***FUNCIONAMIENTO DEL PROYECTO VACUNAS***

***BASES DE DATOS***

La primera sección es muy importante, en esta están las carpetas y archivos de configuración de la base de datos por lo cual haremos un apartado especial, debido a que esta parte es de vital importancia para poder acceder a la persistencia, intentaremos ser los mas detallado posible

Comencemos

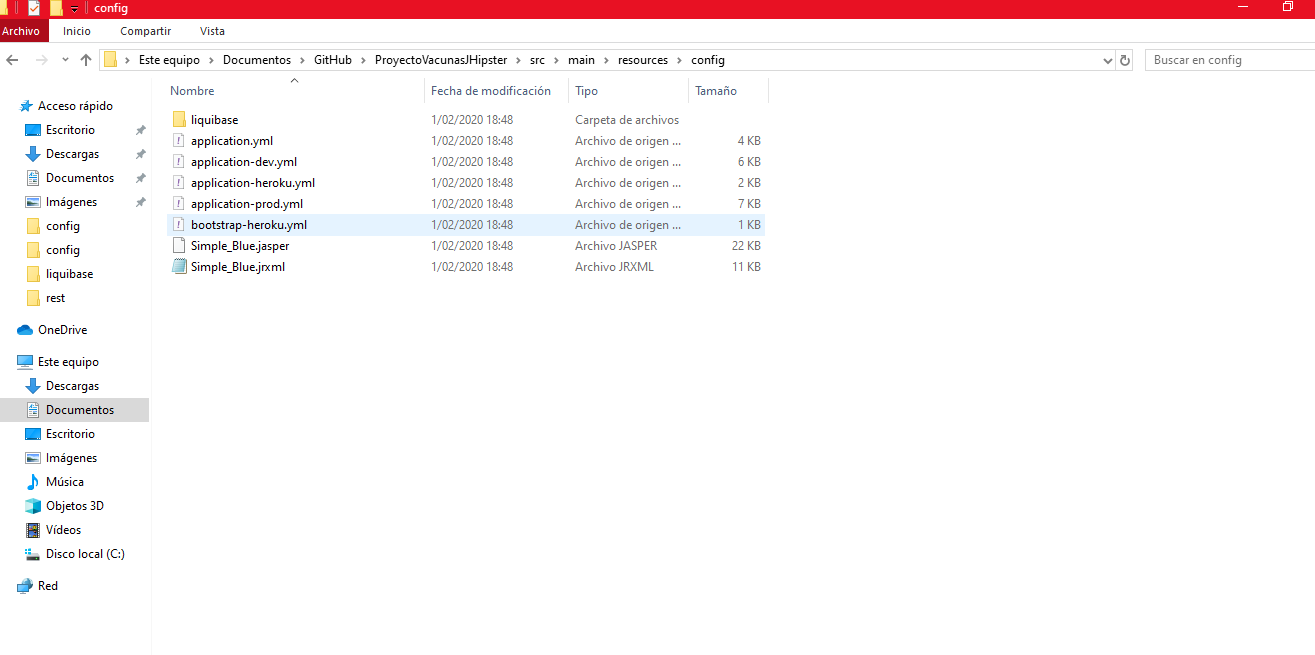


Figura Bases de datos I

Los archivos de configuración se encuentran en la siguiente dirección:

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\resources\config

El archive donde se encuentra la conexión de la base de datos es en

**application-dev.yml** cuando abrimos el archivo no mostrara lo siguiente véase

Figura de Bases de Datos 2



Figura de Bases de Datos 2 en este archivo va la configuración de la base de datos

***ESTRUCTURA DEL PROYECTO***

Con este documento está el funcionamiento y como esta construido el proyecto Vacuna elaborado con el Framework Hípster ,Spring Boot , Hibérnate ,Jpa

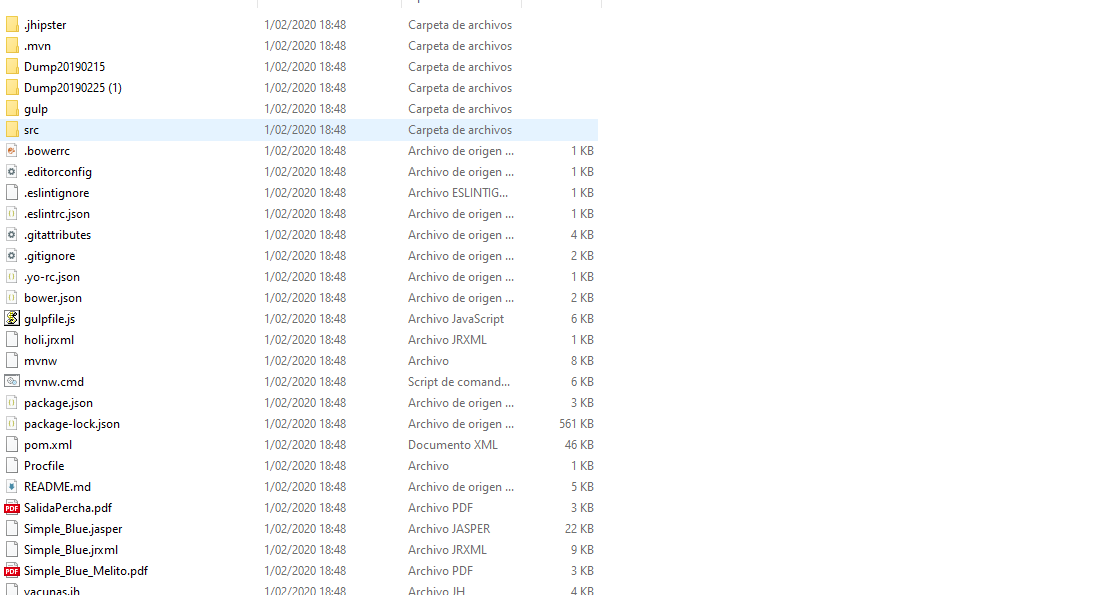


Figura 1 : Estructura del proyecto completo

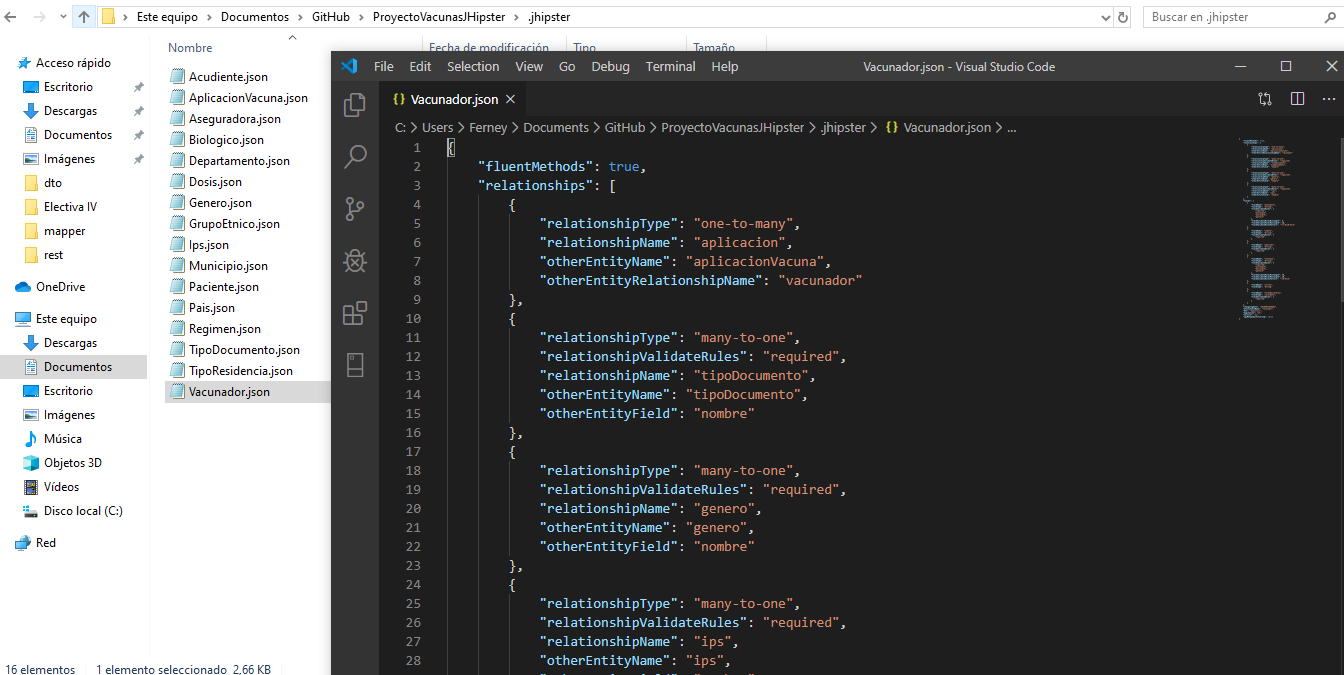


Figura 2: estamos dentro de la carpeta Hípster , la cual contiene el modelado de cada una de las tablas de la base de datos , cada archivo(tabla) es un archivo .json, ejemplo :Vacunador. Json

***MAVEN***

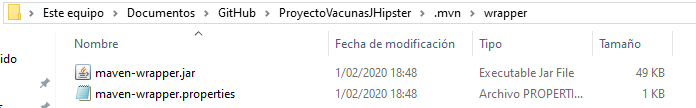


Tabla 3

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\.mvn\wrapper contiene un archivo que genera Maven

***SQL***

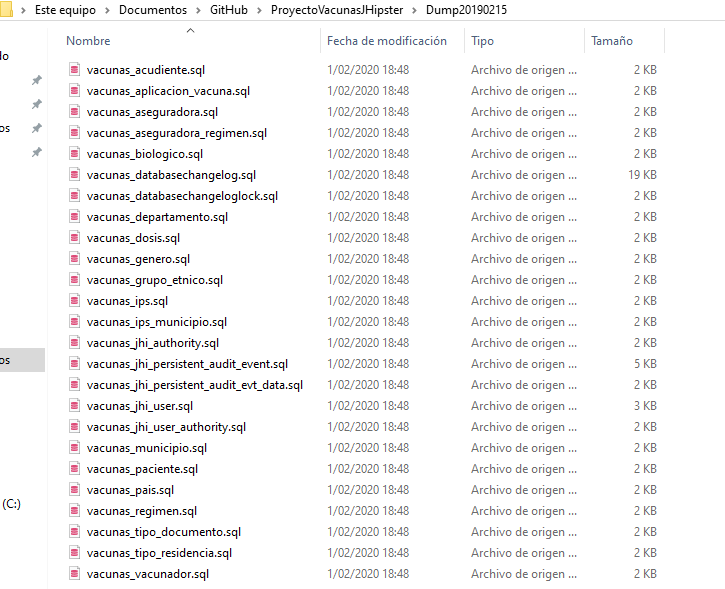
****

Tabla 4

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\Dump20190215 contiene archivos .sql

***JS***

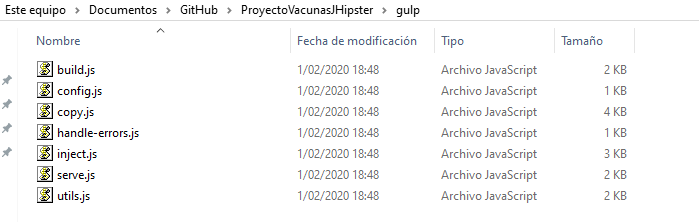


Tabla 5

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\gulp

contiene archivos JavaScript(js)

***DOCKER***

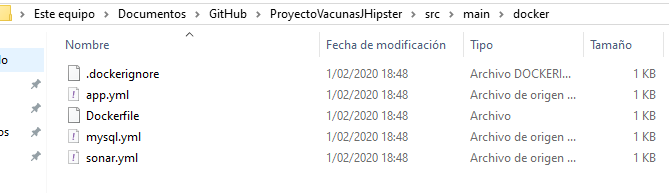


Tabla 6

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\docker

Contiene el Docker con su configuración

***JAVA EE***

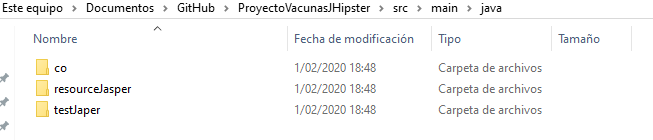


Tabla 7

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java

***DENTRO DEL BACKEND***

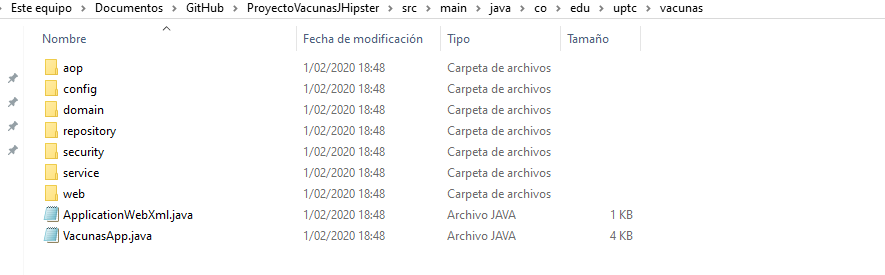


Tabla 8

Contiene toda la lógica del proyecto Vacunas esta es la estructura que se debe preocupar el desarrollador

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas

***LOGGING***

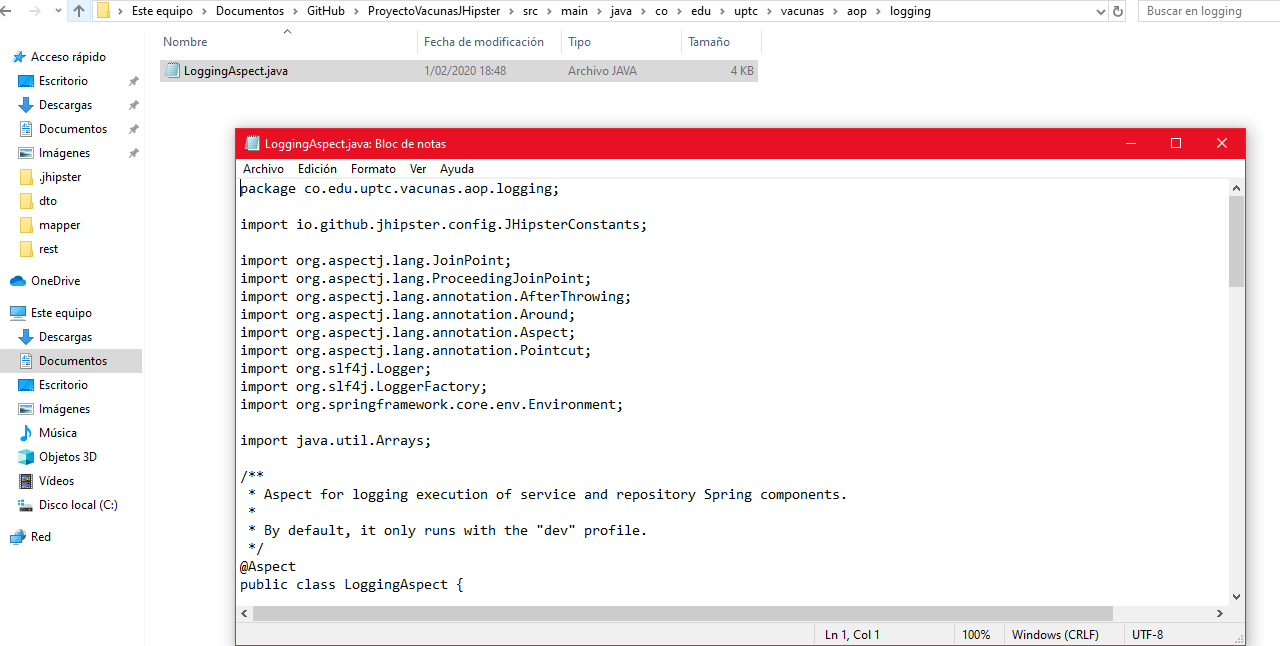


Tabla 9

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\aop\logging

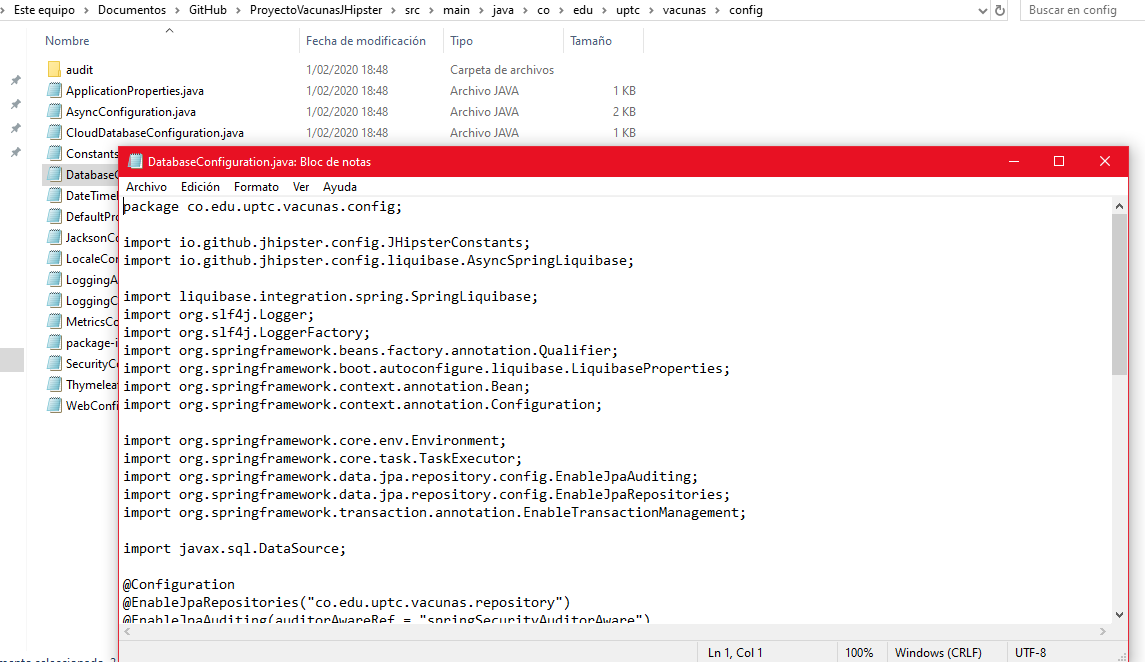


Tabla 10

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\config

En esta carpeta tenemos la configuración de nuestro proyecto , en la imagen se ve la configuración de la base de datos

***CREACION DE DOMAIN***

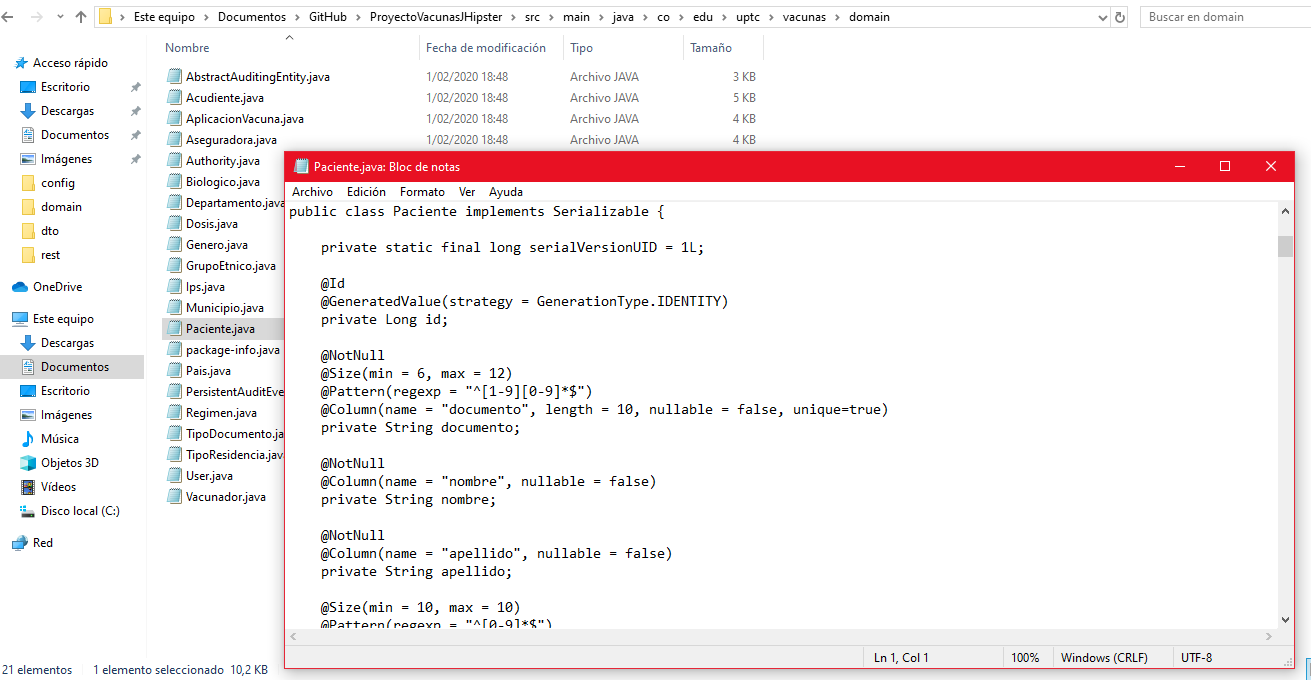


Tabla 10

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\domain

En la table numero 10 tenemos la creación de las clases , y configuración de las entidades de para la base de datos , se ve la clase de Paciente.java

***CONFIGURACION DE REPOSITORY***

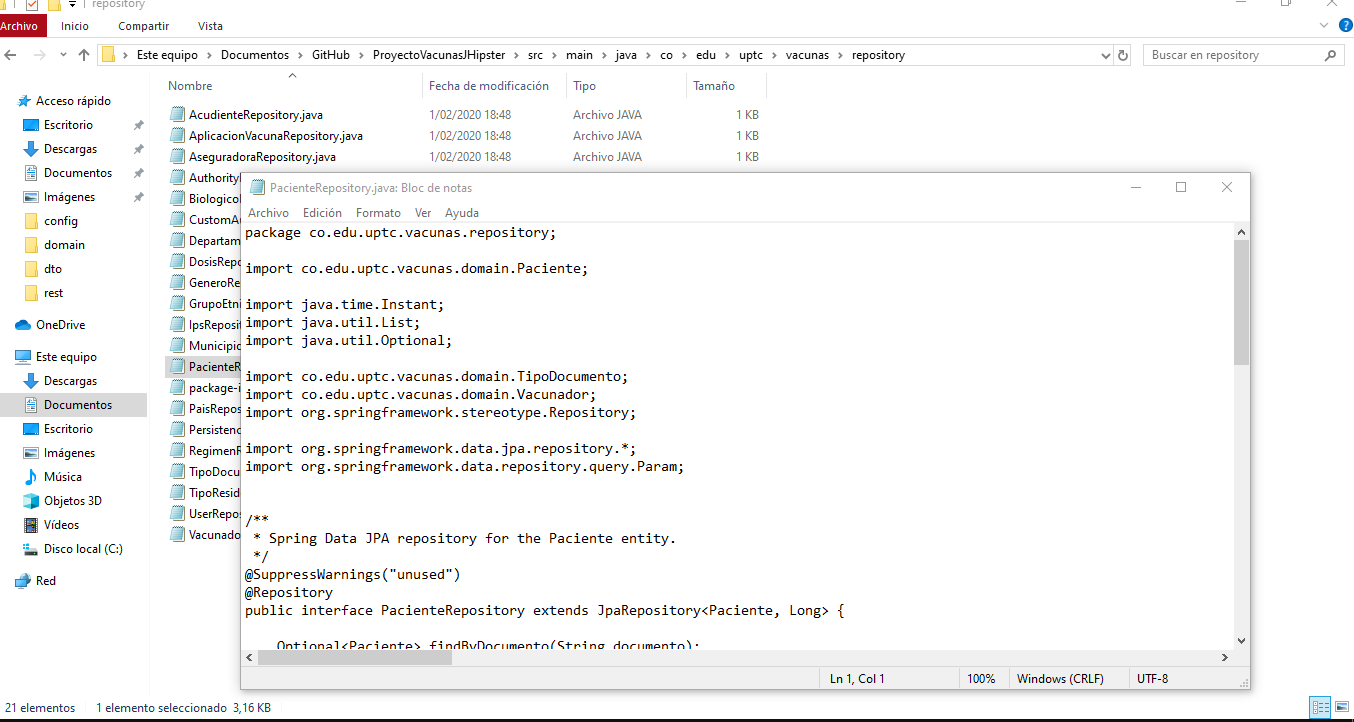


Tabla 11

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\repository

Esta el paquete repository la cuales tiene las clases que extienden de JPARepository <Paciente,Long> en este caso seguimos con el ejemplo de paciente y tenemos la imagen de la interface PacienteRepository

***SPRING SECURITY***

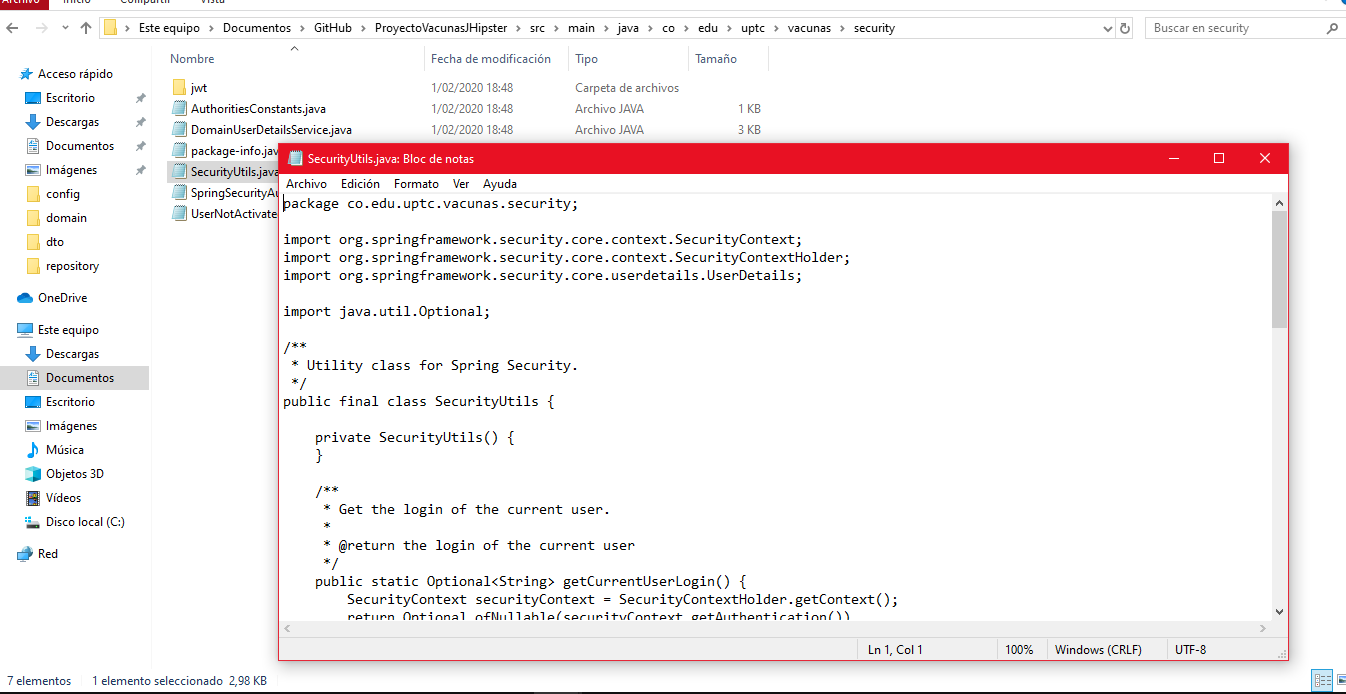


Tabla 12 estamos dentro de la carpeta security

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\security

En esta carpeta esta la configuración de la seguridad del Backend . de la base de datos ,Token y todo lo que con lleva a la seguridad de nuestra aplicación

***DENTRO DE SERVICE***

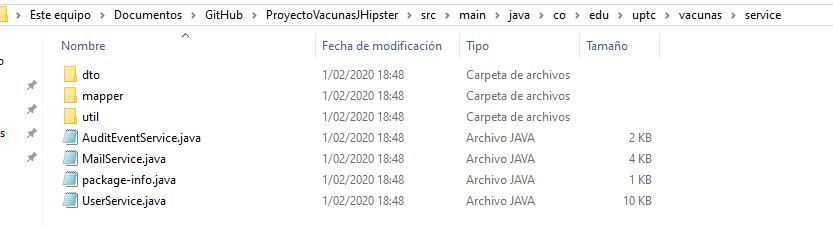


Tabla 13 Estamos dentro de la carpeta vacunas,service la cual contiene diferentes archivos de configuración de los servicios de nuestra aplicación , las carpetas que se ven en la imagen las exploraremos a continuación

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\service

***CREACION DE LOS DTOS’S***

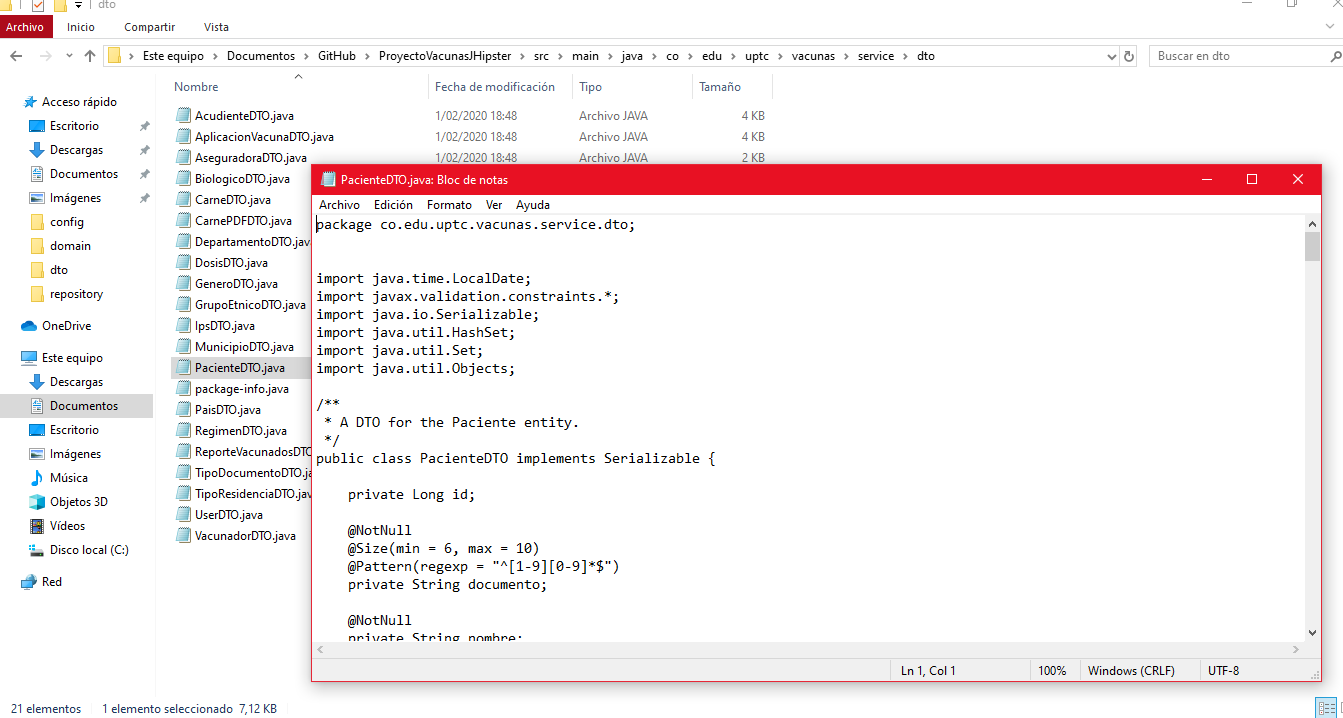


Tabla 14 Como lo dijimos en la tabla numero 13 vamos a explorar cada una de las carpetas que se encuentran en service , la primera de ellas es la carpeta dto, la cual contiene todas las clases DTO de nuestra aplicación , la cual tiene una implementación , nos va a parecer muy similar a las clases que tenemos en domain , pero no es así , es fácil además que continuamos con todo lo relacionado con Paciente , en este caso es PacienteDTO

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\service\dto

***CREACION DE LOS MAPPER***

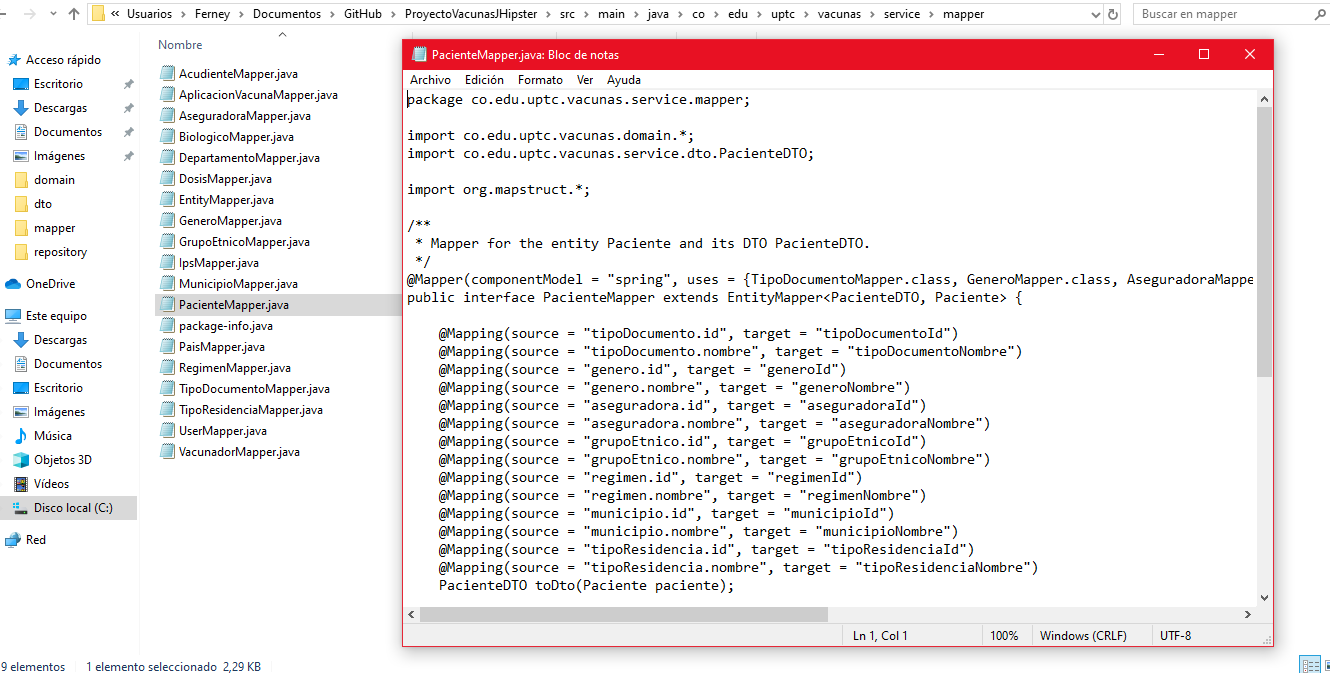


Tabla 15 continuando con la exploración de la carpeta service nos encontramos con la carpeta mapper y dentro encontramos las clases Mapper continuando con el hilo de paciente en este caso nos encontramos con PacienteMapper que es una interface que extiende de ( public interface PacienteMapper extends EntityMapper<PacienteDTO, Paciente>) de EntityMapper y lleva la clase DAO y la clase domain en este caso de domain Paciente.java y de dto la interface pacienteDTO por eso la importancia de esta dos clases

***CREACION DEL REST DE LA APLICACION***

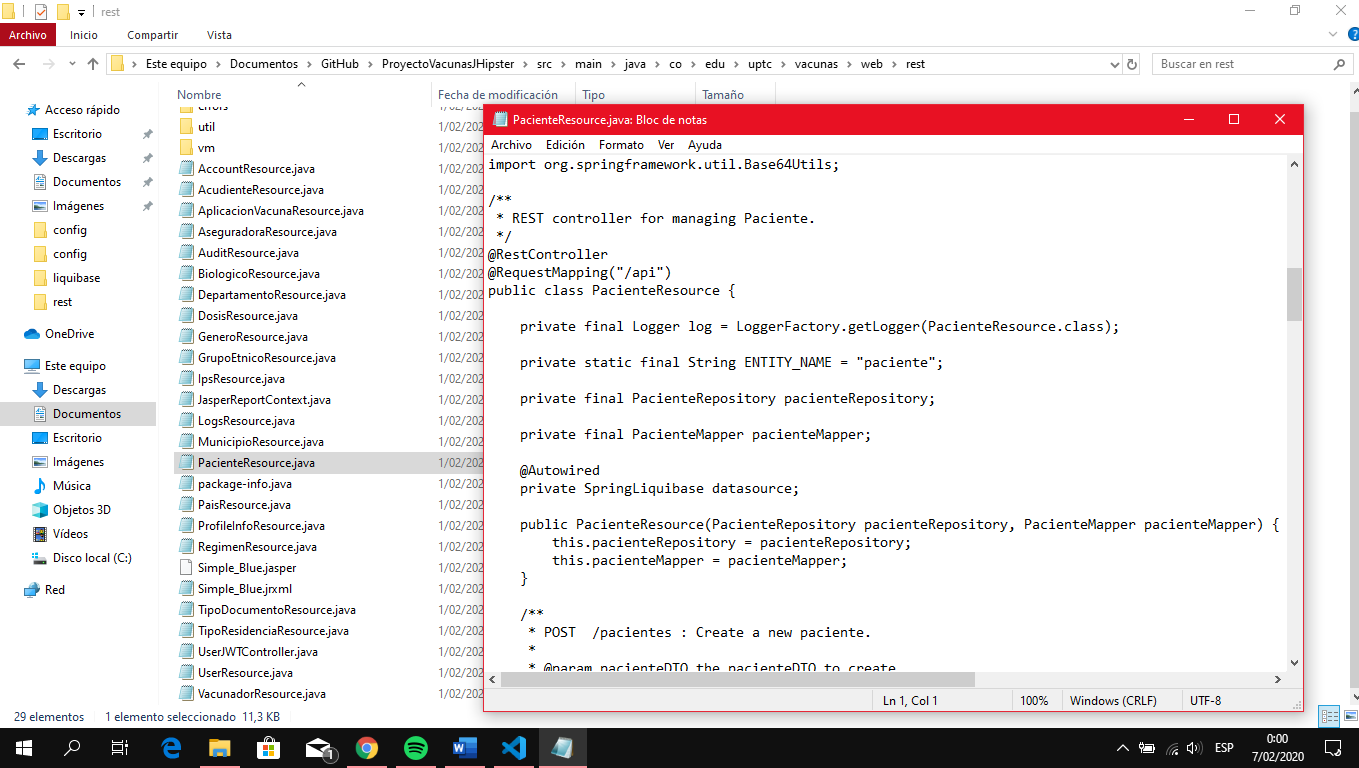


Tabla 16

C:\Users\Ferney\Documents\GitHub\ProyectoVacunasJHipster\src\main\java\co\edu\uptc\vacunas\web\rest

En esta esta dirección encontraremos todo lo correspondiente con nuestra Rest y continuando con paciente , encontraremos a PacienteResource.java, esta clase contiene los métodos HTTP de nuestra aplicación , con el get, delete ,pust, put etc. , este caso serán los métodos de HTTP persona , con esto tenemos casi lista nuestra aplicación del lado del Backend , esta clase Resource será la que se encargara de corre en el servidor y mostrar en algún navegador , al momento de ejecutar cada uno de los métodos en el navegador o en algún software de prueba como por ejemplo Postam , todos los datos en algún archivo Json , estará listo para ser conectada con el Fronted(Angular 8 ) , también en Android , a través de los datos convertidos en json , para continuar con Paciente , veremos toda la información de pacientes en un archivo json , la cual se vera de la siguiente manera

***CLIENTE Y SERVIDOR***

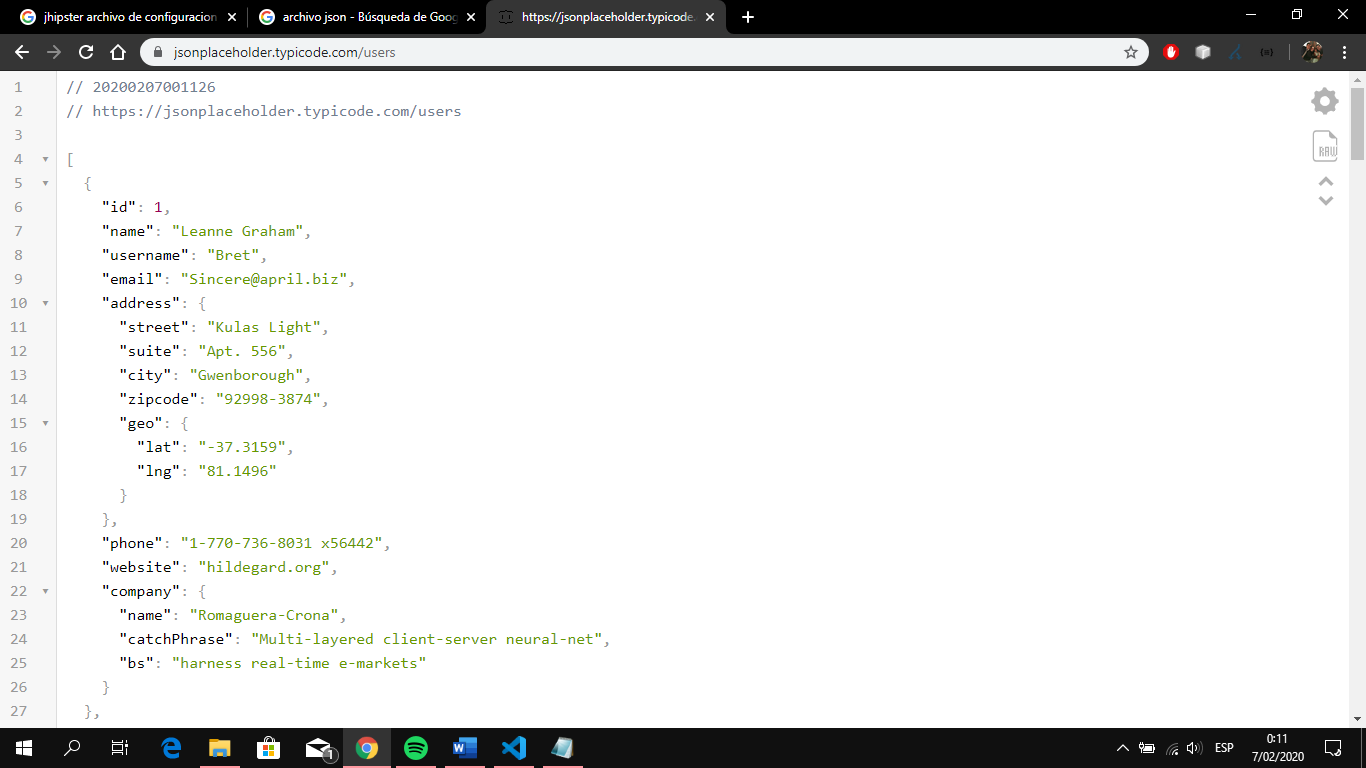


Tabla 17 De esta manera serán mostrados los archivos json después de ejecutar nuestra aplicación Vacunas

***CONCLUSIONES***

La creación será de la siguiente manera

Clase Paciente.java

Interface PacienteDAO.java

Interface PacienteRepository.java

Interface PacienteMapper extends EntityMapper<PacienteDTO, Paciente>).java

Clase PacienteResource.java

De esta manera estará creada nuestra aplicación vacuna del lado del Backend.