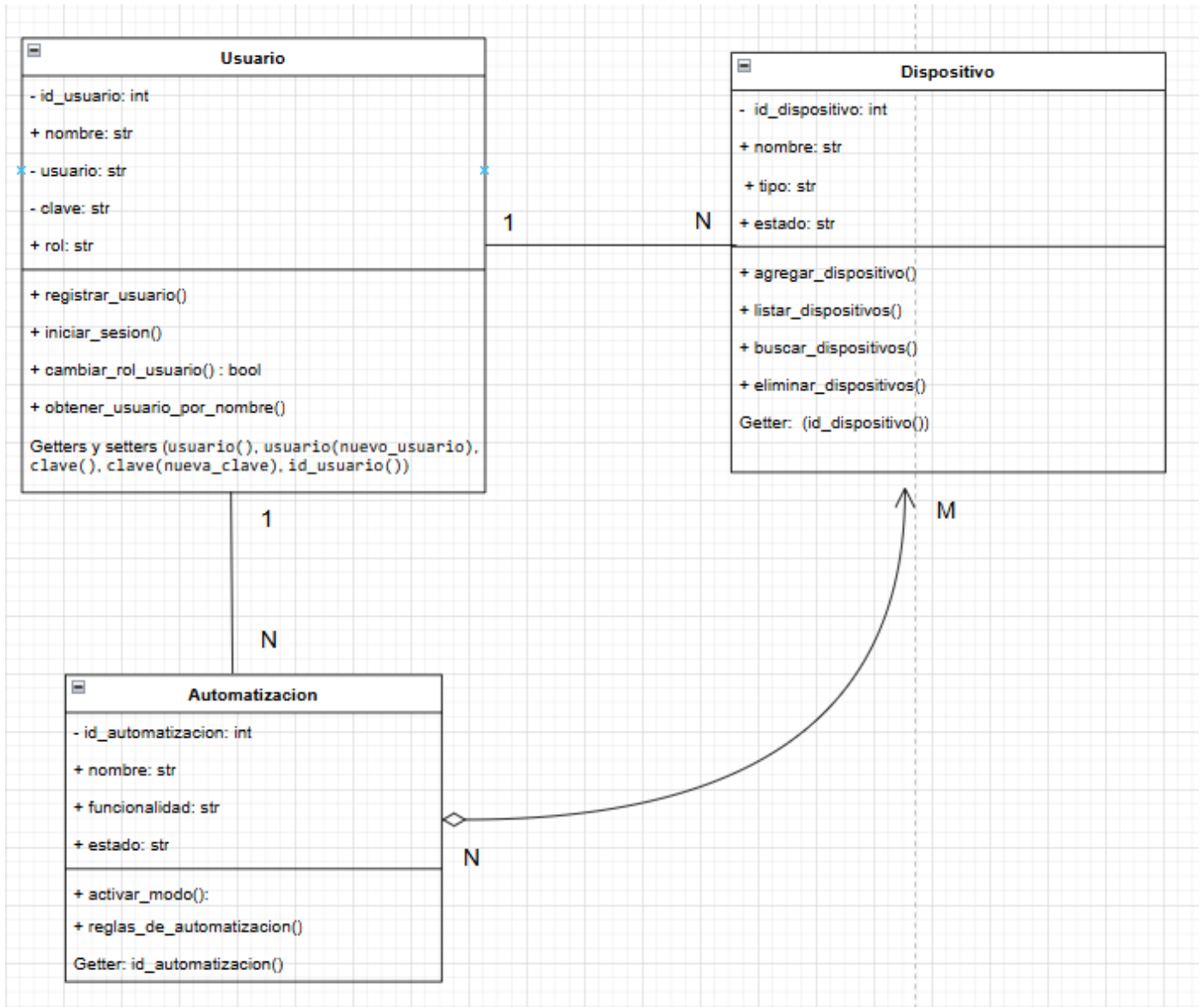


# Diagrama de Clases



## Relaciones entre clases

### 1. Usuario ↔ Dispositivo

- **Tipo:** Asociación
- **Cardinalidad:** 1 Usuario → n Dispositivos
- **Justificación:** Cada usuario puede gestionar múltiples dispositivos, pero un dispositivo está asociado a un solo usuario. Se representa con línea continua y flecha desde Usuario hacia Dispositivo, indicando **navegabilidad**.
-

## 2. Usuario ↔ Automatización

- **Tipo:** Asociación
- **Cardinalidad:** 1 Usuario → n Automatizaciones
- **Justificación:** Los usuarios pueden activar o configurar varias automatizaciones; la flecha apunta del Usuario hacia Automatización para mostrar que **el control lo tiene el usuario**.

## 3. Automatización ↔ Dispositivo

- **Tipo:** Agregación
- **Cardinalidad:** n ↔ m
- **Justificación:** Una automatización contiene referencias a múltiples dispositivos y un dispositivo puede estar en varias automatizaciones. Los dispositivos existen independientemente de la automatización, por eso se usa agregación (rombo vacío en el extremo de Automatización).

## Principios de POO aplicados

### 1. Abstracción:

- Cada clase representa un concepto real del sistema (Usuario, Dispositivo, Automatización), ocultando la complejidad interna de la gestión de datos y operaciones.

### 2. Encapsulamiento:

- Los atributos de las clases se pueden definir como privados o protegidos en Python (`_atributo` o `__atributo`) y se acceden mediante métodos públicos, protegiendo la integridad de los datos.

### 3. Agregación:

- La relación Automatización ↔ Dispositivo refleja agregación: Automatización “usa” varios dispositivos, pero los dispositivos **viven independientemente**.

### 4. Asociación:

- Las relaciones Usuario ↔ Dispositivo y Usuario ↔ Automatización son asociaciones simples que muestran navegación y control.