Facultad de ingeniería Informática

Informe trabajo final Taller Programacion 1

Integrantes: Gonzalo León, Ignacio Cantando Flego, Pedro Arias

Fecha de entrega: 13/11/2023

1. Introducción

   Dado un proyecto hecho por un conjunto de programadores nosotros teniamos que testear la aplicación aplicando los conceptos aprendidos durante la cursada.

   Teníamos a nuestro alcance la documentación del JAVADOC, una especie de SRS y el código fuente.

   A continuación se van a notificar los errores encontrados de todas las clases testeadas.

2. Análisis

   a. ClientePuntaje: Sin error
   b. Contratacion: Sin error
   c. EmpleadoPretenso
      i. testCalculaComision1:

1	ticket	ticket ={Locacion: Home Office; Renumeracion: 100.000; Carga Horaria: media; TipodePuesto: junior; ExperienciaPrevia: media; EstudiosCursados:secundario}, EmleadoPretenso={username ="pepe21", password="123", realname="pepe", telefono="12124", apellido="Ramirez":edad=21;. puntaje=-10}

      ii. testCalculaComision5:

5	ticket	ticket ={Locacion: Home Office; Renumeracion: 100.000; Carga Horaria: media; TipodePuesto:Junior; ExperienciaPrevia: media; EstudiosCursados:secundario}, EmleadoPretenso={username ="pepe21", password="123", realname="pepe", telefono="12124", apellido="Ramirez":edad=21;. puntaje=85}

Se puede observar que hay un fallo en el CalculaComision

d. Empleador:

i. testConstructor1:

número de prueba	Datos de entrada	Valor
	username	*"pepe21"*
	password	*"abc"*
	realname	*""*
	telefono	*"5487541"*
	rubro	*"SALUD"*
1	*tipopersona*	*"JURIDICA"*

```
org.junit.ComparisonFailure: El realname no es el mismo
Expected :
Actual   :5487541
```

ii. testConstructor2:

	username	*"pepe21"*
	password	*"abc"*
	realname	*"pepe"*
	telefono	*""*
	rubro	*"SALUD"*
2	*tipopersona*	*"JURIDICA"*

```
org.junit.ComparisonFailure: El realname no es el mismo
Expected :pepe
Actual   :
```

iii. testConstructor3:

	username	*"pepe21"*
	password	*"abc"*
	realname	*"pepe"*
	telefono	*"5487541"*
	rubro	*"SALUD"*
3	*tipopersona*	*"JURIDICA"*

```
org.junit.ComparisonFailure: El realname no es el mismo
Expected :pepe
Actual   :5487541
```

Se puede observar que hay un fallo en el getRealName().

e. Ticket: LimiteInf=5000, LimiteSup=15000

i. Escenario 1: **ESCENARIO 1: TICKET= {Locacion=HOMEOFFICE Carga Horaria = media TipoDePuesto: Junior Estudios: primario Exp: nada Renumeracion=1000 }**

1. testComparacionRenumeracion2:

2	otro	*otro ={Locacion: Presencial; Renumeracion: 10000; Carga Horaria: completa; TipodePuesto:senior; ExperienciaPrevia: media; EstudiosCursados:secundario}*

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :-0.5
Actual   :1.0
```

2. testComparacionTotal1:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :6.0
Actual   :5.0
```

3. testComparacionTotal2:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :0.5
Actual   :2.5
```

4. testGetComparacionCargaHoraria3:

3	otro	*otro ={Locacion: Cualquiera; Renumeracion: 100000; Carga Horaria: extendida; TipodePuesto:managment; ExperienciaPrevia: mucha; EstudiosCursados:tercicario}*

```
java.lang.AssertionError: No coincide  carga horaria
Expected :1.0
Actual   :-1.0
```

ii. Escenario 2: Ticket ={**Locacion= Presencial ,Carga Horaria = completa, TipoDePuesto: Senior, Estudios: secundario, Exp: media, Remuneración=10000**}

1. testComparacionTotal1:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :-2.0
Actual   :-1.5
```

2. testComparacionTotal2:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :6.0
Actual   :5.0
```

3. testComparacionTotal3:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :0.5
Actual   :1.0
```

iii. Escenario 3: Ticket = {**Locacion= Cualquiera, Carga Horaria = extendida, TipoDePuesto: Managment, Estudios: terciario, Exp: mucha, Renumeracion=100000**}

1. testGetComparacionPuesto3:

3	*otro*	*otro ={Locacion: Cualquiera; Renumeracion: 100000; Carga Horaria: extendida; TipodePuesto:managment; ExperienciaPrevia: mucha; EstudiosCursados:tercicario}*

```
java.lang.AssertionError: No coincide  puesto laboral
Expected :1.0
Actual   :-0.5
```

2. testComparacionTotal1:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :-4.0
Actual   :-3.0
```

3. testComparacionTotal2:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :-1.0
Actual   :-2.0
```

4. testComparacionTotal3:

```
java.lang.AssertionError: No coincide el getRenumeracion
Expected :6.0
Actual   :3.5
```

5. testGetComparacionCargaHoraria3:

3	otro	otro ={Locacion: Cualquiera; Renumeracion: 100000; Carga Horaria: extendida; TipodePuesto:managment; ExperienciaPrevia: mucha; EstudiosCursados:tercicario}

```
java.lang.AssertionError: No coincide  carga horaria
Expected :1.0
Actual   :-0.5
```

Se puede observar que hay un fallo en los metodos:
getComparacionCargaHoraria
getComparacionTotal
getComparacionPuesto
getComparacionRenumeracion

f. Agencia:
   i. testRegistroEmpleador6:
   Cuando se ingresa un rubro incorrecto no lanza excepcion ImposibleCrearEmpleadorException

   ii. testRegistroEmpleador8:
   Cuando ya hay un empleador registrado con el mismo username, no lanza excepcion NewRegisterException

   iii. testSetLimitesRenumeracion2:
   Cuando se pone un limite inferior mayor al superior no lanza excepcion LimiteSuperiorRemuneracionInvalidaExceptio n

iv. testGatillarRonda:

	Estado de su lista	tickets	Lista de postulantes
Lista de empleados pretensos	{pepe21,lucas34}	pepe21={Locacion: Home Office; Renumeracion: 150.000; Carga Horaria: media; TipodePuesto:junior; ExperienciaPrevia: baja; EstudiosCursados:secundario} lucas34={Locacion: presencial; Renumeracion: 100.000; Carga Horaria: media; TipodePuesto:junior; ExperienciaPrevia: media; EstudiosCursados:secundario}	Vacia
Lista de empleadores	{luis,pedro}	luis={Locacion: Home Office; Renumeracion: 150.000; Carga Horaria: media; TipodePuesto:junior; ExperienciaPrevia: baja; EstudiosCursados:secundario} pedro={Locacion: presencial; Renumeracion: 100.000; Carga Horaria: media; TipodePuesto:junior; ExperienciaPrevia: media; EstudiosCursados:secundario}	vacia
estadoContratacion	VERDADERO	-	-

En el escenario 6, al terminar la ronda, al empleador no se le penaliza con una resta de 20 puntos, sino que se le suma 20 puntos.

Se puede observar que hay un fallo en los metodos:
RegistroEmpleador
setLimitesRenumeracion
AgenciaFromAgenciaDTO
gatillarRonda

La cobertura de cada camino en el test de caja blanca nos quedo asi: (los caminos con sus valores, y el grafo ciclotímico estan en el otro excel)

3. Conclusión

Se presentaron varios desafíos a la hora de testear, el principal fue detectar en cada método los posibles escenarios con sus respectivas baterías de pruebas. Luego otro gran desafío que tuvimos fue armar el conjunto básico de caminos independientes del ejercicio de caja blanca por el método general.

A pesar de las dificultades, fue una experiencia nueva y muy buena, ya que este trabajo nos ayudó a trabajar en equipo como lo hacen los profesionales y además nos ayudó a entender mejor el concepto de testing.

Nota: Leonel decía que esta materia era muy aburrida pero la verdad que a nosotros nos pareció muy interesante todos los temas aprendidos durante la cursada (salvo el test de GUI).