

Justificación

Clases y responsabilidades

- **Usuario:** Representa a cada persona registrada en el sistema. Sus atributos (`_nombre`, `_contraseña`, `_es_admin`) permiten diferenciar administradores de usuarios estándar. Sus métodos (`iniciar_sesion()`, `registrarse()`, `ver_datos()`) encapsulan la lógica de autenticación y gestión de datos personales.
- **Dispositivo:** Modela los aparatos del hogar inteligente. Incluye atributos (`_nombre`, `_estado`, `_esencial`) y métodos (`encender()`, `apagar()`, `modificar_estado()`), que representan acciones reales del dominio.
- **Automatizacion:** Representa reglas o acciones automáticas aplicadas sobre dispositivos. Contiene atributos (`_nombre`, `_tipo`, `_activa`) y método principal `ejecutar()`.
- **SmartHome:** Clase central que administra colecciones de usuarios, dispositivos y automatizaciones. Sus métodos (`agregar_usuario()`, `agregar_dispositivo()`, `ejecutar_automatizacion()`) permiten gestionar y coordinar las demás entidades.

RELACIONES Y CARDINALIDADES

- **Agregación** (SmartHome —○→ Usuario/Dispositivo): SmartHome tiene usuarios y dispositivos (cardinalidad 1..*), pero pueden existir independientemente.
- **Composición** (SmartHome —◆→ Automatizacion): SmartHome está compuesto por automatizaciones (cardinalidad 1..*). Si se destruye SmartHome, se destruyen sus automatizaciones.
- **Asociación/Dependencia** (Automatizacion —→ Dispositivo): Las automatizaciones usan dispositivos para ejecutarse (cardinalidad ..).

Principios de POO aplicados

- **Abstracción:** Cada clase representa un concepto clave del sistema domótico, ocultando detalles innecesarios.
- **Encapsulamiento:** Los datos están protegidos (atributos privados `_`) y solo se manipulan mediante métodos públicos (`+`), asegurando coherencia.
- **Agregación:** SmartHome contiene usuarios y dispositivos que mantienen existencia independiente.
- **Composición:** SmartHome se compone de automatizaciones con dependencia de las otras partes.
- **Herencia** (a futuro): se podría especializar Usuario en UsuarioAdmin o UsuarioEstandar, reutilizando atributos y comportamientos.