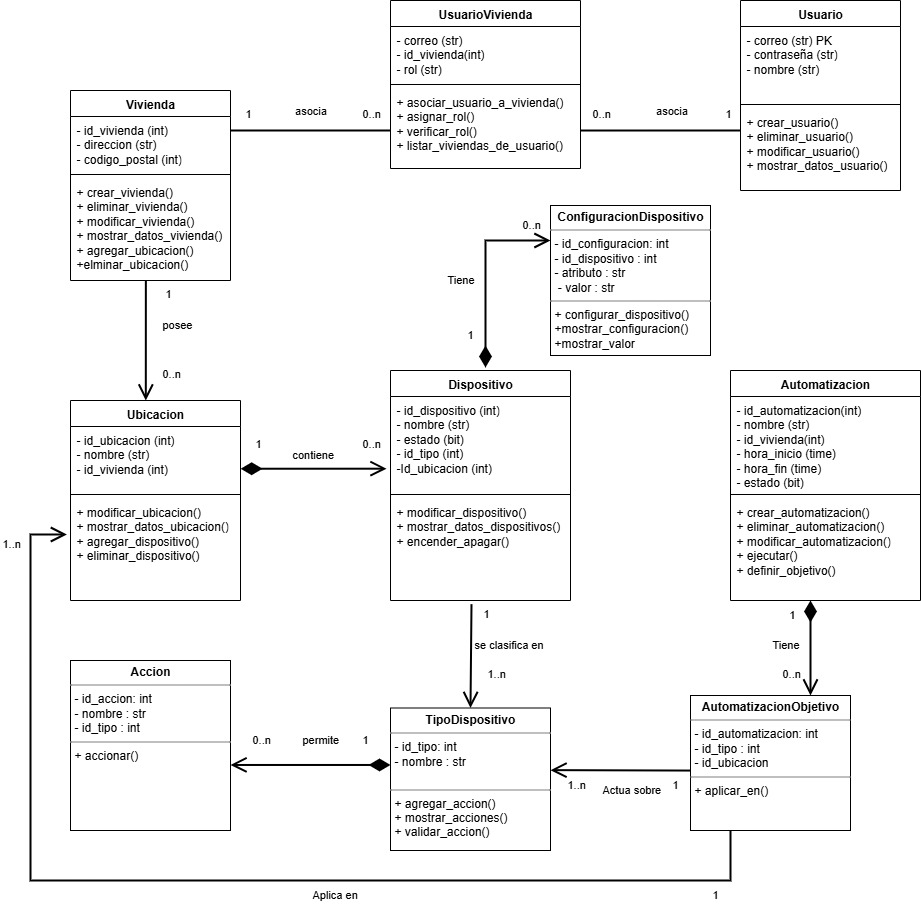
**Diagrama de clases UML**

****

**Análisis del diseño UML (clases y relaciones)**

**Usuario – UsuarioVivienda – Vivienda**  
**Asociación** (bidireccional), no composición porque ninguno depende de la existencia del otro.

usuarioVivienda es una entidad que representa una tabla intermedia utilizada para asociar usuarios a viviendas (un usuario o muchos pueden pertenecer a una o muchas casas) y definir que rol poseen en cada una.

**Vivienda – Ubicación – Dispositivo**

**Composición**: una Vivienda compone sus Ubicaciones, y una Ubicación compone sus Dispositivos.

Si eliminas la Vivienda, también desaparecen sus ubicaciones (ambientes).

Si eliminas una ubicación, también eliminas los dispositivos que estén dentro.

**Dispositivo – TipoDispositivo**  
**Asociación**: El dispositivo “sabe” qué tipo es, pero un TipoDispositivo existe sin necesidad de instancias.

No es composición ni agregación porque:

* El TipoDispositivo existe independientemente de los dispositivos (ejemplo: podes definir un nuevo tipo “Cafetera” aunque todavía no tengas ninguna cafetera registrada).
* Por lo tanto, nosedestruye si eliminás un Dispositivo.

**Dispositivo – ConfiguraciónDispositivo**  
**Composición**: la configuración no existe sin el dispositivo.

Ejemplo: “temperatura = 24°C” no tiene valor si no está ligado a un dispositivo concreto (ej: Aire acondicionado del living).

**TipoDispositivo – Acción**  
**Composición**: las acciones solo existen en función de un tipo (ej: “Subir temperatura” no existe sin un “Aire”).

Si eliminás el TipoDispositivo también desaparecen sus acciones (encender, subir temperatura…).

**Automatización – AutomatizacionObjetivo**  
**Composición:** AutomatizacionObjetivo es un detalle de implementación de la Automatización. No tiene sentido sin ella (no podés tener un “objetivo” que no pertenezca a ninguna automatización).

Automatización (todo) compone sus AutomatizacionObjetivo (partes).

### **AutomatizacionObjetivo – Ubicacion**

**Asociación simple:** Una **ubicación** existe sin necesidad de una automatización. Una automatización puede ejecutarse en varias ubicaciones distintas.

### **AutomatizacionObjetivo – TipoDispositivo**

**Asociación simple:** Un tipo de dispositivo existe independientemente de la automatización.Una automatización puede aplicar a varios tipos de dispositivos distintos.

**Vivienda – Automatizacion**

**Composición:** Las automatizaciones le pertenecen a una vivienda especifica

# **Justificación del diseño de clases y principios aplicados**

## 1. Principios de la Programación Orientada a Objetos

* **Abstracción:**  
  Se identificaron las entidades principales del dominio y se modelaron como clases que encapsulan sus atributos y comportamientos. Esto permite representar el sistema de domótica de manera simplificada y enfocada en lo esencial.
* **Encapsulamiento:**  
  Cada clase agrupa datos y operaciones relacionadas. Por ejemplo, la clase **Dispositivo** gestiona su propio estado (encender/apagar/configurar), sin exponer directamente sus atributos internos. Esto mejora la seguridad y el mantenimiento del código.
* **Herencia:**  
  En este caso no se implementaron jerarquías de herencia porque el modelo no lo requiere directamente.

Sin embargo, es extensible: a futuro, podrían definirse clases hijas de **Dispositivo** (ej. Luz, AireAcondicionado, Parlante) que hereden atributos y redefinan métodos de configuración o acciones.

* **Composición:**  
  Se aplicó cuando una clase no puede existir sin otra. Ejemplos:
  + Una **Vivienda** **compone** sus **Ubicaciones**.
  + Una **Ubicacion** **compone** sus **Dispositivos**.
  + Un **Dispositivo** **compone** sus **Configuraciones**.
  + Una **Automatizacion** **compone** sus **Objetivos**.  
    En todos estos casos, si se elimina el objeto contenedor, también se eliminan sus partes.
* **Agregación:**  
  No se usó explícitamente, porque ninguna relación es de "todo–parte débil". Ejemplo: un **TipoDispositivo** no “posee” sus **Dispositivos**, sino que solo se asocia con ellos.
* **Asociación:**  
  Se aplicó en relaciones donde los objetos existen de forma independiente, pero necesitan conocerse:
  + **Usuario** con **Vivienda** mediante la clase intermedia **UsuarioVivienda**.
  + **AutomatizacionObjetivo** con **Ubicacion** y con **TipoDispositivo**.
  + **Dispositivo** con **TipoDispositivo**.

## 2. Justificación de las relaciones del diagrama

**Usuario ↔ UsuarioVivienda ↔ Vivienda (Asociación con clase intermedia):**  
Se creó la clase intermedia porque la relación es muchos a muchos y tiene un atributo propio (**rol**). No es composición, ya que tanto Usuario como Vivienda pueden existir sin la relación.

**Vivienda ↔ Ubicación (Composición):**  
Una ubicación no tiene sentido sin una vivienda asociada. Por eso, la eliminación de la Vivienda implica la eliminación de todas sus Ubicaciones.

**Ubicación ↔ Dispositivo (Composición):**  
Un dispositivo siempre pertenece a una ubicación. Si se borra la Ubicación, sus dispositivos dejan de existir en el sistema.

**Dispositivo ↔ ConfiguraciónDispositivo (Composición):**  
La configuración depende totalmente del Dispositivo. Una configuración aislada (ej: “temperatura = 24°”) no tiene sentido sin el aparato al que pertenece.

**Dispositivo ↔ TipoDispositivo (Asociación simple):**  
Un dispositivo conoce a qué tipo pertenece. No se modeló como agregación porque un tipo de dispositivo (ej: “Aire acondicionado”) existe independientemente de las instancias concretas.

**TipoDispositivo ↔ Acción (Composición):**  
Las acciones posibles dependen totalmente del tipo (ej: “subir temperatura” solo es posible si existe “Aire acondicionado”).

**Automatización ↔ AutomatizacionObjetivo (Composición):**  
Los objetivos forman parte integral de la automatización; no existen de manera independiente.

**AutomatizacionObjetivo ↔ Ubicacion / TipoDispositivo (Asociación simple):**  
El objetivo necesita referenciar ubicaciones y tipos de dispositivo, pero estos objetos existen independientemente de la automatización.

**Aclaracion:**

**Automatización y Vivienda**  
En el modelo, una Automatización está asociada a una Vivienda específica porque sus Objetivos se definen sobre Ubicaciones que pertenecen a esa vivienda.  
No se representa como una composición directa en el diagrama de clases, ya que la dependencia se da de forma **indirecta** a través de la relación con Ubicación y AutomatizacionObjetivo.  
Sin embargo, desde el punto de vista del dominio, **una Automatización no tiene sentido fuera del contexto de una Vivienda**, por lo que conceptualmente puede considerarse parte de ella.