Actividad Integradora N°5

Programación I – Diseño de Clases y Principios de POO

Proyecto:

Sistema Smart Home – Diseño Orientado a Objetos

Integrantes del equipo:

- Rodríguez Facundo
- Altamirano Rocío
- Rodríguez Matías
- Córdoba Diego
- Lanfranco Julia
- Lanfranco Carolina

Materia:

Programación I

Docente:

Rojas Córsico Ivana, Gerlero Martín

Fecha de entrega:

Septiembre de 2025

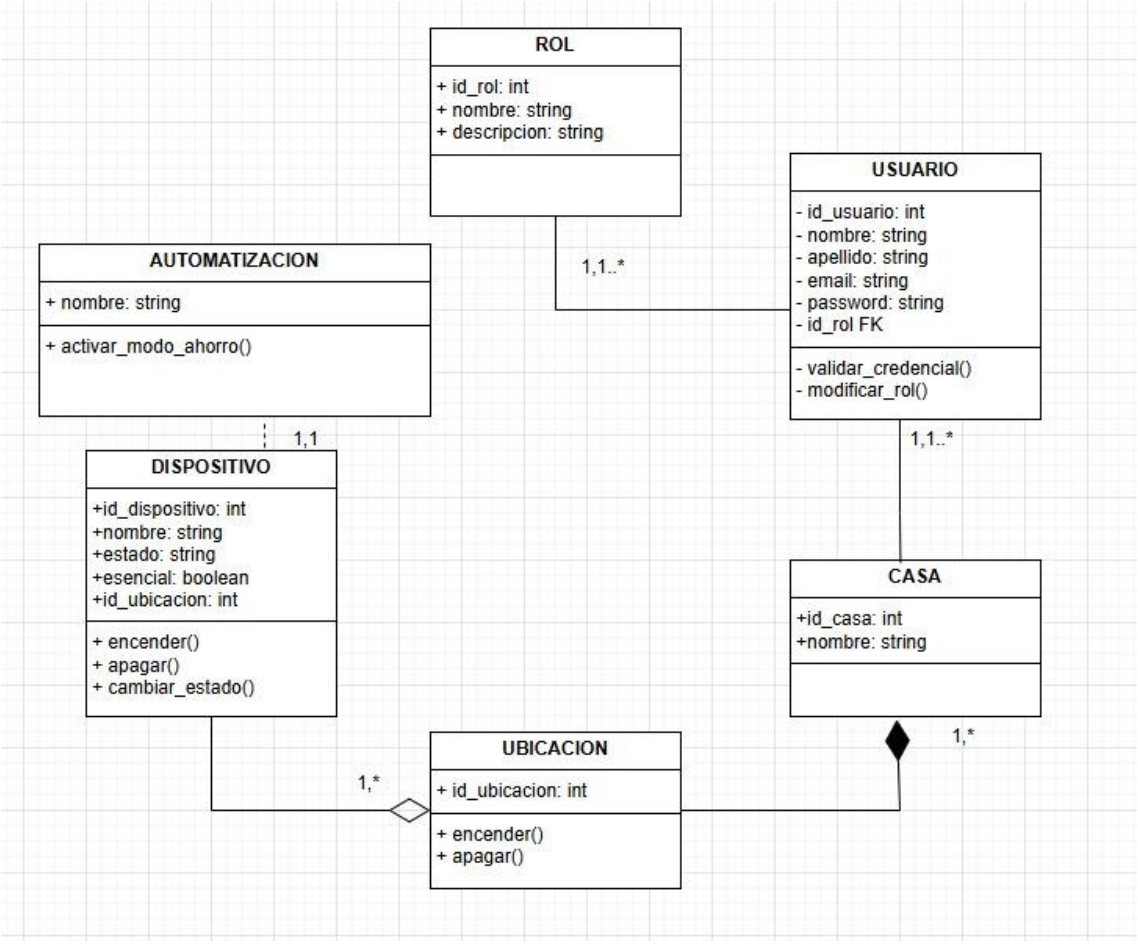
Contenido del documento:

- Diagrama de clases del sistema SmartHome
- Justificación de relaciones entre clases
- Aplicación de principios de Programación Orientada a Objetos (POO)

Introducción

Este documento presenta el diseño orientado a objetos del sistema SmartHome, incluyendo el diagrama de clases y la justificación de las relaciones entre entidades. Se aplican los principios fundamentales de la Programación Orientada a Objetos (POO) para lograr un modelo claro, modular y coherente con los requerimientos del sistema.

1. Diagrama de Clases



2. Justificación de Relaciones y Principios Aplicados

2.1 Abstracción

La clase USUARIO incluye únicamente los atributos necesarios para la gestión dentro del sistema: id_usuario, nombre, apellido, email, password e id_rol. Se omiten datos irrelevantes como fecha de nacimiento o número de teléfono, ya que no aportan

funcionalidad directa al modelo. Esta selección refleja el principio de **abstracción**, al centrarse en los aspectos esenciales del objeto.

2.2 Encapsulamiento

El atributo estado de la clase USUARIO representa información privada que no se accede directamente desde fuera de la clase. En su lugar, se utilizan métodos específicos para consultar o modificar su valor, lo que permite controlar el acceso y proteger la integridad del dato. Este enfoque aplica el principio de **encapsulamiento**, promoviendo seguridad y modularidad.

2.3 Composición

La relación entre CASA y UBICACION se modela como una **composición**, ya que una ubicación (por ejemplo, cocina o dormitorio) no puede existir de manera independiente sin estar asociada a una casa. Si una instancia de CASA se elimina, todas sus ubicaciones asociadas también deben eliminarse. Esta dependencia fuerte se representa gráficamente mediante un rombo negro en el diagrama.

2.4 Agregación

La relación entre UBICACION y DISPOSITIVO se considera una **agregación**. Un dispositivo puede existir sin estar asignado a una ubicación específica (por ejemplo, puede estar almacenado o en tránsito), pero una ubicación puede contener múltiples dispositivos. Esta relación flexible se representa con un rombo blanco, indicando una vinculación débil entre las clases.

2.5 Asociación y Cardinalidad

Se establecen las siguientes asociaciones entre clases, con sus respectivas cardinalidades:

- ROL → USUARIO: relación de uno a muchos (1 a 1..*). Un rol puede ser asignado a varios usuarios, pero cada usuario posee un único rol.
- USUARIO → CASA: relación de uno a muchos. Un usuario puede gestionar varias casas, pero cada casa pertenece a un solo usuario.
- CASA → UBICACION: relación de uno a muchos. Una casa puede contener múltiples ubicaciones internas.
- UBICACION → DISPOSITIVO: relación de uno a muchos. Una ubicación puede tener varios dispositivos asignados.
- AUTOMATIZACION → DISPOSITIVO: relación de uno a uno (1 a 1). Cada automatización afecta a un único dispositivo, y cada dispositivo está vinculado a una única automatización.

Estas asociaciones permiten modelar el comportamiento del sistema de forma precisa y escalable.

2.6 Dependencia

La clase AUTOMATIZACION presenta una **dependencia** funcional respecto a la clase DISPOSITIVO. El método activar_modo_ahorro() requiere acceder a los dispositivos para modificar su estado y apagar aquellos que no son esenciales. Esta dependencia se representa en el diagrama mediante una flecha punteada, indicando que AUTOMATIZACION utiliza objetos de DISPOSITIVO, aunque estos pueden existir independientemente.

2.7 Representación Gráfica

- La **composición** entre CASA y UBICACION se indica con un rombo negro, reflejando la dependencia total entre ambas clases.
- La **dependencia** entre AUTOMATIZACION y DISPOSITIVO se representa con una línea punteada, mostrando que AUTOMATIZACION utiliza instancias de DISPOSITIVO sin poseerlas.