**Software Architecture Documentation (SAD)**

**Equipo: πpo**

**Sergio Cárdenas**

**Juan Sebastián Díaz**

**Sergio Guzmán**

**Julián Manrique**

**Carlos Mario Sarmiento**

**Arquitectura y Diseño de Software**

**2018-10**

**Universidad de los Andes**

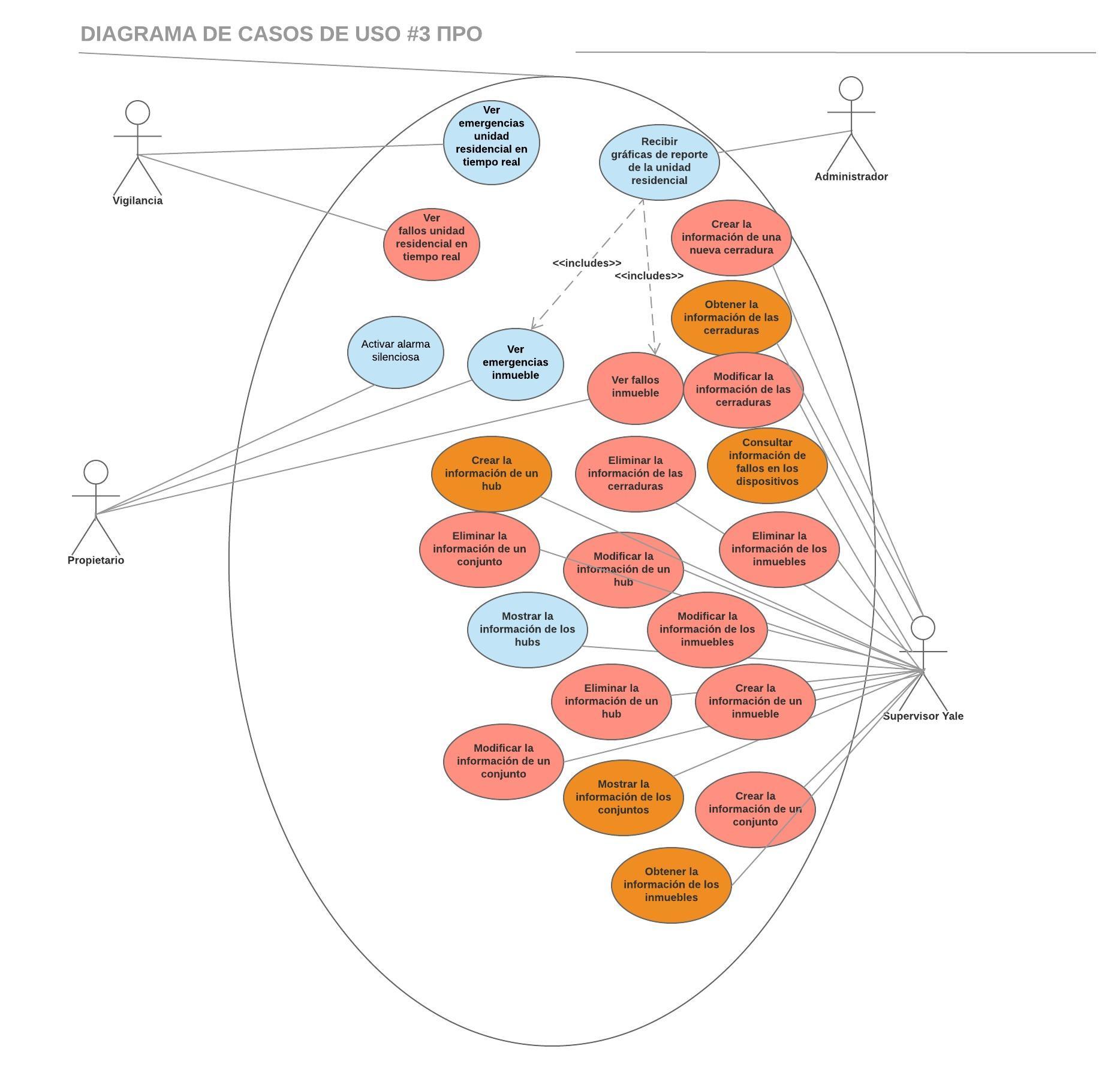
1. **Restricciones**

Según el documento de descripción del sistema, solo se tienen restricciones de tecnología en la comunicación, siendo estas las siguientes:

* El Hub y la cerradura se comunican a través de la red inalámbrica interna del inmueble.
* El Hub se comunica con el sistema central de Yale a través de la conexión de internet de un proveedor externo.
* El sistema de Yale se comunica con el sistema de vigilancia a través de un canal de Internet dedicado.

1. **Casos de Uso**

De acuerdo con el documento de especificaciones dado e instrucciones posteriores, se determinó que el sistema tenía cuatro actores principales: el propietario de un inmueble, el administrador de una unidad residencial, un operario de vigilancia de la misma, y el administrador de Yale. Los casos de uso de estos son los siguientes:



1. **Escenarios de Calidad**

Los escenarios de calidad encontrados son los siguientes:

|  |  |  |
| --- | --- | --- |
| Identificador | Tipo | Prioridad |
| EC1 | Escalabilidad | Alta |
| Fuente | | |
| Cerraduras | | |
| Estímulo | | |
| Notificador de emergencias y fallos. | | |
| Ambiente | | |
| Sobrecargado | | |
| Medida Esperada | | |
| El sistema debe poder recibir 300.000 peticiones en una ventana de 1 minuto con 0% de error y tiempo de respuesta de máximo 1 segundo. | | |

Según el documento, “el sistema debe estar disponible 24 horas, los 7 días a la semana” dado que se trata de un sistema de seguridad, determinamos que esto es equivalente a una disponibilidad de sigma 6.

|  |  |  |
| --- | --- | --- |
| Identificador | Tipo | Prioridad |
| DI1 | Disponibilidad | Alta |
| Fuente | | |
| Sistema | | |
| Estímulo | | |
| Notificador de emergencias y fallos. | | |
| Ambiente | | |
| Normal | | |
| Medida Esperada | | |
| El sistema debe tener una disponibilidad de sigma 6, o 99.9999%. | | |

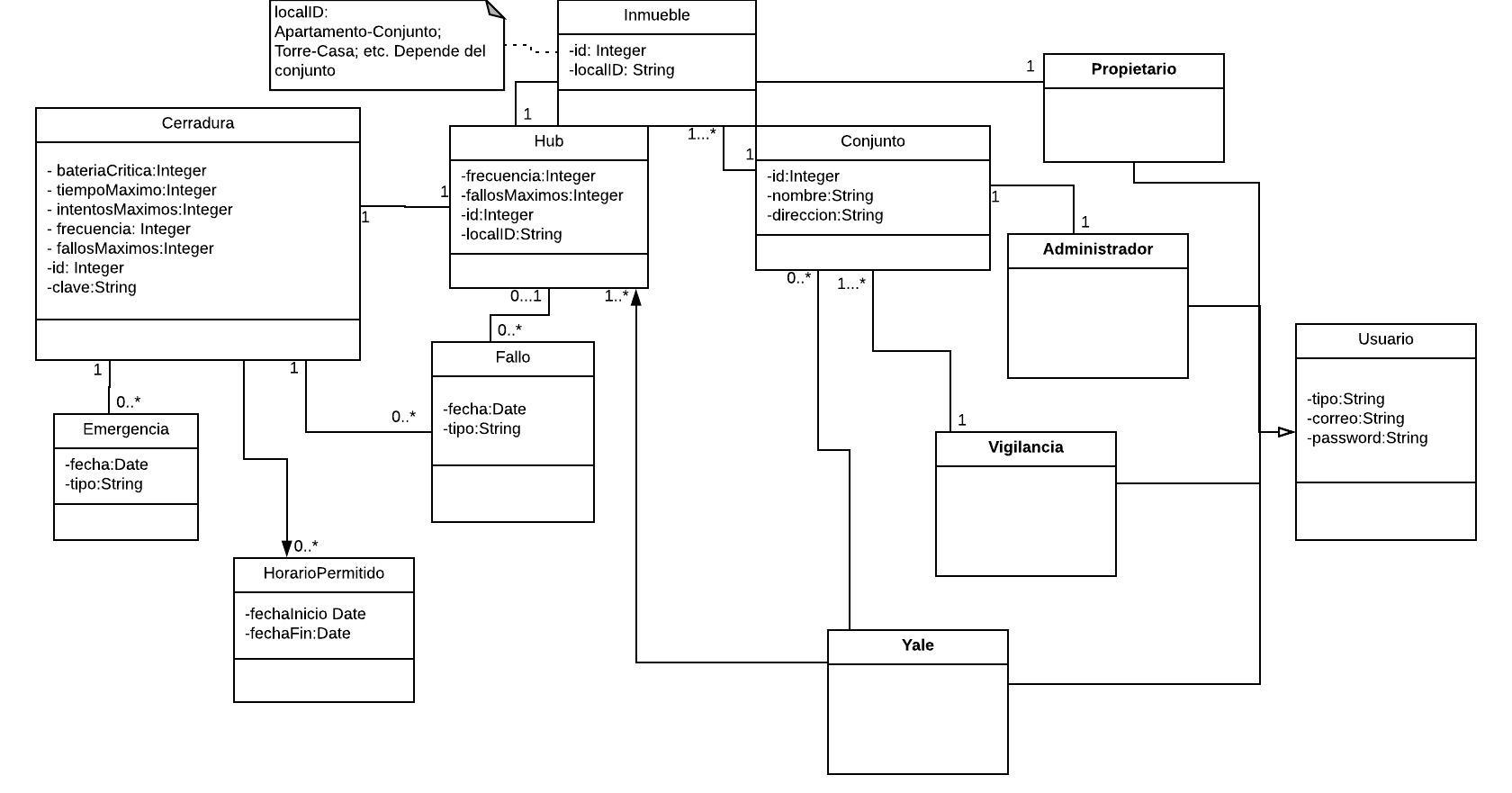
|  |  |  |
| --- | --- | --- |
| Identificador | Tipo | Prioridad |
| SE1 | Seguridad | Alta |
| Fuente | | |
| Todos los actores | | |
| Estímulo | | |
| Ingreso al portal Web | | |
| Ambiente | | |
| Normal | | |
| Medida Esperada | | |
| El portal Web debe tener autenticación y autorización dependiendo de los 4 roles (propietario, administrador, seguridad, Yale). | | |

|  |  |  |
| --- | --- | --- |
| Identificador | Tipo | Prioridad |
| SE2 | Seguridad | Media |
| Fuente | | |
| Cerraduras | | |
| Estímulo | | |
| Envío de alarmas al sistema | | |
| Ambiente | | |
| Normal | | |
| Medida Esperada | | |
| El sistema central de Yale solo debe recibir notificaciones y alarmas de dispositivos registrados y asegurados. | | |

|  |  |  |
| --- | --- | --- |
| Identificador | Tipo | Prioridad |
| US1 | Usabilidad | Alta |
| Fuente | | |
| Seguridad Privada | | |
| Estímulo | | |
| Tablero de Control | | |
| Ambiente | | |
| Normal | | |
| Medida Esperada | | |
| La seguridad privada debe tener un tablero de control tipo dashboard en el cual se vean claramente los estados de los dispositivos del inmueble, así como sus alarmas. | | |

1. **Modelo Conceptual**

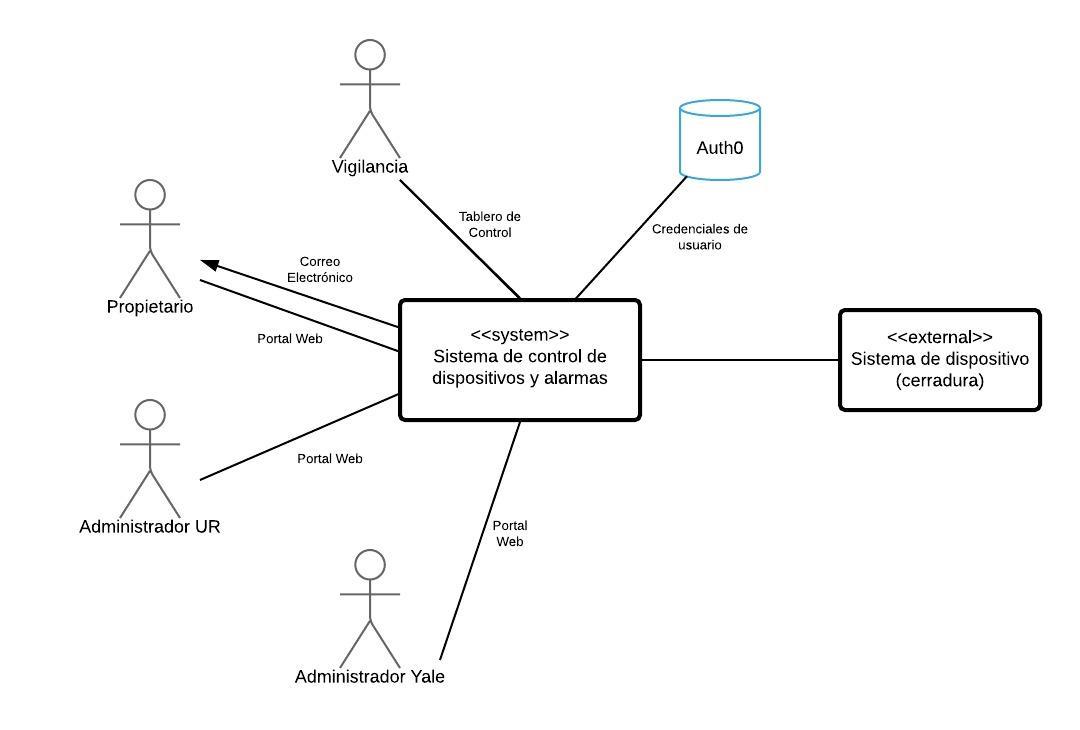
Analizando todo lo anterior, se determinó usar el siguiente modelo conceptual:



Mostrado como diagrama de clases, ya que es más claro así que a partir de un modelo E/R. Estas no son las únicas clases del proyecto, únicamente aquellas que corresponden a los objetos persistidos en Base de Datos.

1. **Vista de Contexto**

La vista de contexto del sistema construido es la siguiente:

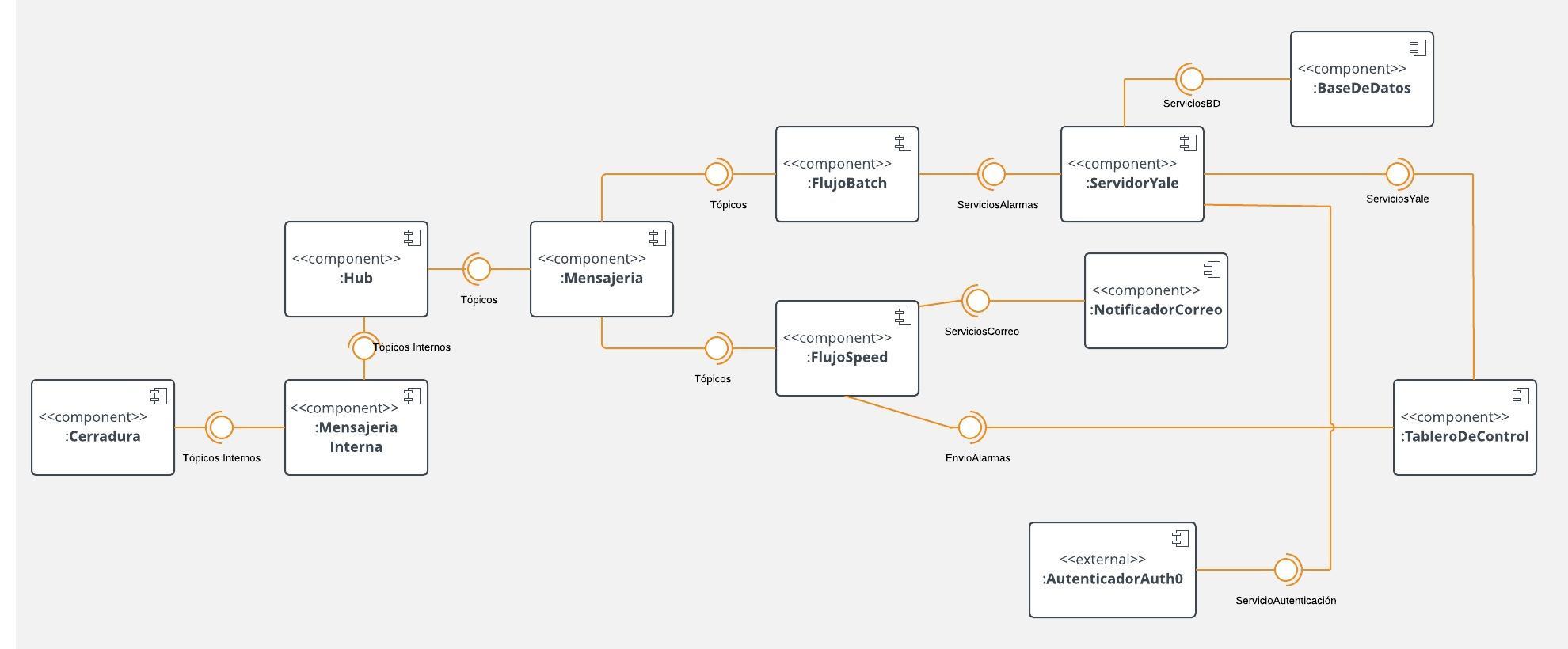


Se decidió una base de datos externa para almacenar credenciales ya que se consideraba más segura para propósitos del proyecto. También vale la pena destacar que el sistema envía notificaciones de correo al propietario de un inmueble, y este además puede acceder al sistema por medio del portal web (como todos los demás actores).

Por otro lado, aunque para esta versión se desarrolló el sistema de control de la cerradura, en la aplicación real se espera que esta tenga un sistema externo, aparte al construido.

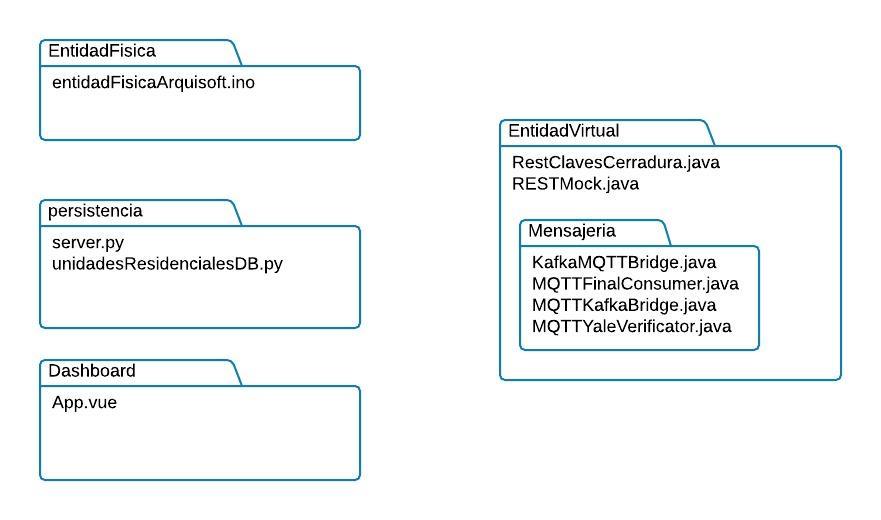
1. **Vista de Componentes**

Desde la vista de componentes, el sistema se ve de la siguiente manera:

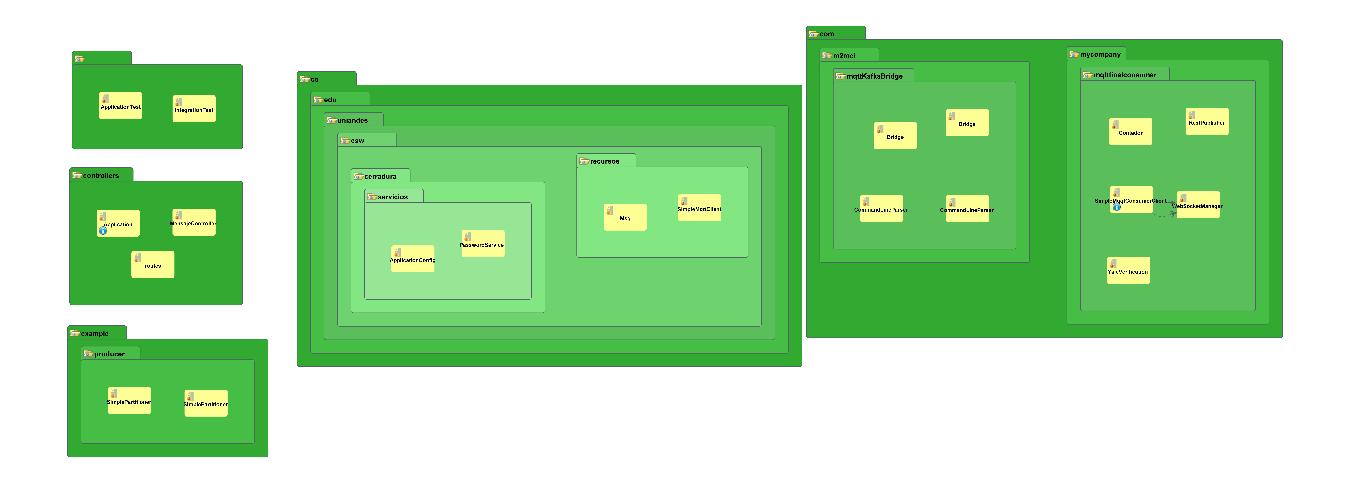


1. **Vista de Desarrollo**

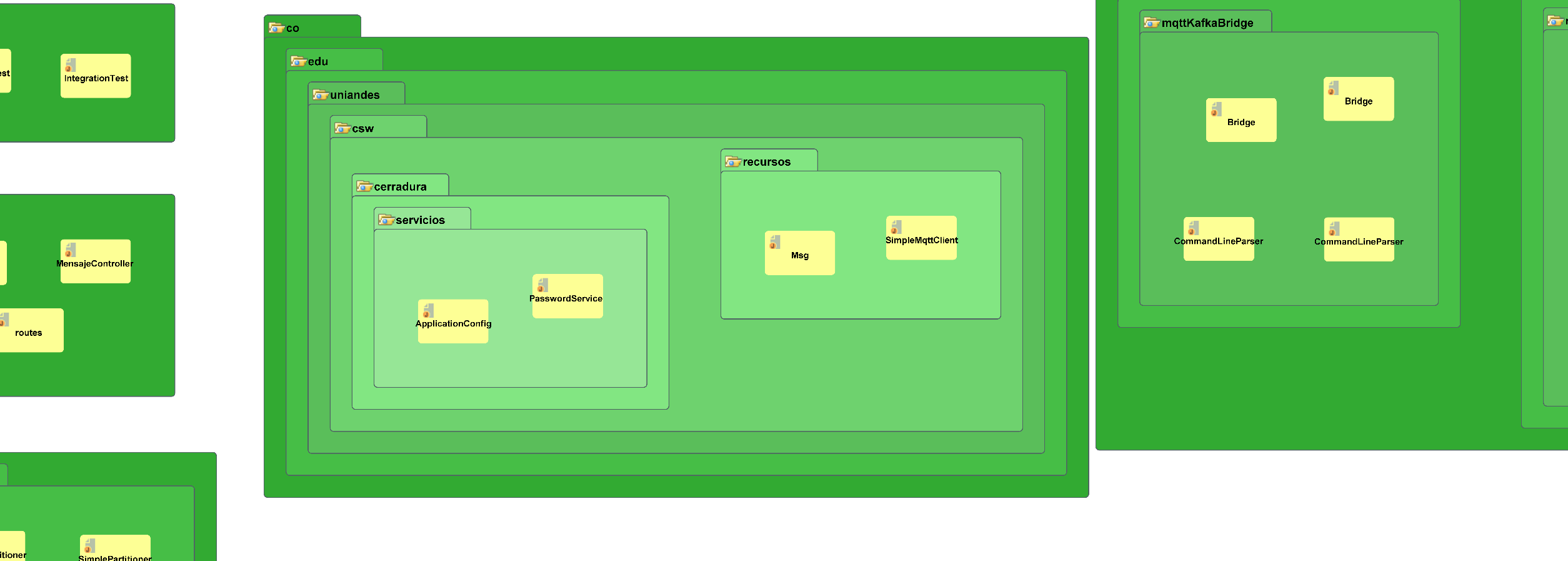
Dada la complejidad del proyecto, es prácticamente imposible tener un solo diagrama de paquetes detallado de todo el sistema. Es el ámbito general tenemos el siguiente diagrama:

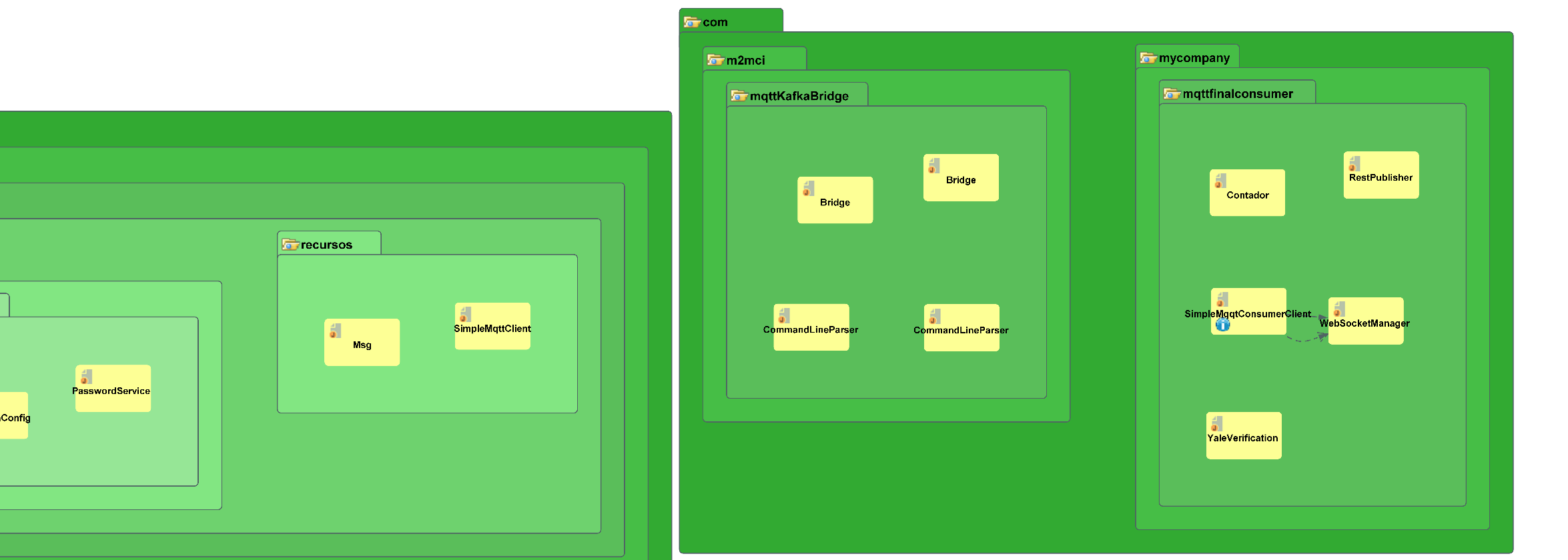


Donde los *.java* y *.vue* no son archivos Java ni JavaScript, sino proyectos de este tipo. Para estos podemos encontrar los siguientes diagramas detallados, para la Entidad Virtual:

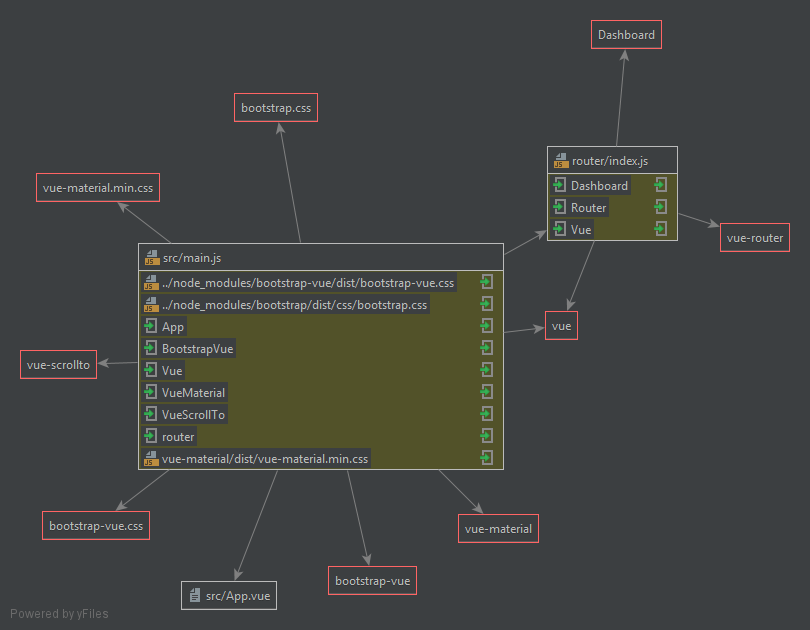


Como no se puede leer fácilmente, se incluyen también las tres grandes partes en mayor detalle:

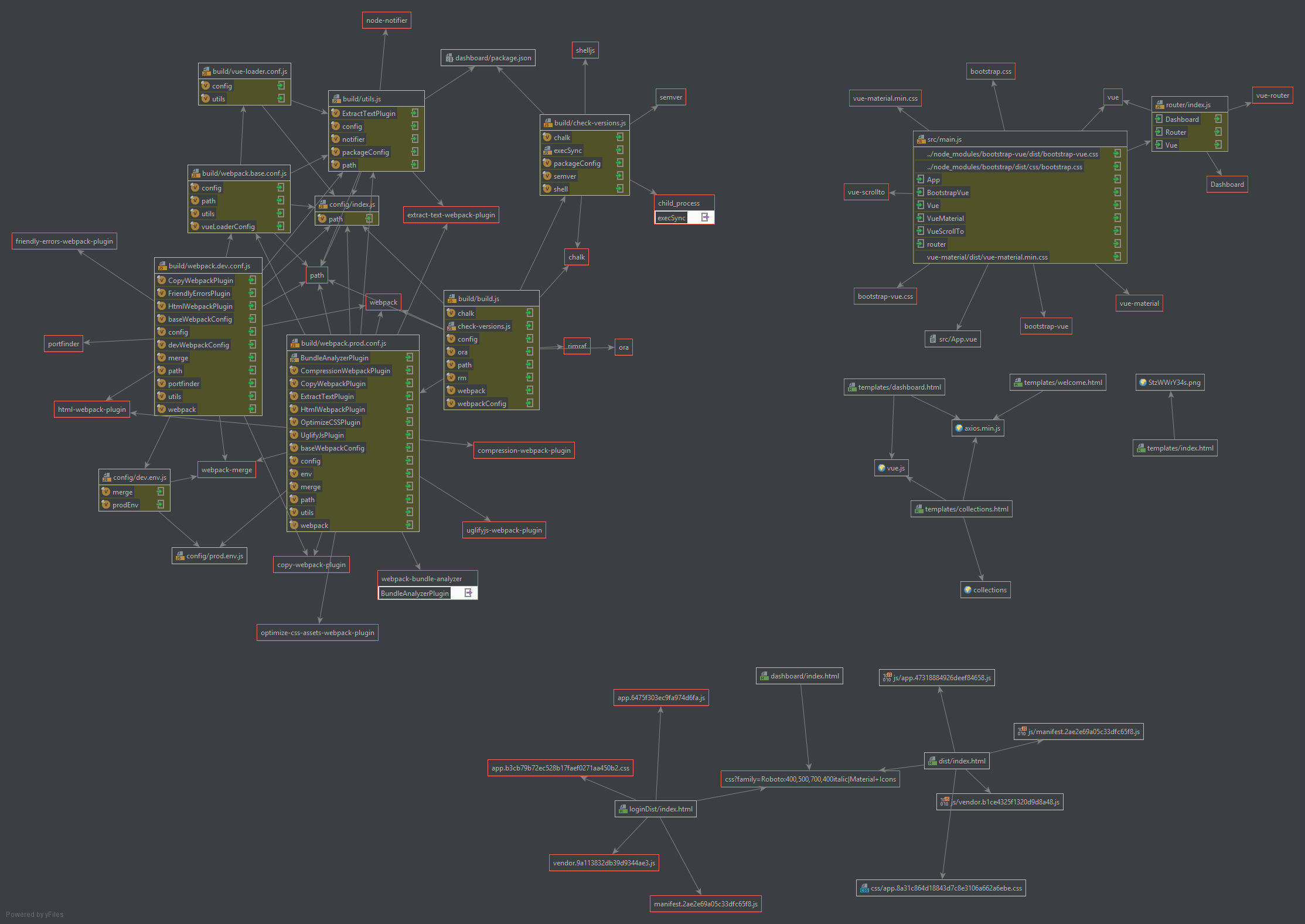


w

Por otro lado, para el Dashboard, la estructura es la siguiente:

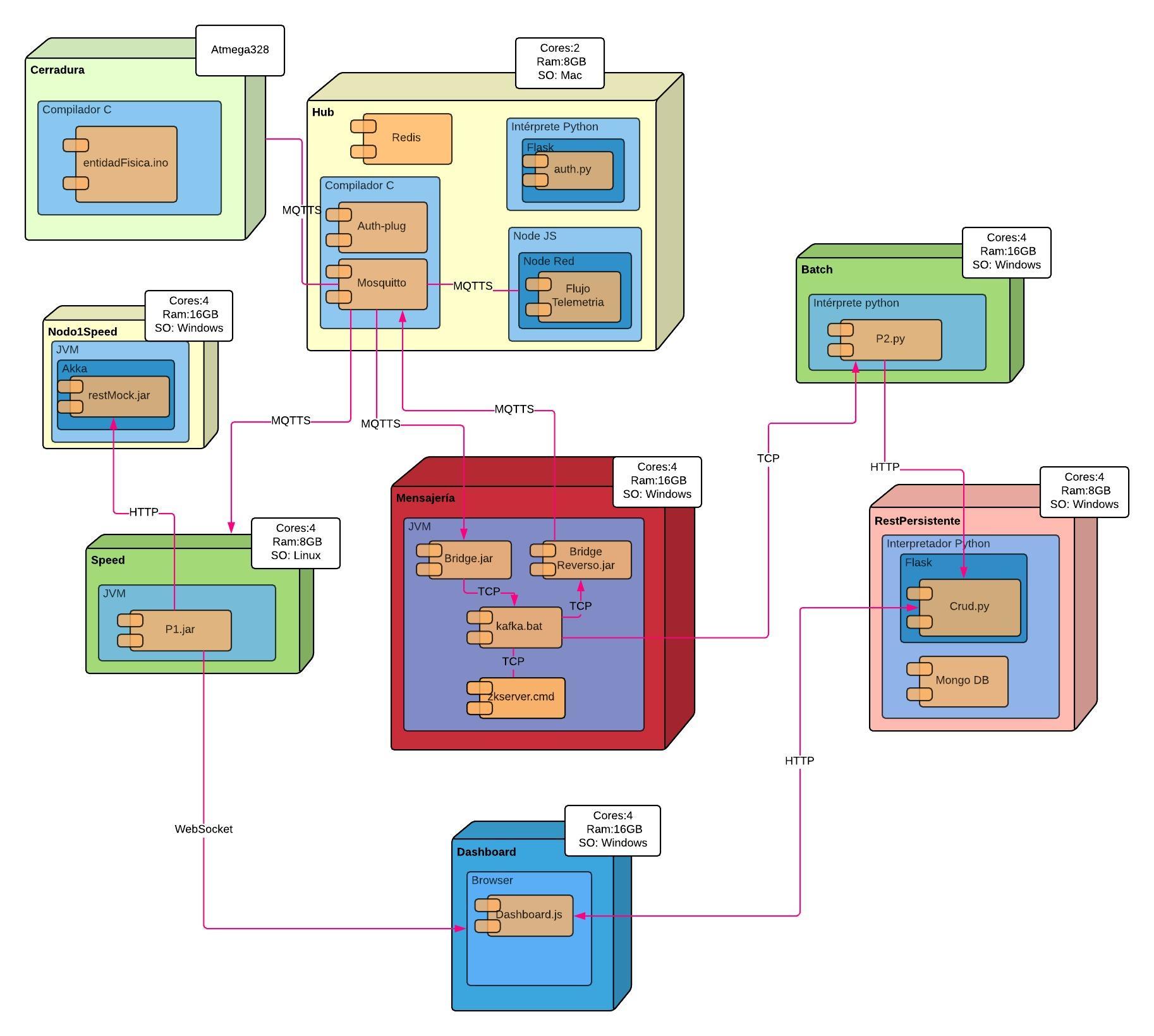


Esta componente del sistema tiene además un gran número de dependencias debido a los frameworks usados. Estas dependencias pueden verse en el siguiente diagrama:



1. **Vista de Despliegue**

La vista de despliegue final de sistema es la siguiente:



En este diagrama los dispositivos de cerradura y Hub son uno por inmueble, y el Dashboard se corre en el browser del cliente, con las especificaciones que este pueda tener. Además, para poder acomodar el requerimiento de carga, se decidió hacer múltiples copias del nodo Speed, cada una atendiendo peticiones de una región diferente, con el objetivo de dividir la carga esperada entre ellos. Actualmente se planificaron cuatro de estos nodos (todos en el sistema de Yale), correspondientes a las regiones Norte, Centro, Sur, y Periferia.

Los demás nodos tienen solo una copia de computación cada uno, y se encuentran en el sistema central de Yale.