**Autor:** Héctor Alexander Martínez.

**Fecha:** 03/02/2018

**Descripción de la prueba realizada**

La prueba es un proyecto MAVEN, tiene una configuración básica para su funcionamiento con tecnologías JSF, PRIMEFACES, JEE y Bootstrap, la configuración del proyecto está hecho para JAVA APLICATION SERVER (WlidFly).

El proyecto tiene una estructura sencilla, solo interactúan la vista (index.xhtml) y el controlador (RappiController.java), y me apoyo de 2 clases, para el manejo de la matriz (MatrizOperacionUtil.java) y el de las operaciones (OperacionUtil.java).

**Descripción de las clases**

**RappiController.java**: Es el controlador y dentro de él están las variables, listas y funciones necesarias para interactuar con la vista y validaciones, también realice las funciones que hacen el cálculo del resultado y construcción de la estructura,

**MatrizOperacionUtil.java:** Es una clase que cree a conveniencia para manejar la definición del cubo, contienen tres atributos.

**OperacionUtil.java:** Es otra clase que cree a conveniencia para el manejo de la operaciones UPDATE y QUERY contienen un atributo tipo para la identificación de la operación y 6 valores que necesitan QUERY, y para UPDATE solo utilizo 4.

**Index.xhtml:** es la pantalla principal del proyecto, tienen tecnologías, HTML, JSF, PRIMEFACES, Y BOOTSTRAP, es responsive.



**Respuesta a las preguntas planteadas:**

**Principio de responsabilidad única (Cohesión):** Es destinar las clases para que realicen una única cosa, el propósito de esto de mitigar el efecto de cambios en nuestro código, cuando haya que hacer refactory o mejora

**Código Limpio:** Para mí un código limpio es aquel que cumple los principios de Cohesión, está documentado, evita redundancias de código, se distribuye en capas y las clases tienen bajo acoplamiento.

**Microservicios:** Los microservicios es una forma de desarrollar software en donde se distribuyen diferentes funcionalidades en pequeñas aplicaciones, ejecutándose de forma autónoma y comunicándose entre ellos a través de peticiones HTPP(POST, GET, PUT, DELETE etc...), la principal ventaja que los microservicios ofrecen es la escalabilidad de las aplicaciones y facilitan el manejo de procesos para una aplicación grande, un cambio en un módulo no afectará a los otros, y su desarrollo no necesita de un gran equipo de trabajo, la desventajas es que si se tiene una gran cantidad de microservicios es difícil, tratar de integrarlos todos.