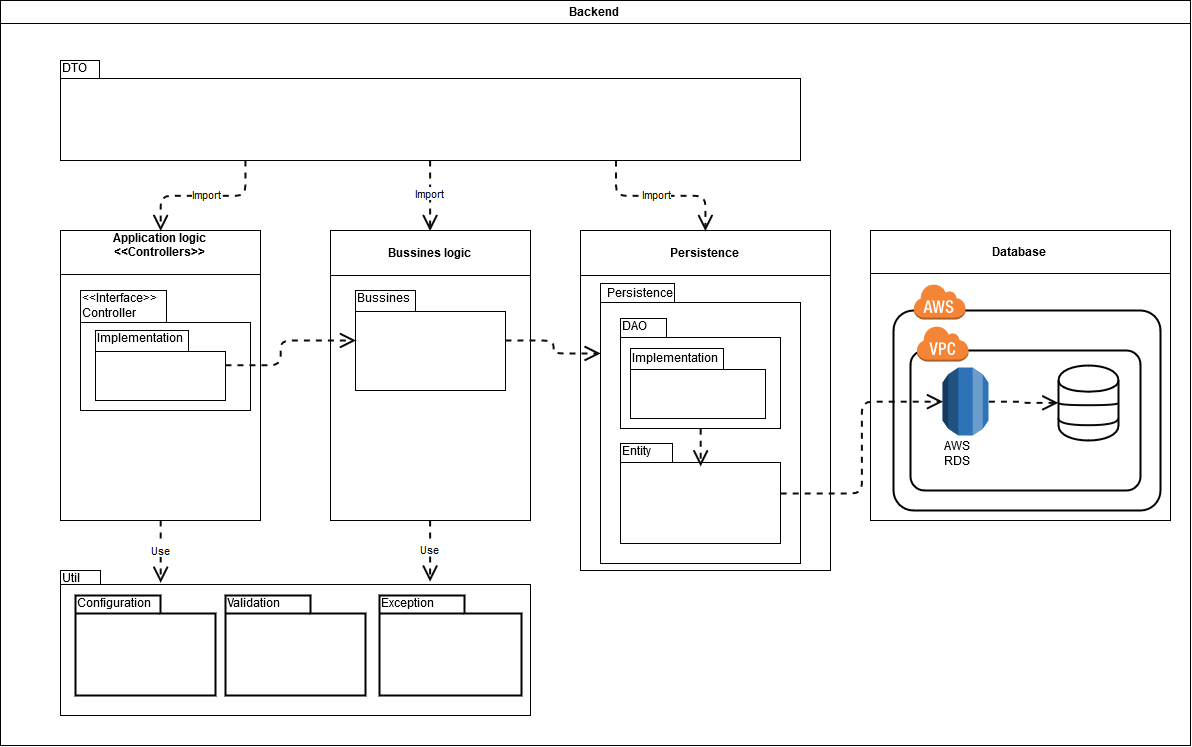
**ARQUITECTURA PROPUESTA PARA EL BACKEND Y EL FRONTEND**

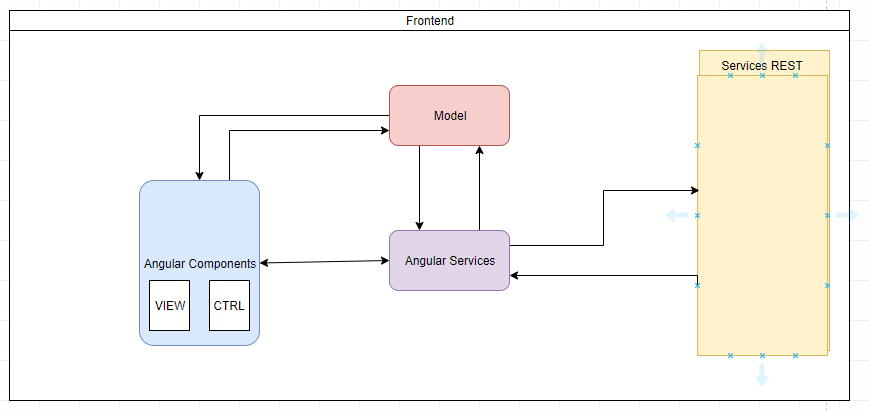
**Arquitectura a nivel Backend**



Se propone una arquitectura derivada de MVC, en la cual se tiene la siguiente distribución:

* La capa ​Controller maneja la conexión con el cliente, recibiendo las peticiones que son enviadas para recuperar los datos. Esto mediante la implementación de interfaces.
* La capa ​Business procesa las validaciones de los datos que se enviaron a través de la petición, y envía o recibe datos de la capa ​Persistence
* La capa ​Persistence tiene dos elementos importantes:
  + Los archivos ​DAO los cuales son interfaces implementadas con las que se realizan las consultas a la base de datos, haciendo uso de la información recibida y las entidades.
  + El paquete ​Entity maneja todo el mapeo ORM que se realiza con las clases de la aplicación y las tablas en la base de datos.
* Simultáneamente se manejan dos capas de forma transversal:
  + El paquete ​DTO se encarga de todo el paso de información a lo largo de las capas, de esta manera sólo se utilizan objetos de la entidad en la capa ​DAO.
  + El paquete ​Util contiene todo lo referente a seguridad y configuración. En él se manejan las validaciones, las constantes, las excepciones y demás elementos necesarios para tener una aplicación controlada.
* Por último, para la base de datos, se utilizó un motor ​SQL mediante la implementación de PostgreSQL. Para un mejor despliegue y facilidad tanto en desarrollo como en producción, se utiliza ​AWS para almacenar la base de datos, mediante el servicio ​RDS. Así se tiene la base de datos en la nube, disponible en todo momento.

**Arquitectura a nivel Frontend (Angular 7)**



* **Modules**: se usan para exportar funcionalidades propias hacia otros módulos para ser utilizados.
* **Templates, Directives and Data Binding:** combina HTML con Angular Markup y modificar elementos HTML antes de mostrarlos. Las directivas de plantilla proporcionan lógica de programa, y ​​el marcado vinculante conecta los datos de su aplicación y el DOM. ***Event Binding***se usa para enlazar eventos a su aplicación y responder a la entrada del usuario en el entorno de destino actualizando los datos de su aplicación. ***Data Binding*** se utiliza para pasar datos de la clase de componente y le permite interpolar valores que se calculan a partir de los datos de su aplicación en el HTML.
* **Services and dependency injection: *Services*** son clases que se crean para datos o lógica que no tiene una vista específica y se desea compartir con otros componentes.
* **Roting:** es un servicio de angular que proporciona la navegación de la aplicación por url