

**INSTITUTO SUPERIOR POLITÉCNICO DE CÓRDOBA**

**TECNICATURA SUPERIOR EN DESARROLLO DE SOFTWARE**

**Módulo Programador**

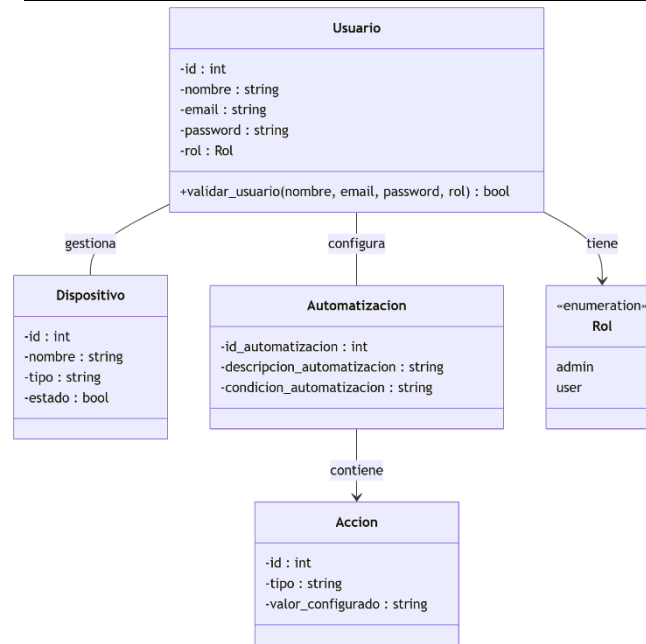
**Actividad Integradora N°5**

---

## **INTEGRANTES**

- 42258135 Víctor Andrés Bianchi Núñez
- 42383964 Federico David Udovich
- 35257982 Federico Martin Pierrestegui
- 36223373 Gabriel Alejandro Farias

## Diagrama de Clase de SmartHome Solutions



### Descripción general del diagrama

El diagrama de clases representa la estructura principal del sistema de gestión de hogar inteligente. Muestra las clases que intervienen en el control y automatización de dispositivos, junto con sus atributos y relaciones. Este modelo permite visualizar cómo los usuarios interactúan con los distintos componentes del sistema.

---

### Descripción de las clases y relaciones

#### *Usuario*

Representa a las personas que utilizan el sistema.

Contiene los datos básicos: id, nombre, mail, password y un rol definido por el tipo enumerado **Rol** (puede ser *admin* o *user*).

Incluye el método `validar_usuario()` que se encarga de comprobar la validez de los datos ingresados, asegurando que la información registrada sea correcta y coherente.

#### **Relaciones:**

- **Usuario – Rol (tiene):** cada usuario posee un rol determinado, definido por el enum Rol.
- **Usuario – Dispositivo (gestiona):** un usuario puede gestionar varios dispositivos dentro del sistema. Esta relación es de asociación, ya que los dispositivos pueden existir de forma independiente y ser reasignados a otros usuarios.
- **Usuario – Automatización (configura):** un usuario puede crear múltiples automatizaciones personalizadas que definan rutinas o comportamientos automáticos.

---

### *Dispositivo*

Representa los dispositivos IoT del hogar, como luces, sensores, termostatos, enchufes inteligentes o cámaras.

Incluye atributos básicos como identificador, nombre, tipo y estado (encendido o apagado).

Esta clase modela los elementos físicos que pueden ser controlados y monitoreados dentro del sistema.

---

### *Automatización*

Permite crear rutinas o escenarios automáticos que definen comportamientos programados del sistema.

Contiene una descripción y una condición que determina cuándo debe ejecutarse (por ejemplo, “hora = 20:00” o “temperatura > 30”).

Los usuarios configuran estas automatizaciones para que el sistema responda de forma inteligente ante determinadas situaciones.

#### **Relación:**

- **Automatización – Acción (contiene):** esta relación es de composición, ya que las acciones dependen completamente de la automatización que las contiene. Si una automatización se elimina, todas las acciones asociadas también se eliminan.

---

### *Acción*

Representa una operación específica que se ejecuta dentro de una automatización.

Sus atributos son el tipo de acción (por ejemplo, encender, apagar, notificar o cambiar\_estado) y un valor configurado que define los detalles de la ejecución.

Cada acción está vinculada directamente a una automatización, lo que refleja la lógica de ejecución secuencial del sistema.

---

### *Rol*

Es una enumeración (<>) que define los posibles tipos de usuario del sistema: *admin* o *user*.

Permite diferenciar los niveles de acceso y las responsabilidades dentro de la plataforma.