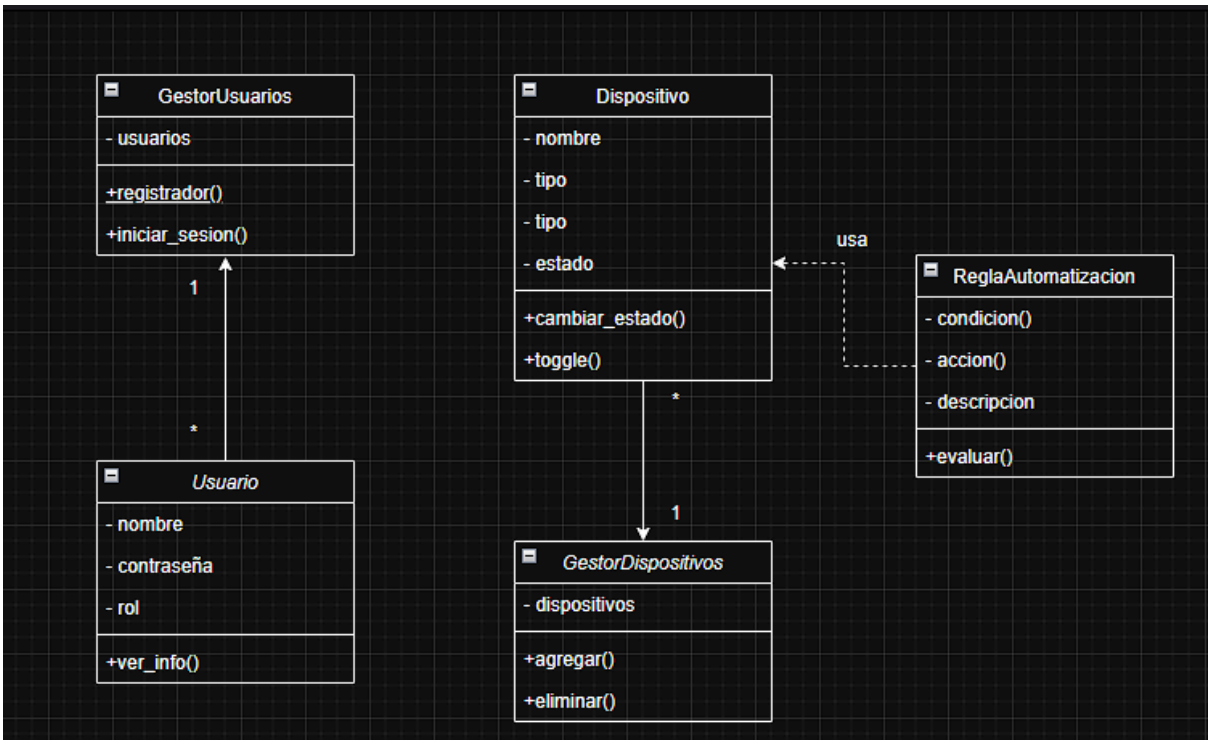
Evidencia 5

Punto 2: Programación

Grupo 18:
integrante: Fabricio Andres Cocconi Huenz DNI 46708260

Diagrama de clases



Justificacion de relaciones y principios de programacion orientada a objetos

Partiendo de las estructuras de datos del programa de consola y el modelo relacional de la evidencia 3 se diseño el siguiente diagrama de clases que representa las entidades principales del sistema como users devices y automation junto con sus relaciones aplicando principios de POO

Abstraccion
Cada clase representa una entidad concreta del sistema por ejemplo Usuario encapsula la info y funciones de una persona que usa el sistema mientras que Dispositivo representa cualquier aparato inteligente que puede ser gestionado esto permite simplificar el sistema enfocandose en lo esencial

Encapsulamiento
Los atributos estan protegidos y solo se modifican mediante metodos definidos esto ayuda a mantener la integridad de los datos y evitar errores por ejemplo el estado de un dispositivo solo cambia con cambiar_estado o toggle

Herencia
Aunque no se muestra en el diagrama se puede extender la clase Dispositivo para crear subclases como luces sensores o cafeteras lo que permite reutilizar codigo y agregar funciones sin alterar la clase base

Composicion
ReglaAutomatizacion trabaja con otras clases usando funciones que operan sobre dispositivos esto permite crear reglas complejas combinando objetos simples lo que da flexibilidad y modularidad

Agregacion
GestorUsuarios y GestorDispositivos manejan listas de objetos Usuario y Dispositivo respectivamente esta relacion indica que los gestores administran los objetos pero estos pueden existir por separado

Responsabilidad unica
Cada clase tiene una funcion clara GestorUsuarios administra usuarios GestorDispositivos se encarga de los dispositivos esto hace que el sistema sea mas facil de mantener y entender

Relaciones entre clases
Los gestores organizan y administran colecciones de objetos facilitando la gestion del sistema
ReglaAutomatizacion depende de los dispositivos para ejecutar acciones lo que permite automatizar tareas de forma eficiente

En resumen el diseño aplica principios de POO para lograr un sistema modular escalable y facil de mantener donde cada clase cumple un rol especifico y las relaciones estan bien definidas.