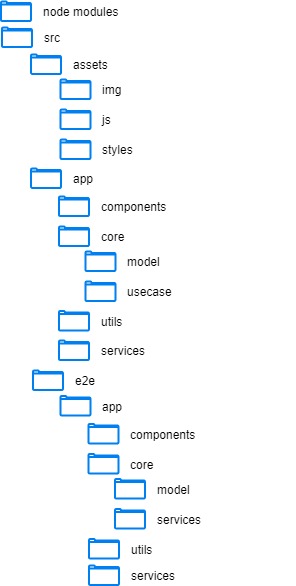
ESTRUCTURACIÓN DE LA APLICACIÓN

**Sprint 1**

# Frontend



**Node modules:** directorio en el cual se guardan las librerías de nodejs, como las de angular y las de terceros.

**Src:** directorio en el cual van todos los recursos de la aplicacion y tambien la aplicacion misma con sus respectivas pruebas.

**Assets:** directorio donde se guardan los recursos de la aplicación, como imágenes, estilos genéricos y scripts genéricos.

**Img:** directorio donde se encuentran todas las imágenes usadas en la aplicación, tanto iconografía en svg como en formato de imagen.

**Js:** directorio en el cual se consignan las librerías js que no están en hechas para angular pero se quieren usar en la aplicación.

**Styles:** directorio en el cual se consignan los estilos de ccs de estas librerías o styles genéricos para la aplicación.

**App:** directorio donde se encuentran todos los scripts de la aplicación.

**Components:** directorio para las páginas y componentes más pequeños debido a que angular está orientado a componentes.

**Core:** directorio en el cual va consignada toda la lógica de aplicación y el modelo.

**Model**: directorio en el cual van las entidades de la aplicación.

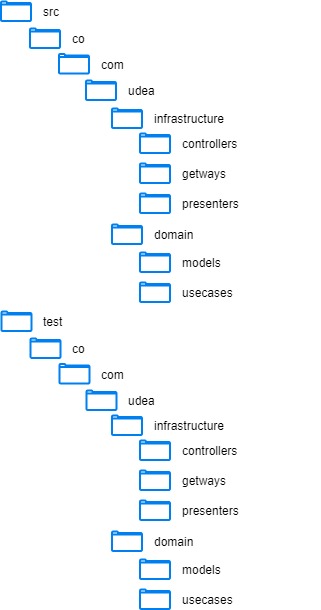
**Usecase:** directorio en el cual va la lógica de negocio independiente a la lógica de la capa de presentación.

**Utils:** directorio en el cual van los validadores, o utilidades para la aplicación como mapeo de objetos a otros, etc.

**Services:** directorio en el cual van todos los servicios o providers de angular para traer las información o llevar la información al backend.

**E2e:** directorio en el cual van todas las pruebas de nuestra aplicación con Jasmin y Karma, la estructura es igual a la estructura de la aplicación para que sea más comprensible encontrar el test de los diferentes directivas, servicios o controladores de nuestra aplicación.

# Backend



**src:** directorio en el cual se encuentra todo el código fuente de la aplicación.

**test:** directorio en el cual se encuentran los test de cada una da las clases de java de la aplicación.

**co.com.udea.infrastructure:** en este paquete van las conexiones con apis externas, servicios rest, la capa de presentación, etc.

**co.com.udea.infrastructure.controllers:** paquete en el cual van los REST y/o SOAP expuestos por nuestra aplicación, también la conexión con dispositivos externos.

**co.com.udea.infrastructure.getways:** paquete en el cual va la implementación de los repositorios de base de datos y servicios externos.

**co.com.udea.infrastructure.presenters:** paquete en el cual va la interfaz de usuario o mensajes del backend.

**co.com.udea.domain:** paquete en el cual se encuentra el core de la aplicación y las interfaces para los repositorios y/o servicios externos.

**co.com.udea.domain.model:** paquete en el cual se encuentran las entidades de la aplicación, no necesariamente deben ser las mismas que usan los repositorios.

**co.com.udea.domain.usecase:** paquete en el cual va toda la lógica del negocio.