

# Reporte del diseño: proyecto de IoT YALE

---

## Arquitectura de software **C<sup>o</sup>ndax**

Jeison Mateo Contreras  
Juliana Jaime  
Stephannie Jimenez Gacha  
Mauricio Neira  
Laura Valeria Vanegas García

## Contenido:

1.	Introducción.....	3
2.	Arquitectura.....	4
2.1.	Punto de vista del contexto.....	4
2.1.1.	Diagrama de contexto.....	4
2.2.	Vista logica.....	5
2.2.1.	Modelo conceptual.....	5
2.3.	Vista funcional.....	6
2.3.1.	Diagrama de casos de uso.....	6
2.3.2.	Diagrama de componentes.....	8
2.4.	Vista física del sistema.....	9
2.4.1.	Diagrama de despliegue.....	9
2.5.	Vista de concurrencia.....	10
2.5.1.	Diagrama de concurrencia.....	10
3.	Entidad física: sensores y actuadores.....	11
3.1.	Código.....	11
3.2.	Seguridad.....	11
4.	Middleware.....	11
4.1.	Funcionamiento.....	11
4.2.	Seguridad.....	11
5.	Servicios.....	12
5.1.	REST: Funcionamiento.....	12
5.2.	REST: Seguridad con Auth0.....	12
6.	Conclusiones.....	13

# 1. Introducción

Yale es una de las empresas más antiguas del mundo y probablemente la más conocida en la industria de mecanismos de bloqueos o control de acceso a espacios confinados. Fundada en 1868 en Connecticut, Estados Unidos, por Henry R. Towne y Linus Yale, el famoso inventor del cerrojo de pasador. Por 150 años se ha dedicado a este negocio soportando su éxito en la innovación, lo que se materializó en el registro de patentes y su expansión a todo el mundo.

En la actualidad, Yale cuenta con una amplia gama de productos digitales que son capaces de conectarse a internet y ser manejados de manera remota por sus usuarios. Por muchos años se han enfocado en el sector de clientes residenciales y cuentan con líneas de productos muy fuertes como: cerraduras digitales, cerraduras electromecánicas, cerraduras de sobreponer, electroimanes, mirillas digitales, circuito cerrado de televisión (CCTV), cajas fuertes, video porteros, etc.

En cada una de las líneas de productos antes mencionadas se cuenta con un número importante de productos, pero indiscutiblemente el mayor valor para sus clientes se genera cuando se combinan sus productos en la implementación de espacios seguros digitales, lo que realiza a través de “Yale Connect” que es una aplicación móvil que permite controlar sus dispositivos de seguridad desde un Smartphone conectado a internet en cualquier lugar del mundo. Para lograr este objetivo, Yale ha construido un dispositivo para enlazar, controlar y compartir la información de las cerraduras digitales que forman parte de la solución, el cual es comercialmente conocido como “Yale Connect Hub”. Para conocer más pueden ver el siguiente video: [https://www.youtube.com/watch?time\\_continue=14&v=\\_gLu33hS47o](https://www.youtube.com/watch?time_continue=14&v=_gLu33hS47o)

Las soluciones propuestas actualmente por Yale se enfocan en clientes individuales, pero están proyectando ofrecer una solución a mayor escala para unidades residenciales donde el número de viviendas puede estar entre 100 y 200, donde cada una podría contar con una implementación a la medida de las necesidades y deseos de sus propietarios.

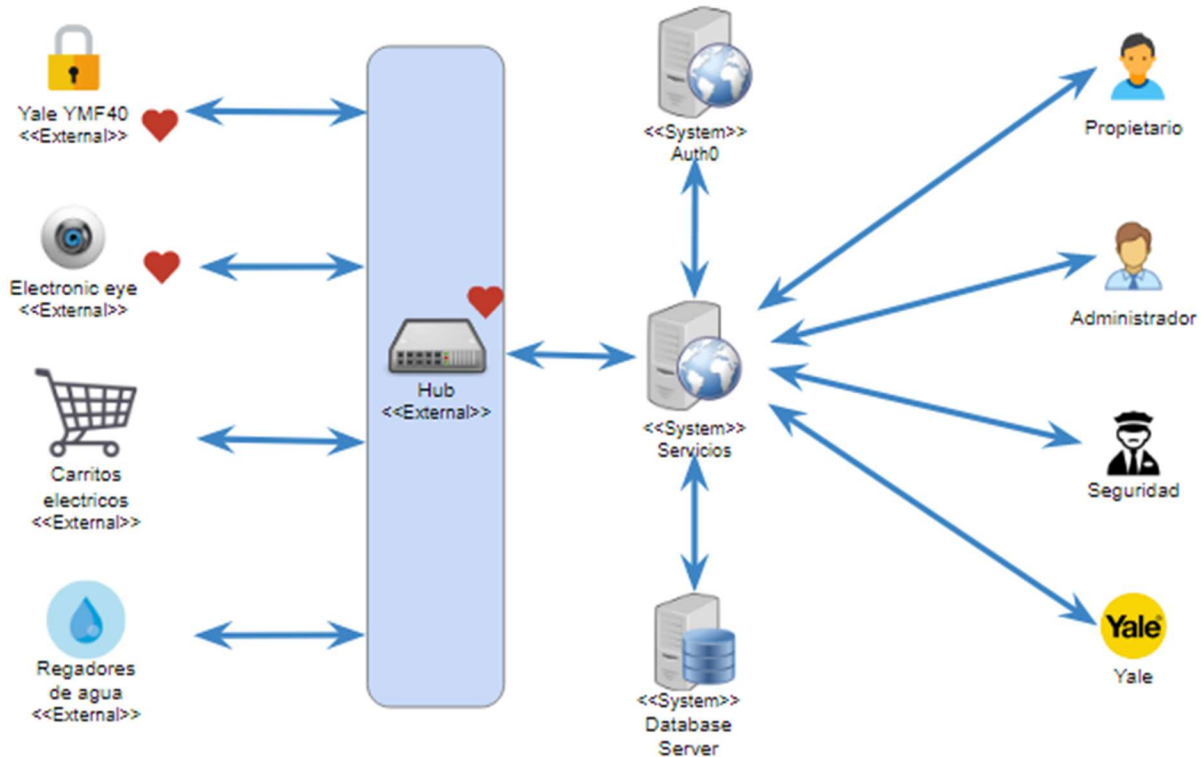
Basado en esta premisa y en el interés de Yale Colombia de ampliar su portafolio de servicios de seguridad dirigido a unidades residenciales (i.e., conjuntos o edificios), ha lanzado el reto a grupos de arquitectos de software buscando que propongan servicios basados en tecnologías IoT, que permitan generar servicios atractivos para sus clientes.

## 2. Arquitectura

La siguiente sección muestra las diferentes vistas de la arquitectura usada en el proyecto.

### 2.1. Punto de vista del contexto

#### 2.1.1. Diagrama de contexto

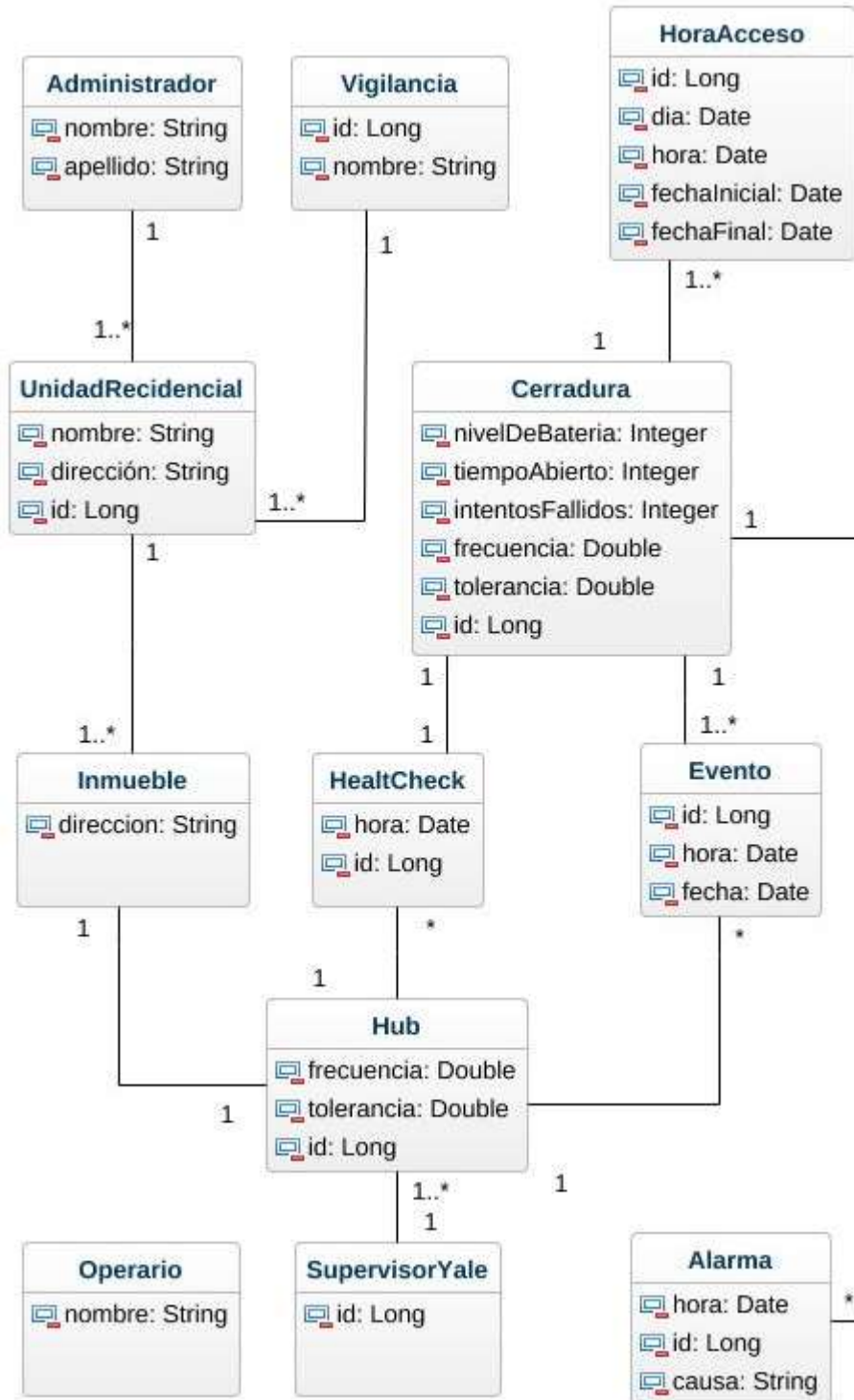


Para este diagrama, se tiene en cuenta un componente de innovación. Principalmente, la cerradura Yale TMF40 y el Yale HUB, adicionalmente se tiene:

- Un ojo electrónico: En caso de que, cuando un domicilio llegue a la unidad residencial o algún otro tipo de arribo, el propietario sea notificado de forma mas rapida y sin la intermediación de la seguridad.
- Un juego de carritos electrónicos: Cuando se hace mercado y se quiere llevar las múltiples cantidades de paquetes a la casa o apartamento requerido, estos carritos pueden ser de muy gran ayuda, sabiendo la ruta predefinida para cargar las cosas compradas hasta un hogar.
- Regadores de agua: En unidades residenciales con zonas verdes internas o plantas en macetas, estos rociadores serán de gran ayuda para regar dichas plantas de forma preprogramada.

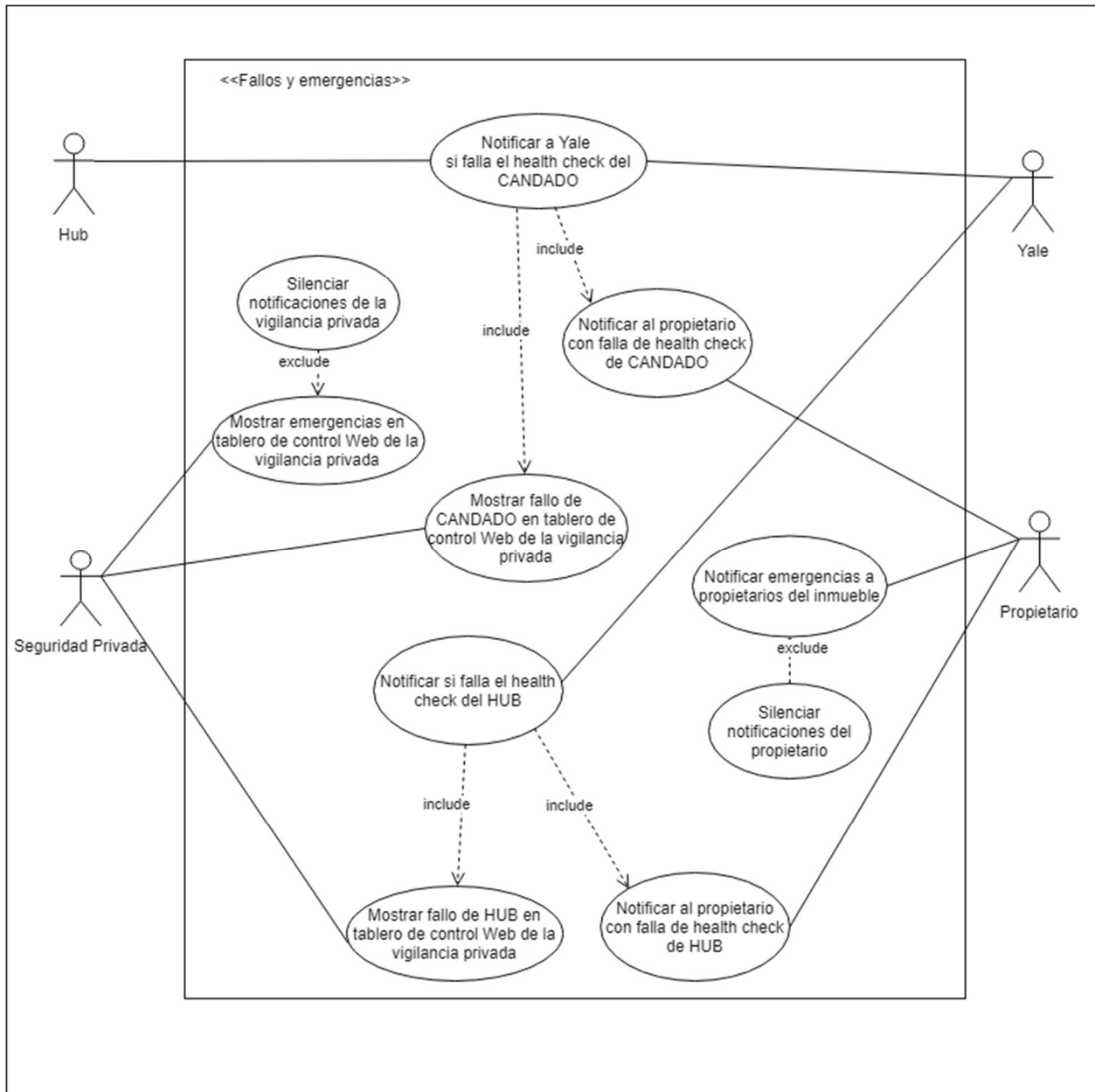
## 2.2. Vista lógica

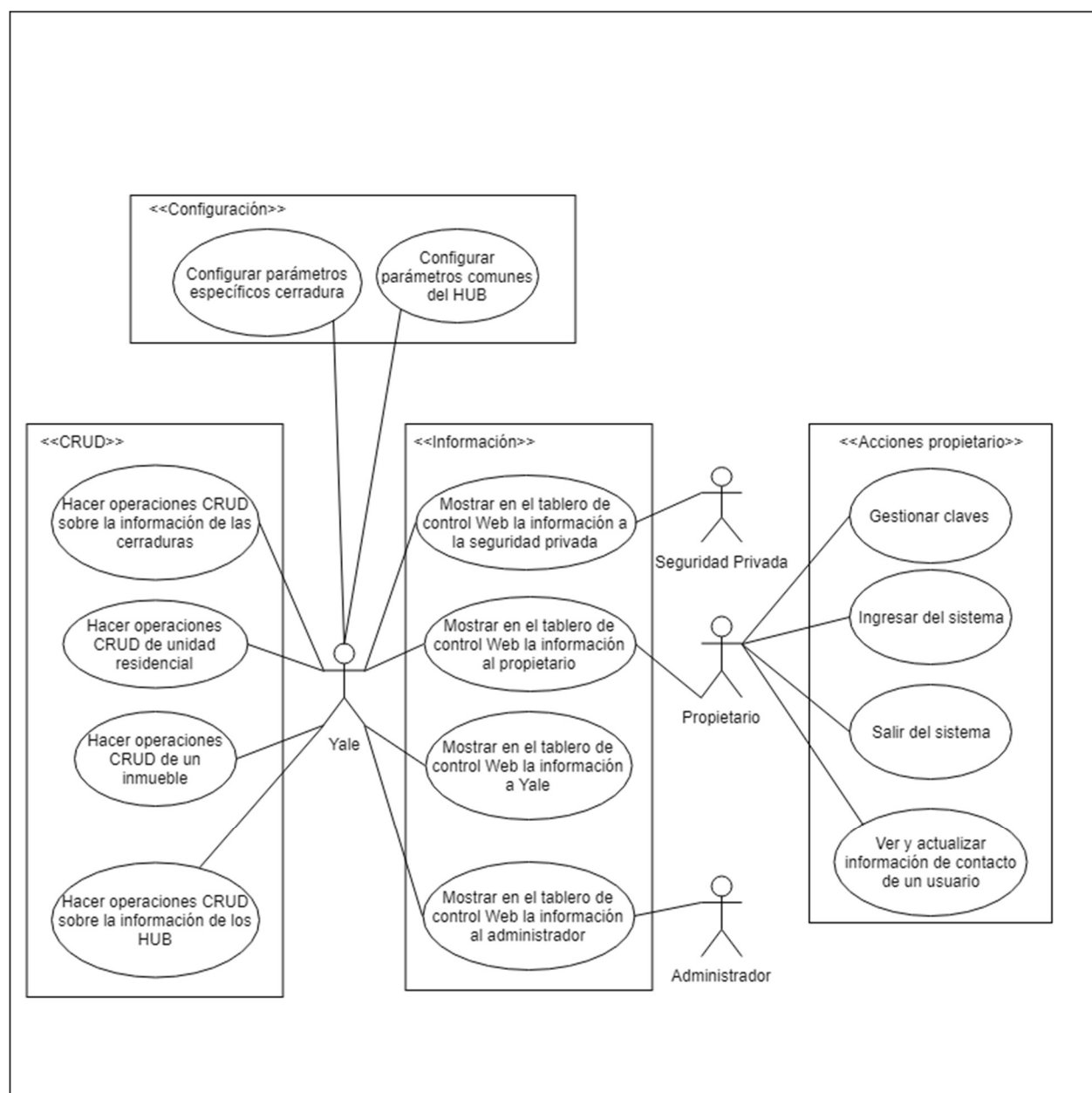
### 2.2.1. Modelo conceptual



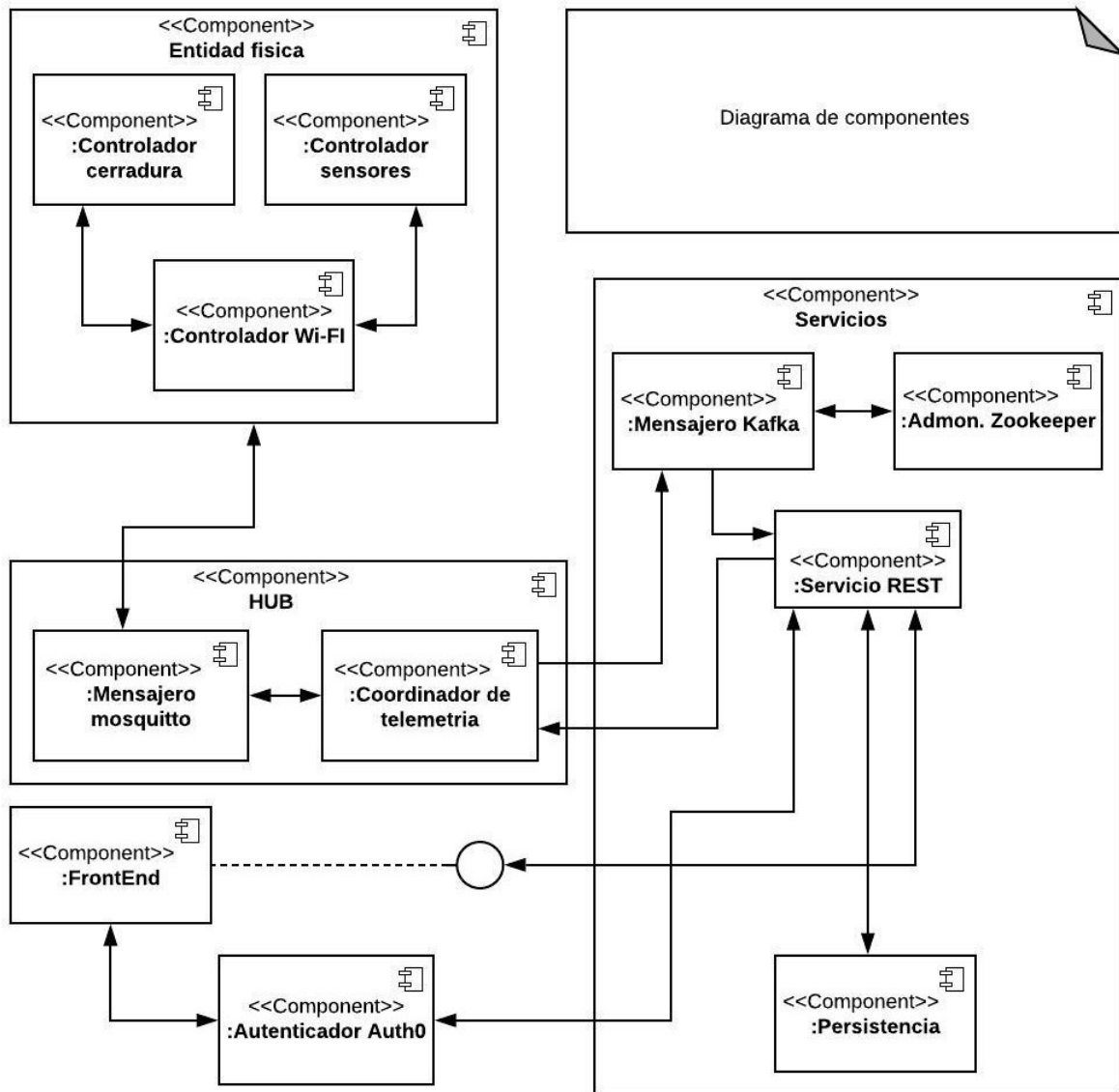
## 2.3. Vista funcional

### 2.3.1. Modelo de casos de uso





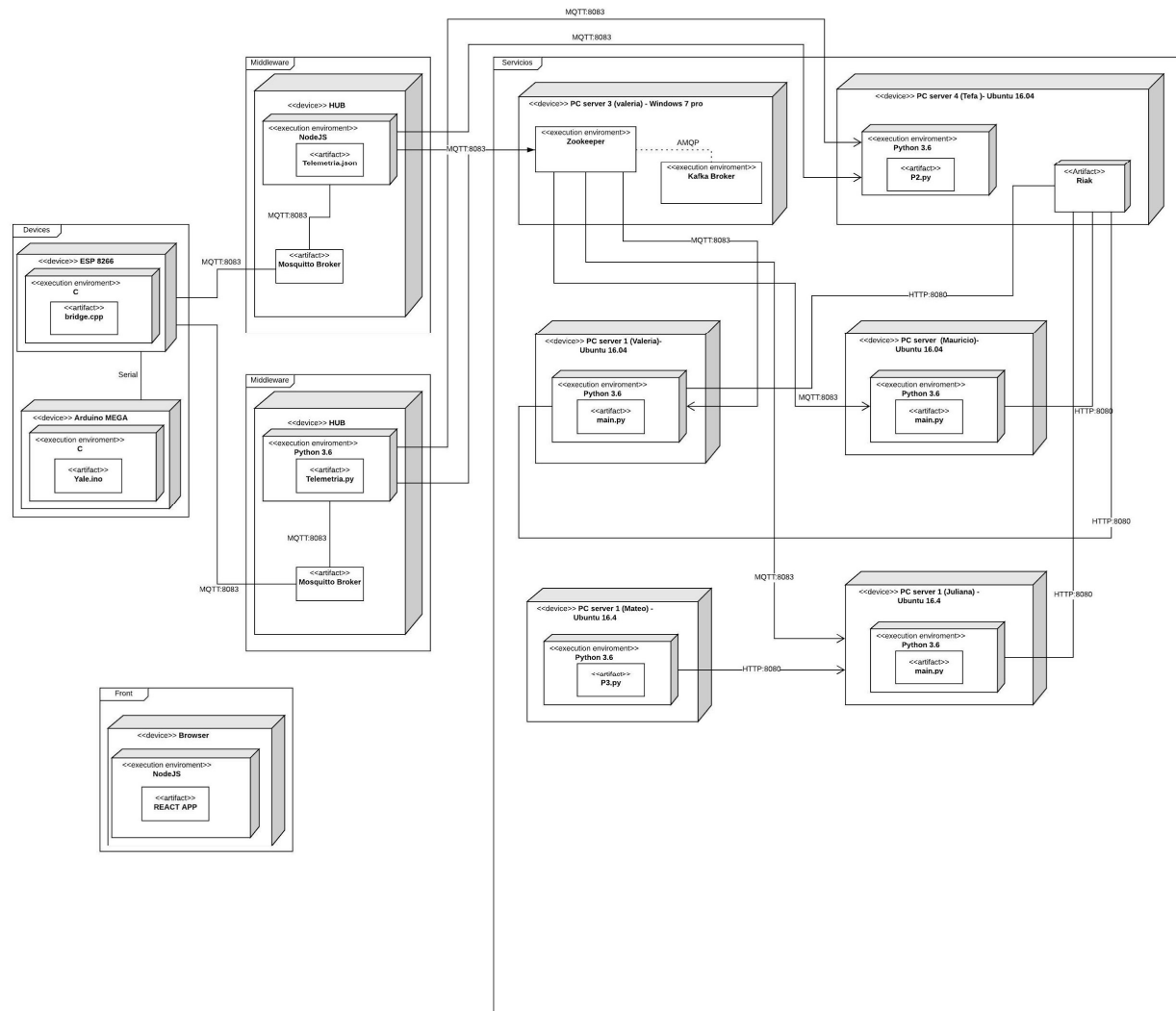
## 2.3.2. Diagrama de componentes





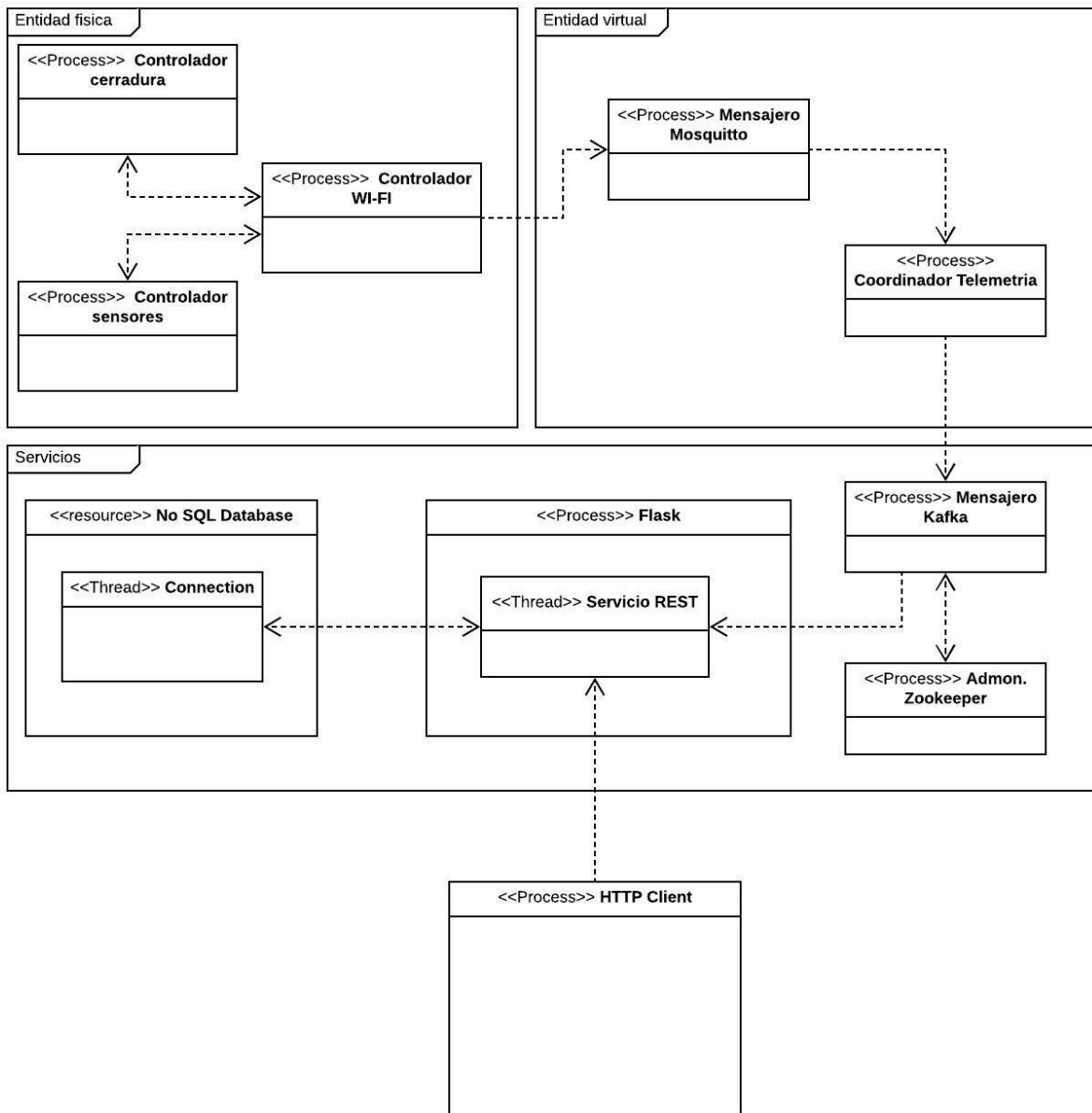
## 2.4. Vista física

### 2.4.1. Diagrama de despliegue



## 2.5. Vista de concurrencia

### 2.5.1. Diagrama de concurrencia



## 3. Entidad física: Sensores y actuadores

### 3.1. Código

- ❖ `setup()`: Inicializa toda la configuración y variables de la cerradura.
- ❖ `loop()`: Tiene todo el proceso que se encarga de verificar el estado de la cerradura.
- ❖ `setColor(int redValue, int greenValue, int blueValue)`: Establece el color del led de estado.
- ❖ `compareKey(String key)`: Compara la clave del parámetro con las guardadas en el EEPROM.
- ❖ `processCommand(String command)`: Divide el comando en partes para ser procesadas.
- ❖ `addPassword(int val, int index)`: Agrega una contraseña nueva al EEPROM.
- ❖ `updatePassword(int val, int index)`: Actualiza una contraseña específica.
- ❖ `deletePassword(int index)`: Elimina una contraseña específica.
- ❖ `deleteAllPasswords()`: Elimina todas las contraseñas.

### 3.2. Seguridad

- ❖ Se utiliza el envío de comandos cifrados, además, se utiliza un protocolo secreto para darle más seguridad a la transferencia.

## 4. Middleware

### 4.1. Funcionamiento

- ❖ El HUB tiene un servidor MQTT que recibe todas las peticiones tanto del usuario para configuración, a través de un tópico, como de la información recolectada por la cerradura, a través de un tópico para cada barrio.
- ❖ Finalmente se tiene un programa interno que hace la mediación entre el Mosquitto broker y el servidor de Yale.

### 4.2. Seguridad

- ❖ Para el Middleware no se maneja seguridad aún.

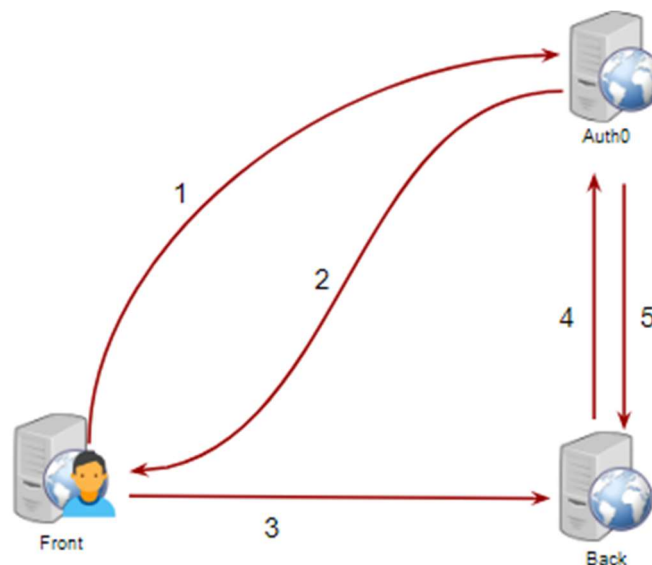
## 5. Servicios

### 5.1. REST: Funcionamiento

- ❖ Los servicios comienzan desde un servidor de Kafka + Zookeeper, al cual llegan las peticiones de los diferentes HUBs, estas son procesadas por los servicios rest.
- ❖ Los servicios rest están disponibles para el usuario, a través del FrontEnd. También para las peticiones que llegan vía MQTT.

### 5.2. REST: Seguridad con Auth0

- ❖ Los servicios rest están protegidos con Auth0, lo que exige al usuario Autenticación y Autorización.
- ❖ Para esta implementación se ilustra mejor en el siguiente diagrama:



1. Se solicita desde el front un token a Auth0
2. Auth0 Autentica y envia el token al front
3. El usuario envía este token para conectarse al back
4. El back nuevamente verifica con auth0 la autenticidad del token y revisa la autorización.
5. El back recibe una notificación (correcta o denegada) para realizar la operación

Finalmente el usuario recibe respuesta

## **6. Conclusiones**

Se agrego el uso de la memoria secundaria de la cerradura (EEPROM) la cual permite almacenar las contraseñas de forma permanente en la cerradura.

Se añade el atributo de calidad: seguridad, a la solución necesaria para el problema de Yale. Para esto se usó Auth0 que implementa técnicas de Autenticación y Autorización, protegiendo así los servicios.