Para la primera entrega del experimento 1 se decidió implementar un modelo vista controlador, pero no un MVC clasico, porque los controladores se comportan como actores,por lo tanto se puede decir que se implemento una arquitectura de actores. Esta, se desarrolló en Play para garantizar el desempeño y la latencia requerida y aprovechar que una plicaion PLay es reactiva, lo que facilita el desarrollo de la misma y reduce tiempo de desarrollo.

La arquitectura de actores se basa en una comunicación por mensajes entre actores, además de que se tienen unos supervisores que permiten balancear la carga entre los diferentes actores. En esta arquitectuara los actores que reciben las peticiones del servixio Rest "se liberan" y quedan disponibles para recibir mas peticiones, por el contrario en JAX RS estos no quedan libres hasta que la logica del programa no hay respondido.De esta manera,la arquitectura es más eficiente en cuanto a desempeño. Esto a la vez se da ya que play trabaja de manera asincrónica, manejando un mailbox donde se almacenan las peticiones, lo cual permite a los actores trabajar de la manera anteriormente descrita, evitando que queden bloqueados, esperando respuesta de otros, y asi disminuyendo la latencia de los servicios implementados.

Por otro lado, para las pruebas de los servicios que utilicen un .update() a la base de datos tendrán resultados incongruentes con las pruebas de Jmeter. Esto se debe a que solo se está modificando una tupla lo que lleva a que se encolen los threads ya que estos no pueden modificar todos al tiempo, en las gráficas se puede observar que la latencia aumenta proporcional a la cantidad de threads siendo esta la prueba del argumento anterior.