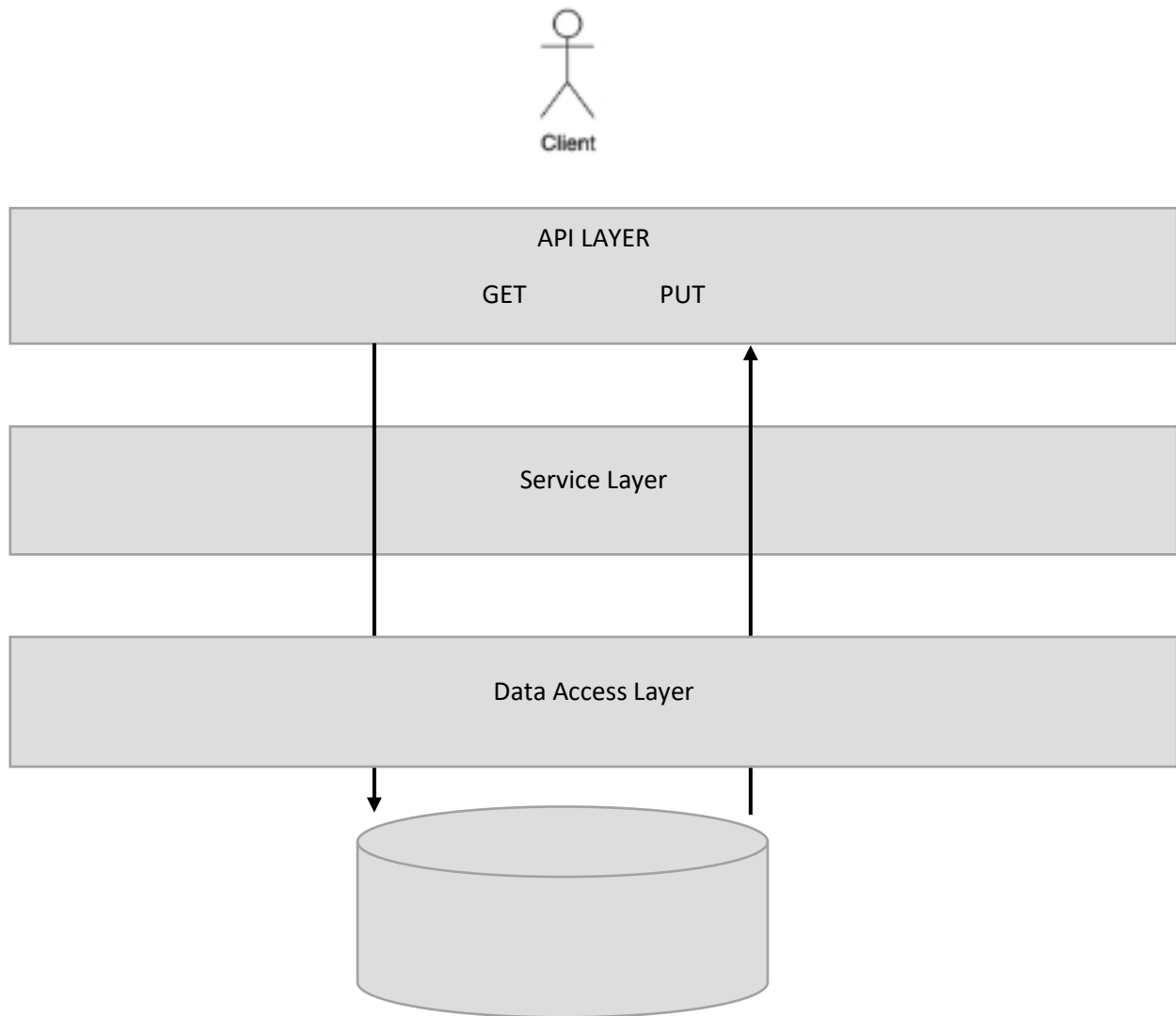


Bitácora Ceiba test.

### Sistema de vacunación para COVID.



La arquitectura implementada en el desarrollo del software “*Sistema para vacunación COVID*” fue mvc (modelo vista controlador) que nos permite separar los datos de la aplicación. Se exponen servicios como GET y POST según las reglas de negocio especificadas mediante el framework SpringBoot de java.

Igualmente se tiene capa de servicio para la lógica de negocio como la verificación de usuarios existen y reconocimiento de usuarios mayores de edad. Para esto se declaran 3 métodos que realizan la lógica de negocio, el método *save* para la creación de usuarios, el método *validYear* para validar que el usuario a registrar sea mayor de edad y el método *finAllUser*. Así mismo se declara un

4to método que utiliza la interfaz *JpaRepository* de la dependencia *Spring Data JPA* para facilitar la persistencia de los datos en base de datos en memoria ya que es un software de prueba.

La capa de acceso de datos es utilizada para convertir los objetos de tipo dominio a entidad y viceversa de esta forma se logra separar responsabilidades dentro de la lógica de persistencia de datos

El api backend es desarrollado en lenguaje java con ayuda del framework SpringBoot, se crea un proyecto de tipo gradle y se descargan las dependencias necesarias, en este caso Spring Initializr permite arrancar ágilmente proyectos de tipo SpringBoot, un proyecto Gradle en lenguaje JAVA mediante empaquetamiento Jar al ser el más en común para empaquetamientos java con tecnología JPA para la persistencia de datos en memoria ya que se ha elegido trabajar con una base de datos volátil para fines de testeo.

Para la creación de servicios web se utiliza la dependencia SpringWeb, esta nos permite agilizar el desarrollo de aplicaciones tipo Spring MVC que es en la cual nos estamos basando para esta ocasión.

DIAGRAMA DE FUJO

