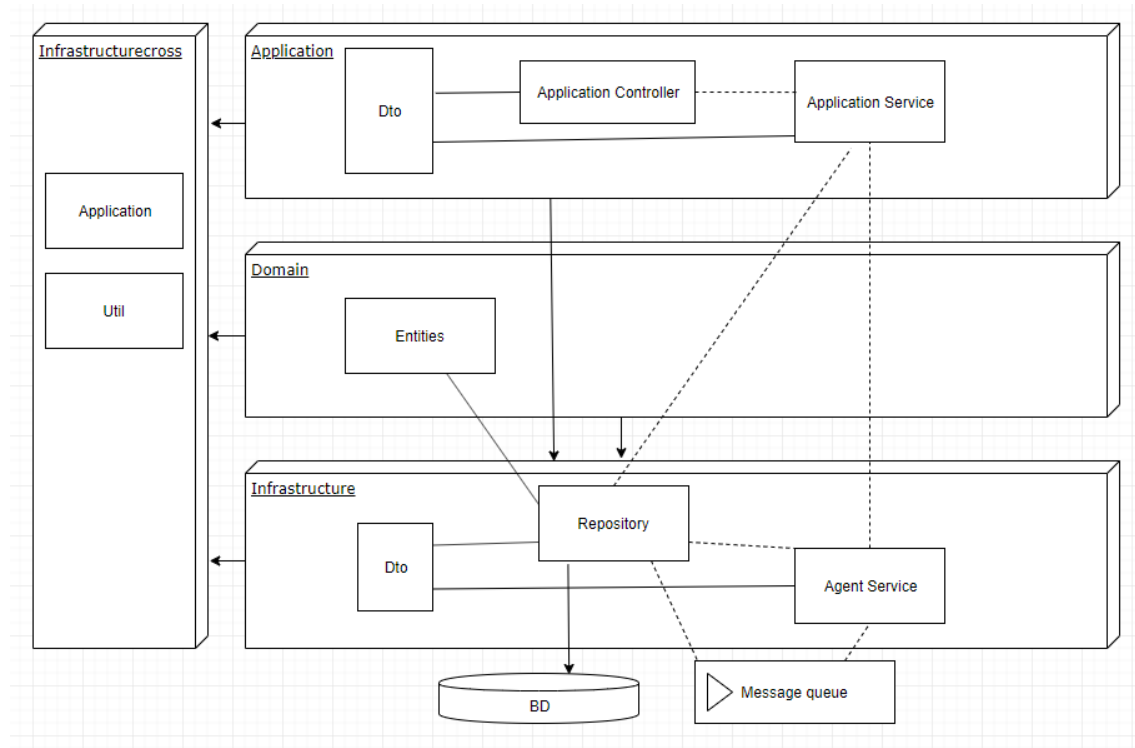


Mutants Project.

Basado en Domain Drive Design.



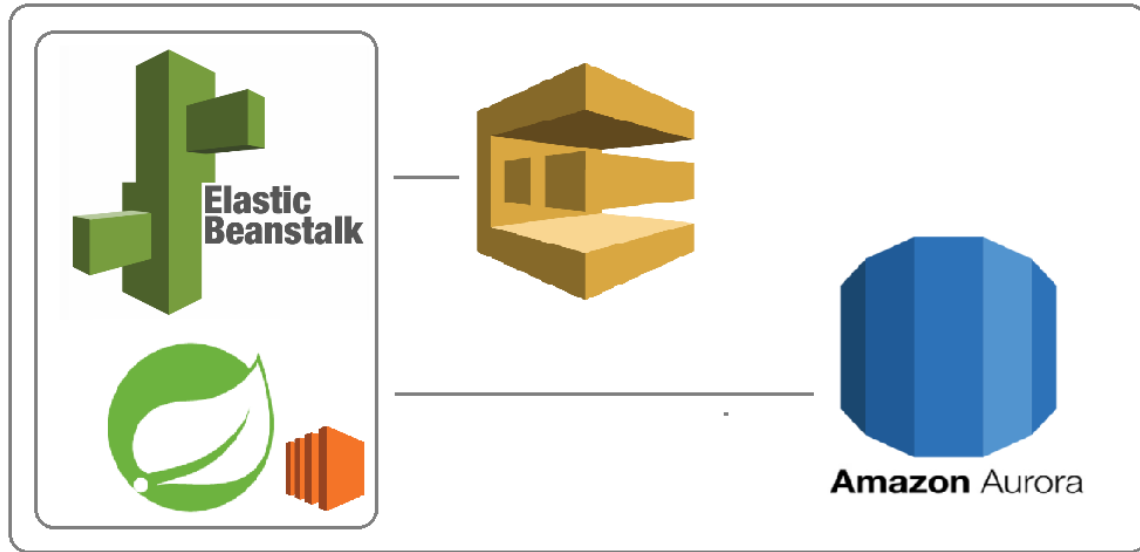
Se utiliza por la separación que existe entre los distintos componentes de la Aplicación, se genera cohesión en métodos de clases y bajo acoplamiento entre ellas.

La línea segmentada significa que existe interfaces entre clases para utilizar sus métodos, y las líneas con flecha indican la dependencia utilizada.

Para la solicitud de ingreso de 1000000 de registros a la base de datos, se utiliza una cola de mensajes. El publicador de estos mensajes y el que consume estos mensajes, es el infrastructure service. De forma asíncrona inserta los mensajes escuchados, a la base de datos AWS Aurora.

Para el Proyecto se utiliza lenguaje de programación Java con Spring Boot.

AWS



Se utiliza AWS como PaS.

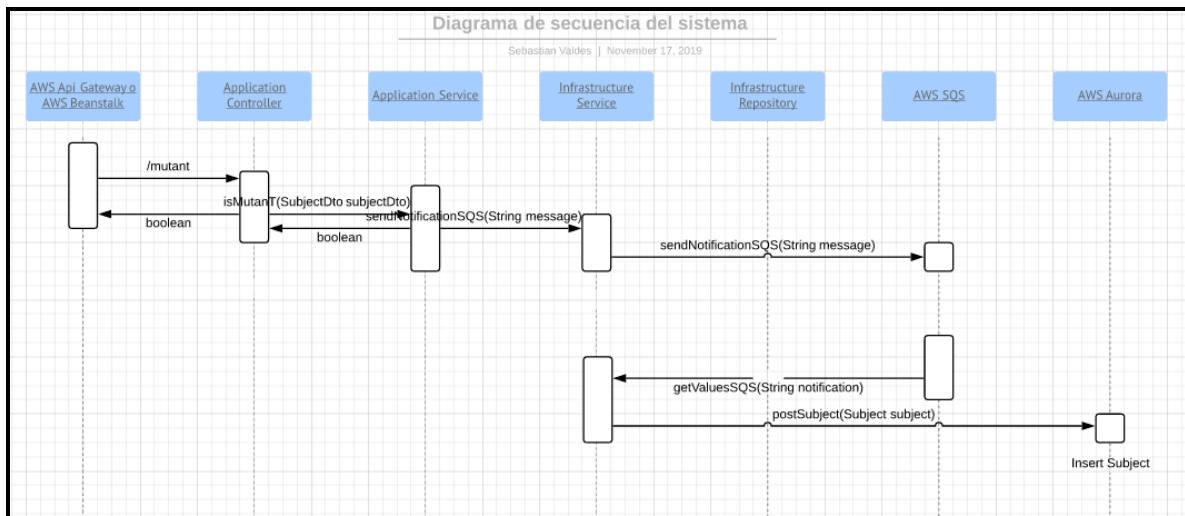
La imagen muestra Spring Boot el cual se genera un WAR que es usado por la Elastic Beanstalk usando una instancia de Amazon EC2.

A su vez se consume cola de mensajes Amazon SQS y una base de datos Amazon Aurora.

Para el error 403 se utilizó la exposición de los servicios por API Gateway, ya que desde Beanstalk no fue posible realizar el control de Httpstatus indicado.

Secuencia de Sistemas.

/mutant



/stats

