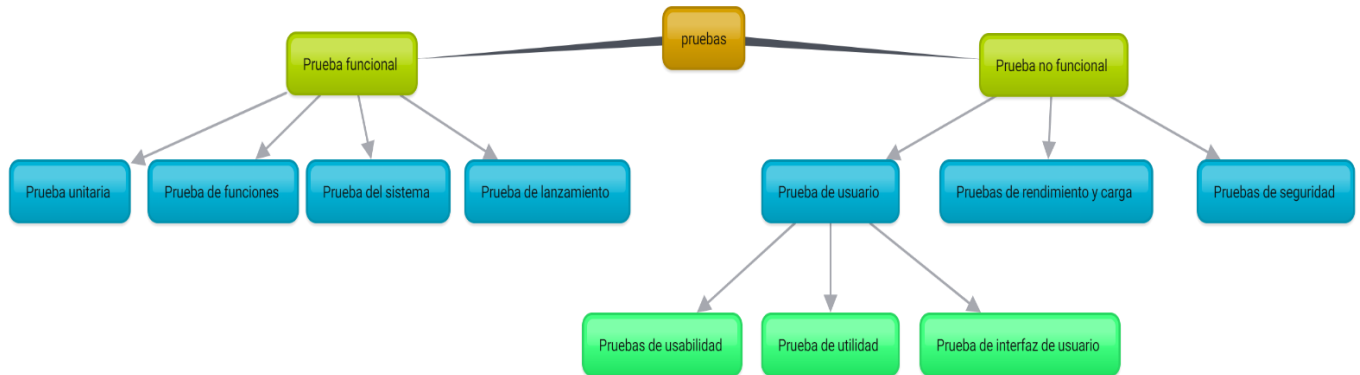


Desarrollo

1_



2_ las recomendaciones que incluye el autor son: programar varias entradas para cada partición de equivalencia, identificar los límites de estas particiones para luego elegir las entradas de estos límites, identificar particiones de equivalencia y crear test de entradas que generen resultados en esas particiones.

3_

Ejemplo: "RUT de persona".

- Permite caracteres numéricos.
- Obligatoriamente debe tener un carácter guion (-).
- Debe tener un largo entre 9 y 10 caracteres
- 19797114-2

Partición de equivalencia	Característica
RUT Correcto 1 Ej. 19797114-2	La entrada incluye solo caracteres numéricos, tiene un carácter guion (-) y un largo de 10 caracteres.
RUT Incorrecto 2 Ej. 8765432-2	La entrada incluye solo caracteres numéricos, tiene un carácter guion (-) y un largo de 9 caracteres.
RUT Incorrecto 1 Ej. 765432-2	La entrada incluye solo caracteres numéricos, tiene un carácter guion (-) pero tiene un largo de 8 caracteres.
RUT Incorrecto 2 Ej. 876543244-2	La entrada incluye solo caracteres numéricos, tiene un carácter guion (-) pero tiene un largo de 11 caracteres.

RUT Incorrecto 3 Ej. 197971142	La entrada incluye solo caracteres numéricos, tiene un largo de 10 caracteres, pero no tiene un carácter guion (-)
RUT Incorrecto 4 Ej. 19797114.2	La entrada incluye solo caracteres numéricos, tiene un largo de 10 caracteres, pero solo se permite el carácter guion (-)

4_ Las sugerencias son:

- Preocuparse por el usuario final a quien va dirigida la aplicación, puesto a que este ocupara el sistema.
- Los casos de prueba tienen que ser fácil de comprender, claro y preciso.
- Ejecutar escenarios con los diferentes roles que tiene el sistema.

5_

- Mockito, permite crear objetos dobles para realizar pruebas. Mi nivel de conocimiento es 1.
- JUnit, herramienta para realizar pruebas unitarias. Mi nivel de conocimiento es 1.