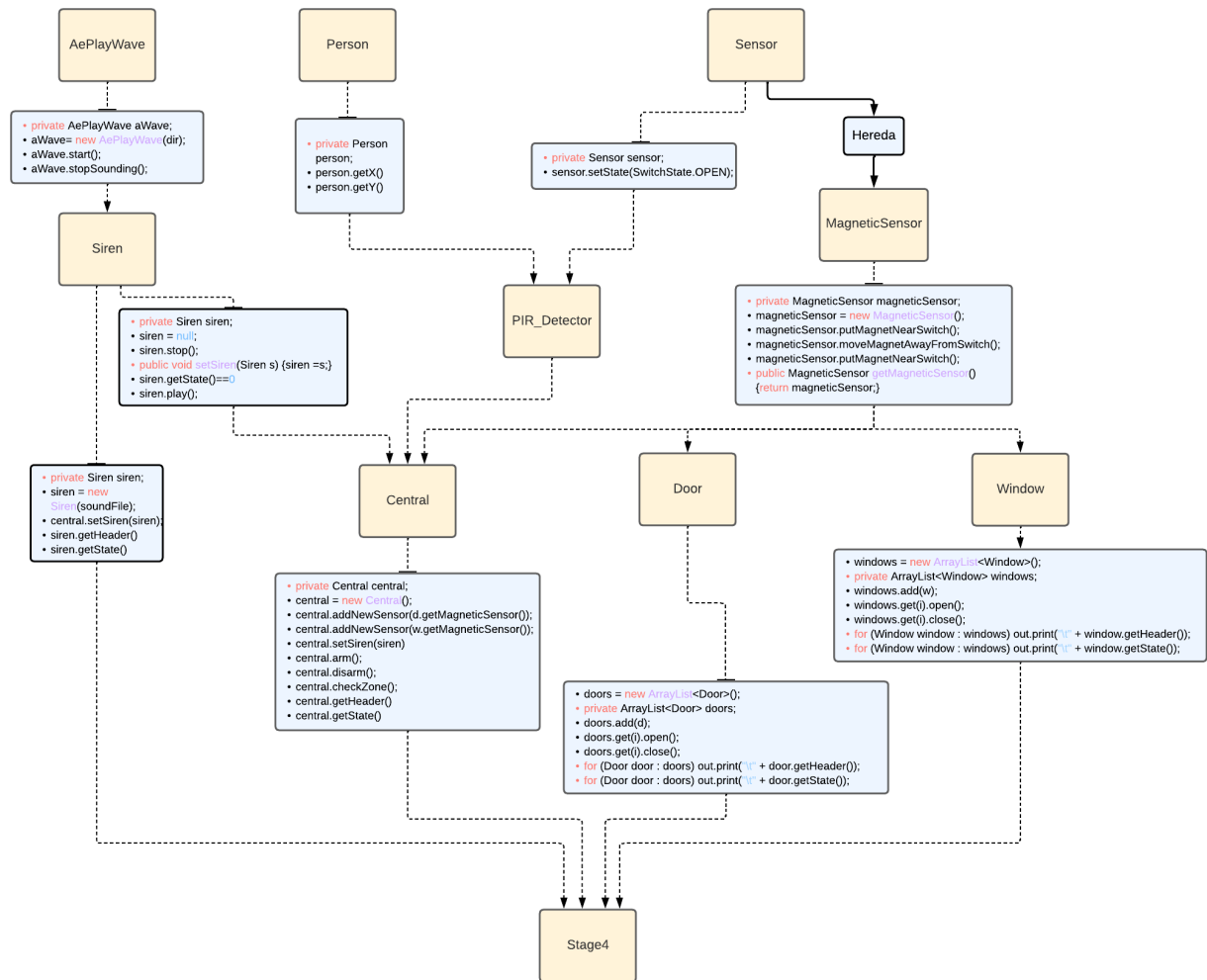


## **Documentación:**

- **AePlayWave:** Clase por defecto. Utilizada para reproducir sonidos a la hora de hacer funcionar el programa.
- **Siren:** Clase que representa una alarma. Su principal función es la de poseer métodos para cambiar el estado del objeto de encendido a apagado o viceversa, así, reproduciendo un archivo de audio. (Se utilizan métodos y/o atributos de: AePlayWave).
- **Person:** Clase que representa una persona. Permite mover y conocer la posición en el plano del objeto con coordenadas x e y.
- **Sensor:** Clase que representa los atributos de un sensor. Cuenta con métodos para activar o desactivar el estado del objeto (Abierto/Cerrado).
  - **MagneticSensor:** Clase que representa un sensor magnético. Contiene información de los estados de un sensor magnético y si este está activo o no. (Hereda de la clase: Sensor).
- **PIR\_Detector:** Clase que representa un sensor de movimiento. Verifica si una persona se encuentra dentro de un área específica propia del objeto. (Se utilizan métodos y/o atributos de: Person, Sensor).
- **Central:** Clase que representa una central de información. Aquí se recopilan y organizan datos para activar o desactivar alarmas y sensores. (Se utilizan métodos y/o atributos de: Siren, PIR\_Detector, MagneticSensor).
- **Door:** Clase que representa una puerta. Se encarga de la creación del objeto y el cambio de su estado (Abierto/Cerrado). (Se utilizan métodos y/o atributos de: MagneticSensor).
- **Window:** Clase que representa una ventana. Al igual que la clase **Door**, **Window** crea el objeto y posibilita la revisión y el cambio del estado (Abierto/Cerrado). (Se utilizan métodos y/o atributos de: MagneticSensor).

- **Stage4:** Clase que representa el nivel por donde la persona avanzara. Se configuran las cantidades de puertas y ventanas y cuando estas se abren, definiendo, también, los posibles casos de input () que puedan llegar a solicitar al momento de correr el programa. (Se utilizan métodos y/o atributos de:

## Diagrama:



### **Problemas en el proceso:**

1. Al implementar el archivo de sonido de la alarma, esta no se podía desactivar en el momento deseado

**Solución:** Se descubrió que el Sensor se construía con un estado abierto, y las puertas en estado cerrado haciendo que la alarma estuviera activa en momentos no deseados.

2. Falta de comprensión de la lógica de los tipos de datos enum.

**Solución:** Trabajar a partir de la lógica base de las otras funciones creadas en los códigos iniciales entregados por el profesor copiando la forma de estas funciones tratando de imitarlos.

3. Intento de Implementación de la siguiente condición en la clase Central para realizar una comparación de clases: `// if(s.getClass().equals(PIR_Detector.class));`

**Solución:** No se logró resolver el problema por lo que se le dio otro enfoque a la condición. Se podría encontrar una solución dando más tiempo a la búsqueda del funcionamiento de la lógica y sintaxis de las condiciones con clases.