

Practica10

Nombres: Alex Travez, Mateo Oviedo

Fecha de realización: 08 /01/2025

Fecha de entrega: 09/01/2025

Sistema de Control de Puerta para Perro

Análisis y Diseño Orientado a Objetos

1. Análisis de Requerimientos

1.1 Actores del Sistema

- **Perro:** Usuario principal que interactúa con la puerta mediante ladridos
- **Propietario:** Usuario que controla la puerta mediante control remoto
- **Control Remoto:** Dispositivo que permite la apertura de la puerta
- **Sistema:** Actor que maneja el cierre automático y registro de eventos

1.2 Casos de Uso

1. Abrir Puerta

- **Nombre:** UC001 - Abrir Puerta
- **Actores:** Perro, Propietario
- **Condiciones de entrada:**
 - Perro registrado en el sistema
 - Puerta cerrada
- **Flujo de eventos:**
 - El perro ladra o el propietario usa el control remoto
 - Sistema autentica al perro
 - Sistema abre la puerta
 - Sistema inicia temporizador
- **Condiciones de salida:**
 - Puerta abierta
 - Evento registrado en log
- **Requerimientos especiales:**

- Tiempo de respuesta < 2 segundos

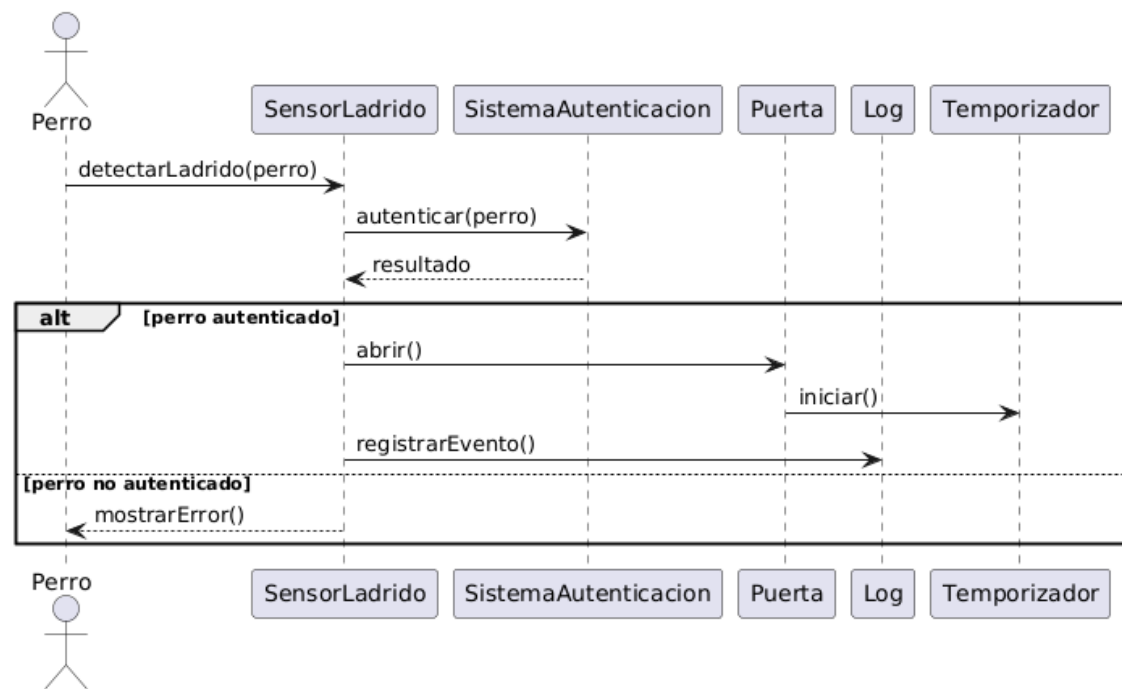
2. Cerrar Puerta Automáticamente

- **Nombre:** UC002 - Cierre Automático
- **Actores:** Sistema
- **Condiciones de entrada:**
 - Puerta abierta
 - Temporizador activo
- **Flujo de eventos:**
 - Temporizador alcanza tiempo configurado
 - Sistema cierra la puerta
 - Sistema registra evento
- **Condiciones de salida:**
 - Puerta cerrada
 - Evento registrado en log

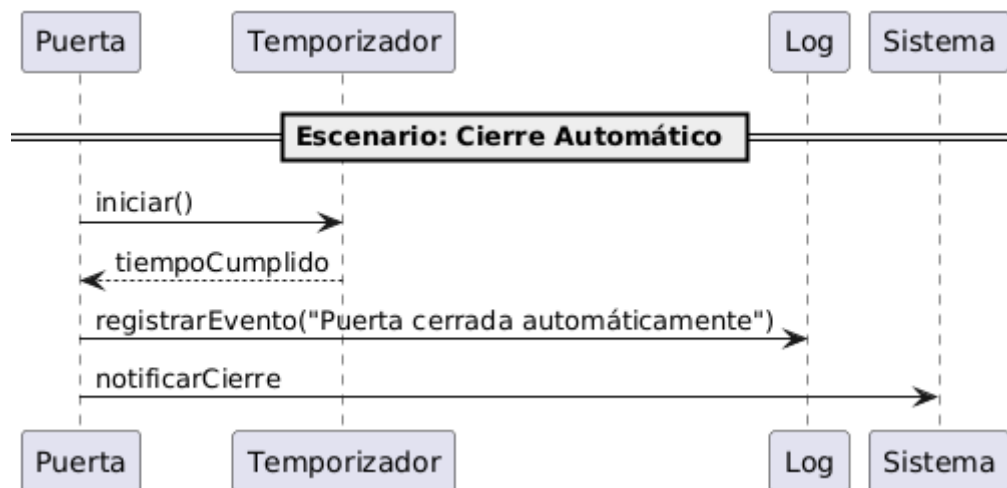
3. Registrar Perro

- **Nombre:** UC003 - Registro de Perro
- **Actores:** Propietario
- **Condiciones de entrada:**
 - Perro no registrado
- **Flujo de eventos:**
 - Propietario inicia registro
 - Sistema captura ID de ladrido
 - Sistema almacena datos del perro
- **Condiciones de salida:**
 - Perro registrado en sistema
 - Evento registrado en log

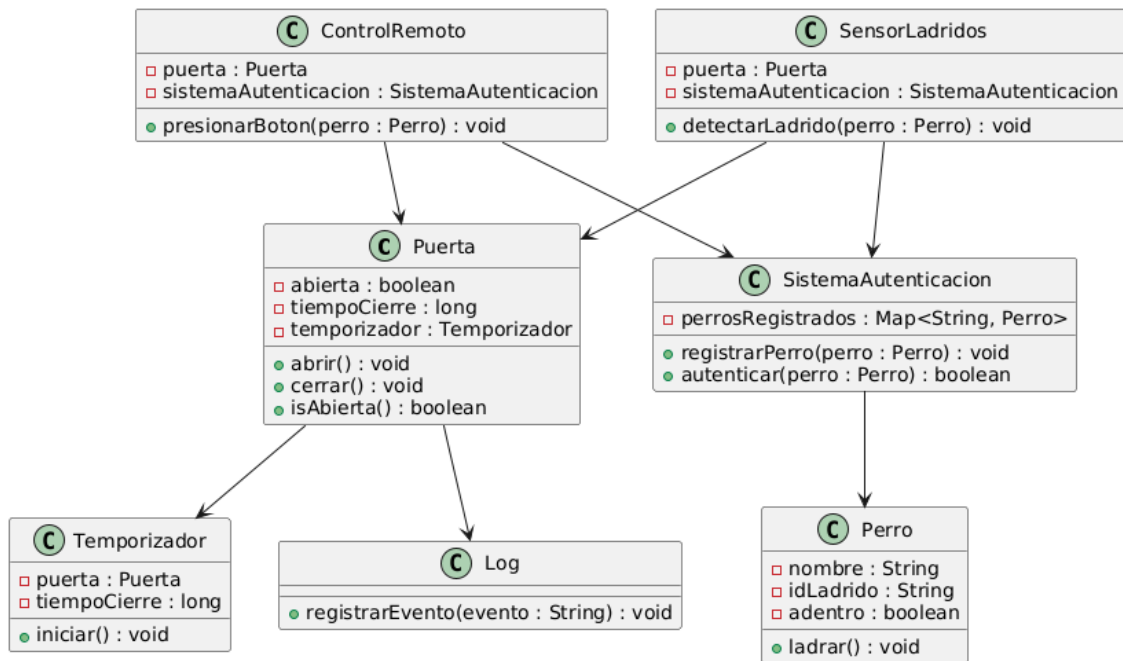




2.3 Diagrama de Secuencia : Cierre Automático



2.3 Diagrama de Clases



3. Conclusiones

1. El sistema implementa exitosamente un diseño orientado a objetos con alta cohesión y bajo acoplamiento.
2. La arquitectura permite una fácil extensión para agregar nuevos métodos de autenticación o control.
3. El uso de patrones de diseño como Observar (para el temporizador) mejora la mantenibilidad del código.

4. Recomendaciones






1. Implementar manejo de excepciones para casos de fallo en la autenticación o problemas mecánicos.
2. Agregar un sistema de configuración para ajustar parámetros como tiempo de cierre.
3. Considerar agregar un modo de operación manual para mantenimiento.

5. Resultados

1. El sistema cumple con todos los requerimientos funcionales establecidos.
2. La implementación permite un registro efectivo de eventos del sistema.
3. El mecanismo de autenticación funciona correctamente para ambos métodos de apertura.

6. Evaluación de Cumplimiento

El sistema cumple satisfactoriamente con lo establecido en los requerimientos.
Específicamente:

-  Implementa dos métodos de apertura (control remoto y ladrido)
-  Incluye cierre automático configurable
-  Proporciona sistema de autenticación de perros
-  Mantiene registro de eventos
-  Implementa correctamente el paradigma orientado a objetos

Referencias :

[1] Analisis y clases de diseño, Lenguaje UML (Material de clase, presentado por David Mejia)[Consultado: 8-JANUARY-2025].