

Trabajo práctico Final

Grupo 1

Integrantes:

Florio Francisco

Frangolini Matias

Iarritu Pedro

Porfilio Bernardo



Caja Negra

1) Clase Cliente:

1.1)

Clases de equivalencia

| public void setCandidato(Cliente candidato) | | | | |
|--|------------------------|---------|-----------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| candidato | candidato != null | Si | Cumple el contrato | 1 |
| | candidato == null | No | No cumple el contrato | 2 |

Batería de Pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|------------------|---|---|------------------|
| 1 | candidato | new Empleado("Matif", "123123", "Matias", "2235825715", "COMERCIO LOCAL", "FISICA") | candidato = new Empleado("Matif", "123123", "Matias", "2235825715", "COMERCIO LOCAL", "FISICA") | 1 |

1.2)

Clases de equivalencia

| public void setTicket(Ticket ticket) | | | | |
|---|------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| ticket | ticket != null | Si | Cumple contrato | 1 |
| | ticket == null | No | No cumple contrato | 2 |

Batería de Pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|------------------|--|---|------------------|
| 1 | ticket | new Ticket("HOMEOFFICE", 1000, "JORNADA_MEDIA", "JUNIOR", "EXP_MEDIA", "PRIMARIOS"); | ticket = new Ticket("HOMEOFFICE", 1000, "JORNADA_MEDIA", "JUNIOR", "EXP_MEDIA", "PRIMARIOS"); | 1 |



1.3)

Clases de equivalencia

| | | | | |
|--|---------------------------|---------|--------------------|-------------------------------------|
| public void setPuntaje(int puntaje) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| puntaje | puntaje es de tipo int | Si | Cumple contrato | 1 |
| | puntaje no es de tipo int | No | No cumple contrato | 2 |

Batería de Pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|------------------|-------|-----------------|------------------|
| 1 | puntaje | 500 | puntaje == 500 | 1 |

1.4)

Clases de equivalencia

| | | | | |
|--|----------------------------|---------|--------------------|-------------------------------------|
| public void setListaDePostulantes(ArrayList<ClientePuntaje> listaDePostulantes) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| listadePostulantes | listaDePostulantes != null | Si | Cumple contrato | 1 |
| | listaDePostulantes == null | No | No cumple contrato | 2 |

Batería de Pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|--------------------|---------------------------------|--|------------------|
| 1 | listaDePostulantes | new ArrayList<ClientePuntaje>() | listaDePostulantes = new ArrayList<ClientePuntaje>() | 1 |



2) Clase EmpleadoPretenso

2.1)

Clases de equivalencia

| Constructor | | | | |
|-----------------|------------------------|---------|------------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| usserName | username == null | No | No respeta el contrato | 1 |
| | username != null | Si | Respeto el contrato | 2 |
| password | password == null | No | No respeta el contrato | 3 |
| | password != null | Si | Respeto el contrato | 4 |
| realName | realName == null | No | No respeta el contrato | 5 |
| | realname != null | Si | Respeto el contrato | 6 |
| telefono | telefono == null | No | No respeta el contrato | 7 |
| | telefono != null | Si | Respeto el contrato | 8 |
| apellido | apellido == null | No | No respeta el contrato | 9 |
| | apellido != null | Si | Respeto el contrato | 10 |
| edad | edad < 0 | No | No respeta el contrato | 11 |
| | edad > 0 | Si | Respeto el contrato | 12 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|------------------|------------|-----------------------|------------------|
| 1 | username | bernip | username = bernip | 2,4,6,8,10,12 |
| | password | 12345 | password = 12345 | |
| | realName | Bernardo | realname = Bernardo | |
| | telefono | 2235369810 | telefono = 2235369810 | |



| | | | | |
|--|----------|----------|---------------------|--|
| | apellido | Porfolio | apellido = Porfolio | |
| | edad | 22 | edad = 22 | |

2.2.1)

Clases de equivalencia

| | | | | |
|--|---------------------------|---------|---------------------------|--|
| ESCENARIO PUNTAJE NEGATIVO | | | | |
| public double calculaComision(Ti cket ticket) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| ticket | ticket valido | Si | Respeto el contrato | 1 |
| | ticket no valido | No | No respeta el contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|---------------------|--|-------|--------------------|---------------------|
| 1 | Ticket de empleado JUNIOR con remuneracion 1000 | | 800 | |
| 2 | Ticket de empleado SENIOR con remuneracion 3000 | | 2700 | |
| 3 | Ticket de empleado MANAGMENT remuneracion 5000 | | 5000 | |

2.2.2)

Clases de equivalencia

| | | | | |
|--|---------------------------|---------|---------------------------|--|
| ESCENARIO PUNTAJE 10 | | | | |
| public double calculaComision(Ti cket ticket) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| ticket | ticket valido | Si | Respeto el contrato | 1 |
| | ticket no valido | No | No respeta el contrato | 2 |



Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|---|-------|-----------------|------------------|
| 1 | Ticket de empleado JUNIOR con remuneracion 1000 | | 800*0,9 | |
| 2 | Ticket de empleado SENIOR con remuneracion 3000 | | 2700*0,9 | |
| 3 | Ticket de empleado MANAGMENT remuneracion 5000 | | 5000*0,9 | |

2.2.3)

Clases de equivalencia

| ESCENARIO PUNTAJE 100 | | | | |
|---|------------------------|---------|------------------------|-------------------------------------|
| public double calculaComision(Ticket ticket) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| ticket | ticket valido | Si | Respeto el contrato | 1 |
| | ticket no valido | No | No respeta el contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases cubiertas |
|------------------|---|-------|-----------------|------------------|
| 1 | Ticket de empleado JUNIOR con remuneracion 1000 | | 800*0,5 | |
| 2 | Ticket de empleado SENIOR con remuneracion 3000 | | 2700*0,5 | |
| 3 | Ticket de empleado MANAGMENT remuneracion 5000 | | 5000*0,5 | |



3)Clase Ticket

3.1.1)E1

Clases de equivalencia

| (E1)ESCENARIO HOMEOFFICE, 1000, MEDIA, JUNIOR, NADA, PRIMARIO | | | | |
|--|--------------------------------|---------|--------------------|-------------------------------------|
| public double getComparacionLocacion(Ticket otro) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con locacion HOMEOFFICE | SI | Cumple el contrato | 3 |
| | ticket con locacion PRESENCIAL | SI | Cumple el contrato | 4 |
| | ticket con locacion CUALQUIERA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 5 |

3.1.2)E1

Clases de equivalencia

| public double getComparacionRemuneracion(Ticket otro) | | | | |
|--|------------------------|---------|--------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |



| | | | | |
|------|-------------------------------------|----|--------------------|---|
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con remuneracion HASTAV1 | SI | Cumple el contrato | 3 |
| | ticket con remuneracion ENTRE V1YV2 | SI | Cumple el contrato | 4 |
| | ticket con remuneracion MASDEV2 | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 5 |

3.1.3)E1

Clases de equivalencia

| public double getComparacionJornada(Ticket otro) | | | | |
|---|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con jornada MEDIA | SI | Cumple el contrato | 3 |
| | ticket con jornada COMPLETA | SI | Cumple el contrato | 4 |
| | ticket con jornada EXTENDIDA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|-------|-----------------|-------------|
|------------------|------------------|-------|-----------------|-------------|



| | | | | |
|---|--------|---|------|-----|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA,PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA,PRIMARIOS) | -0,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MEDIA,PRIMARIOS) | -1 | 5 |

3.1.4)E1

Clases de equivalencia

| public double getComparacionPuesto(Ticket otro) | | | | |
|--|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con puesto JUNIOR | SI | Cumple el contrato | 3 |
| | ticket con puesto SENIOR | SI | Cumple el contrato | 4 |
| | ticket con puesto MANAGEMENT | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,SENIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,MANAGEMENT,EXP_MEDIA, PRIMARIOS) | -1 | 5 |



3.1.5)E1

Clases de equivalencia

| public double getComparacionEstudios(Ticket otro) | | | | |
|--|---------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con estudios PRIMARIOS | SI | Cumple el contrato | 3 |
| | ticket con estudios SECUNDARIOS | SI | Cumple el contrato | 4 |
| | ticket con estudios TERCARIOS | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOME OFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA, SECUNDARIOS) | 1,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MEDIA, TERCARIOS) | 2 | 5 |

3.1.6)E1

Clases de equivalencia

| public double getComparacionExperiencia(Ticket otro) | | | | |
|---|------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con experiencia | SI | Cumple el | 3 |



| | | | | |
|--|---------------------------------|----|-----------------------|---|
| | NADA | | contrato | |
| | ticket con experiencia MEDIA | SI | Cumple el contrato | 4 |
| | ticket con experiencia MUCHA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOME OFFICE,1000,JORNADA_MED IA,JUNIOR,EXP_NADA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COM PLETA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXT ENDIDA,JUNIOR,EXP_MUCHA, PRIMARIOS) | 2 | 5 |

3.2.1)E2

Clases de equivalencia

| ESCENARIO PRESENCIAL, 3000, COMPLETA, SENIOR, MEDIA, SECUNDARIO | | | | |
|--|---------------------------------------|---------|--------------------|---|
| public double getComparacionLocacion(Ticket otro) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con locacion HOME OFFICE | SI | Cumple el contrato | 3 |
| | ticket con locacion PRESENCIAL | SI | Cumple el contrato | 4 |
| | ticket con locacion CUALQUIERA | SI | Cumple el contrato | 5 |



Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA, JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_MEDIA,J UNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_MEDIA,J UNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 5 |

3.2.2)E2

Clases de equivalencia

| public double getComparacionRemuneracion(Ticket otro) | | | | |
|--|--|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con remuneracion HASTAV1 | SI | Cumple el contrato | 3 |
| | ticket con remuneracion ENTRE V1YV2 | SI | Cumple el contrato | 4 |
| | ticket con remuneracion MASDEV2 | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_M EDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,3000,JORNADA_ME DIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,5000,JORNADA_M EDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 5 |



3.2.3)E2

Clases de equivalencia

| public double getComparacionJornada(Ticket otro) | | | | |
|---|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con jornada MEDIA | SI | Cumple el contrato | 3 |
| | ticket con jornada COMPLETA | SI | Cumple el contrato | 4 |
| | ticket con jornada EXTENDIDA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOME OFFICE,1000,JORNADA_MEDIA, JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA, JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA, JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 5 |

3.2.4)E2

Clases de equivalencia

| public double getComparacionPuesto(Ticket otro) | | | | |
|--|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con puesto JUNIOR | SI | Cumple el contrato | 3 |
| | ticket con puesto SENIOR | SI | Cumple el contrato | 4 |
| | ticket con puesto MANAGEMENT | SI | Cumple el contrato | 5 |



Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,SENIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,MANAGMENT,EXP_MEDIA, PRIMARIOS) | -0,5 | 5 |

3.2.5)E2

Clases de equivalencia

| public double getComparacionEstudios(Ticket otro) | | | | |
|--|---------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con estudios PRIMARIOS | SI | Cumple el contrato | 3 |
| | ticket con estudios SECUNDARIOS | SI | Cumple el contrato | 4 |
| | ticket con estudios TERCARIOS | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -0,5 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA, SECUNDARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MEDIA, TERCARIOS) | 1,5 | 5 |



3.2.6)E2

Clases de equivalencia

| public double getComparacionExperiencia(Ticket otro) | | | | |
|---|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con experiencia NADA | SI | Cumple el contrato | 3 |
| | ticket con experiencia MEDIA | SI | Cumple el contrato | 4 |
| | ticket con experiencia MUCHA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_NADA, PRIMARIOS) | -0,5 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MUCHA, PRIMARIOS) | 1,5 | 5 |

3.3.1)E3

Clases de equivalencia

| public double getComparacionLocacion(Ticket otro) | | | | |
|--|------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |



| | | | | |
|--|-----------------------------------|----|-----------------------|---|
| | ticket con locacion HOMEOFFICE | SI | Cumple el contrato | 3 |
| | ticket con locacion PRESENCIAL | SI | Cumple el contrato | 4 |
| | ticket con locacion CUALQUIERA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 5 |

3.3.2)E3

Clases de equivalencia

| public double getComparacionRemuneracion(Ticket otro) | | | | |
|--|-------------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con remuneracion HASTAV1 | SI | Cumple el contrato | 3 |
| | ticket con remuneracion ENTRE V1YV2 | SI | Cumple el contrato | 4 |
| | ticket con remuneracion MASDEV2 | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 1,3 |



| | | | | |
|---|--------|--|---|---|
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_ME DIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_ME DIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 5 |

3.3.3)E3

Clases de equivalencia

| public double getComparacionJornada(Ticket otro) | | | | |
|---|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con jornada MEDIA | SI | Cumple el contrato | 3 |
| | ticket con jornada COMPLETA | SI | Cumple el contrato | 4 |
| | ticket con jornada EXTENDIDA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ases que aba |
|------------------|------------------|---|-----------------|--------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_M EDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_C OMPLETA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_E XTENDIDA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 1 | 5 |

3.3.4)E3

Clases de equivalencia

| public double getComparacionPuesto(Ticket otro) | | | | |
|--|------------------------|---------|--------|------------------------|
| Dato de | Clases de equivalencia | Aplica? | Motivo | Identificador clase de |



| entrada | | | | equivalencia |
|---------|------------------------------|----|--------------------|--------------|
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con puesto JUNIOR | SI | Cumple el contrato | 3 |
| | ticket con puesto SENIOR | SI | Cumple el contrato | 4 |
| | ticket con puesto MANAGEMENT | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|--|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,SENIOR,EXP_MEDIA, PRIMARIOS) | 1 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,MANAGMENT,EXP_MEDIA, PRIMARIOS) | 1 | 5 |

3.3.5)E3

Clases de equivalencia

| public double getComparacionEstudios(Ticket otro) | | | | |
|--|---------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con estudios PRIMARIOS | SI | Cumple el contrato | 3 |
| | ticket con estudios SECUNDARIOS | SI | Cumple el contrato | 4 |
| | ticket con estudios TERCARIOS | SI | Cumple el contrato | 5 |



Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -2 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA, SECUNDARIOS) | -1,5 | 4 |
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MEDIA, TERCARIOS) | 1 | 5 |

3.3.6)E3

Clases de equivalencia

| public double getComparacionExperiencia(Ticket otro) | | | | |
|---|------------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con experiencia NADA | SI | Cumple el contrato | 3 |
| | ticket con experiencia MEDIA | SI | Cumple el contrato | 4 |
| | ticket con experiencia MUCHA | SI | Cumple el contrato | 5 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_NADA, PRIMARIOS) | -2 | 1,3 |
| 2 | ticket | new Ticket(PRESENCIAL,1000,JORNADA_COMPLETA,JUNIOR,EXP_MEDIA, PRIMARIOS) | -1,5 | 4 |



| | | | | |
|---|--------|---|---|---|
| 3 | ticket | new Ticket(CUALQUIERA,1000,JORNADA_EXTENDIDA,JUNIOR,EXP_MUCHA,PRIMARIOS) | 1 | 5 |
|---|--------|---|---|---|

3.4)

Clases de equivalencia

| public doubleComparacionTotal(Ticket otro) | | | | |
|--|--|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| otro | otro != null | SI | Cumple el contrato | 1 |
| | otro == null | NO | NO cumple contrato | 2 |
| | ticket con HOMEOFFICE, HASTAV1, MEDIA, JUNIOR, NADA, SECUNDARIOS | SI | Cumple el contrato | 3 |
| | | | | |
| | | | | |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | ses que aba |
|------------------|------------------|---|-----------------|-------------|
| 1 | ticket | new Ticket(INDISTINTO, 5000, EXTENDIDA, SENIOR, MUCHA, TERCARIOS) | 2 | 1,3 |

4)Clase Contratación

4.1)

Clases de equivalencia

| | | | |
|--|--|--|--|
| public Contratacion(Emp leador empleador, EmpleadoPretens | | | |
|--|--|--|--|



| | | | |
|------------------------|-------------------------------|-----------------|--------------------|
| o empleado) | | | |
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo |
| Empleador | empleador == null | No | No cumple contrato |
| | Empleador != null | Si | Cumple contrato |
| Empleado | Empleado == null | No | No cumple contrato |
| | Empleado != null | Si | Cumple contrato |

Batería de pruebas

| Identificador | Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|---------------|------------------|------------------|--|------------------------------------|-------------------|
| 1 | 1 | empleador | new Empleador("Matif", "123123", "Matias", "2235825715", "COMERCIO LOCAL", "FISICA") | | 2, 4 |
| 2 | | empleado | new Empleado("Matif", "123123", "Matias", "2235825715", "Frangolini", 21) | this.empleado.realName == "Matias" | |

4.2)

Clases de equivalencia

| | | | | |
|---|-------------------------------|-----------------|--------------------|----------------------|
| public void setFecha(GregorianCalendar fecha) | | | | |
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| fecha | fecha es GregorianCalendar | Si | Cumple contrato | 1 |
| | fecha no es GregorianCalendar | No | No cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-------------------------|---------------------------------|-------------------|
| 1 | fecha | new GregorianCalendar() | fecha = new GregorianCalendar() | 1 |



5) Clase Empleador

5.1)

Clases de equivalencia

| <pre>public Empleador(String usserName, String password, String realName, String telefono, String rubro, String tipoPersona)</pre> | | | | |
|--|---|----------|--------------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| Username | username == null | No | No cumple contrato | 1 |
| | username != null | Si | Cumple contrato | 2 |
| Password | password == null | No | No cumple contrato | 3 |
| | password != null | Si | Cumple contrato | 4 |
| realName | realName == null | No | No cumple contrato | 5 |
| | realname != null | Si | Cumple contrato | 6 |
| telefono | telefono == null | No | No cumple contrato | 7 |
| | telefono != null | Si | Cumple contrato | 8 |
| rubro | rubro no existe en clase Constantes | No | No cumple contrato | 9 |
| | rubro existe en clase constantes | Si | Cumple contrato | 10 |
| tipoPersona | tipopersona no existe en clase Constantes | No | No cumple contrato | 11 |
| | tipopersona existe en clase Constantes | Si | Cumple contrato | 12 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|------------|-----------------------|-------------------|
| 1 | ussername | bernip | ussername = bernip | 2,4,6,8,10,12 |
| | password | 12345 | password = 12345 | |
| | realName | Bernardo | realname = Bernardo | |
| | telefono | 2235369810 | telefono = 2235369810 | |
| | rubro | SALUD | rubro = SALUD | |
| | tipoPersona | FISICA | tipoPersona = FISICA | |

5.2)E1

Clases de equivalencia

| | | | | |
|------------------------|--|--|--|--|
| ESCENARIO SALUD | | | | |
|------------------------|--|--|--|--|



| | | | | |
|---|------------------------|-----------|--------------------|---------------|
| public double calculaComision(Ticket ticket) | | CONSULTAR | | |
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| Ticket | Ticket válido | Si | Cumple el contrato | 1 |
| | Ticket no válido | No | No cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|--|-----------------|-------------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 600 | 1 |

5.2)E2

Clases de equivalencia

| | | | | |
|---|------------------------|-----------|--------------------|---------------|
| ESCENARIO COMERCIO LOCAL | | | | |
| public double calculaComision(Ticket ticket) | | CONSULTAR | | |
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| Ticket | Ticket válido | Si | Cumple el contrato | 1 |
| | Ticket no válido | No | No cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|--|-----------------|-------------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA, PRIMARIOS) | 700 | 1 |

5.2)E2

Clases de equivalencia

| | | | | |
|---|--|-----------|--|--|
| ESCENARIO COMERCIO INTERNACIONAL | | | | |
| public double calculaComision(Ticket ticket) | | CONSULTAR | | |



| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
|-----------------|------------------------|----------|--------------------|---------------|
| Ticket | Ticket válido | Si | Cumple el contrato | 1 |
| | Ticket no válido | No | No cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|---|-----------------|-------------------|
| 1 | ticket | new Ticket(HOMEOFFICE,1000,JORNADA_MEDIA,JUNIOR,EXP_MEDIA,PRIMARIOS) | 800 | 1 |

6) Clase ClientePuntaje

6.1)

Clases de equivalencia

| Constructor | | | | |
|-----------------|------------------------|---------|------------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| Usuario | cliente == null | No | No respeta el contrato | 1 |
| | cliente != null | Si | Respeta el contrato | 2 |
| puntaje | puntaje es double | Si | Respeta el contrato | 3 |
| | puntaje no es double | No | No respeta el contrato | 4 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|---|---|-------------------|
| 1 | Cliente | new EmpleadoPretense("Matias", "123456", "2235825715", "Frangolini", 21) | cliente = new EmpleadoPretense("Matias", "123456", "2235825715", "Frangolini", 21) | 2 |
| 2 | puntaje | 500 | puntaje = 500,0 | 4 |

6.2)

Clases de equivalencia

| | | | | |
|--|--|--|--|--|
| public void setCliente(Cliente cliente) | | | | |
|--|--|--|--|--|



| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
|-----------------|------------------------|---------|-----------------------|-------------------------------------|
| cliente | cliente != null | Si | Cumple el contrato | 1 |
| | cliente == nul | No | No cumple el contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|--|--|-------------------|
| 1 | cliente | new Empleado("Matif", "123456", "Matias", "2235825715", "COMERCIO_LOCAL", "FISICA"); | cliente = new Empleado("Matif", "123456", "Matias", "2235825715", "COMERCIO_LOCAL", "FISICA"); | 1 |

6.3)

Clases de equivalencia

| public void setPuntaje(double puntaje) | | | | |
|--|------------------------|---------|--------------------|-------------------------------------|
| Dato de entrada | Clases de equivalencia | Aplica? | Motivo | Identificador clase de equivalencia |
| puntaje | puntaje es double | Si | Cumple el contrato | 1 |
| | puntaje no es double | No | No cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-------|-----------------|-------------------|
| 1 | puntaje | 100 | puntaje = 100,0 | 1 |

7) Clase Agencia

7.1.1) E1=ESCENARIO LISTAS CON DATOS

Batería de pruebas

| public static Agencia getInstance() | | | | |
|-------------------------------------|------------------|------------------------------------|-------------------------------------|-------------------|
| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
| 1 | Ninguno | Agencia a2 = Agencia.getInstance() | Instancia de las agencias coinciden | -- |

7.1.2) **E1**

Clases de equivalencia

| public void setLimitesRemuneracion(int limiteInferior, int limiteSuperior) | | | | |
|---|-------------------------------|----------|--------------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| limiteInferior | limiteInferior>0 | Si | Cumple contrato | 1 |
| | limiteInferior<0 | Si | Cumple contrato | 2 |
| | limiteInferior>limiteSuperior | No | No cumple contrato | 3 |
| limiteSuperior | limiteSuperior>0 | Si | Cumple contrato | 4 |
| | limiteSuperior<=0 | No | No cumple contrato | 5 |
| | limiteSuperior<limiteInferior | Si | Cumple contrato | 6 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-------|---|-------------------|
| 1 | limiteInferior | 2000 | limiteInferior = 2000; limiteSuperior=5000 | 1,4 |
| | limiteSuperior | 5000 | | |
| 2 | limiteInferior | 2000 | LimiteSuperiorRemuneracion InvalidaException | 1 |
| | limiteSuperior | 1000 | | 4,6 |
| 3 | limiteInferior | -3000 | LimiteInferiorRemuneracionInvalidaException | 2,4 |
| | limiteSuperior | 5000 | | |

7.1.3) **E1**

Clases de equivalencia

| public void match(Empleador empleador, EmpleadoPretensio empleado) | | | | |
|---|--|----------|--------------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| empleador | empleador == NULL | No | No cumple contrato | 1 |
| | empleador debe ser valido y registrado | Si | Cumple contrato | 2 |
| empleado | empleado == NULL | No | No cumple contrato | 3 |
| | empleado debe ser valido y registrado | Si | Cumple contrato | 4 |



Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|---------------------|--------------------|-------------------------------|-------------------|
| 1 | empleador, empleado | new Empleador(...) | Agencia.contrataciones.size+1 | 2,4 |
| | | new Empleado(...) | Empleado.getTicket=NULL | |
| | | | Empleador.getTicket=NULL | |
| | | | Empleado.getPuntajes +10 | |
| | | | Empleador.getPuntajes + 50 | |

7.1.4)E1

Batería de pruebas

| public void generaPostulantes(); | | | | |
|---|------------------|-------|---|-------------------|
| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
| 1 | -- | -- | Empleadores con lista de postulantes ordenada | -- |
| 2 | -- | -- | Empleados con lista de postulantes ordenada | -- |

7.1.5)E1

Clases de equivalencia

| public Cliente registroEmpleado(String nombreUsuario, String pass, String nombreReal, String apellido, String telefono, int edad) | | | | |
|--|--------------------------|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| nombreUsuario | nombreUsuario != NULL | Si | Cumple contrato | 1 |
| | nombreUsuario == NULL | Si | Cumple contrato | 2 |
| pass | pass != NULL | Si | Cumple contrato | 3 |
| | pass == NULL | Si | Cumple contrato | 4 |
| nombreReal | nombreReal != NULL | Si | Cumple contrato | 5 |
| | nombreReal == NULL | Si | Cumple contrato | 6 |
| apellido | apellido != null | Si | Cumple contrato | 7 |
| | apellido == null | Si | Cumple contrato | 8 |
| telefono | telefono != null | Si | Cumple contrato | 9 |
| | telefono == null | Si | Cumple contrato | 10 |
| edad | entero sin restricciones | Si | Cumple contrato | 11 |



| | | | | |
|--|-----------------|--|--|--|
| | por el contrato | | | |
|--|-----------------|--|--|--|

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-----------|----------------------|-------------------|
| 1 | nombreUsuario | berni | NewRegisterException | 1,3,5,7,9,11 |
| | pass | berni1234 | | |
| | nombreReal | Bernardo | | |
| | apellido | Porfilio | | |
| | telefono | 223123456 | | |
| | edad | 22 | | |

7.1.6)E1

Clases de equivalencia

| <pre>public Cliente registroEmpleador(String nombreUsuario, String pass, String nombreReal, String apellido, String telefono, int edad)</pre> | | | | |
|---|--|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| nombreUsuario | nombreUsuario != NULL | Si | Cumple contrato | 1 |
| | nombreUsuario == NULL | Si | Cumple contrato | 2 |
| pass | pass != NULL | Si | Cumple contrato | 3 |
| | pass == NULL | Si | Cumple contrato | 4 |
| nombreReal | nombreReal != NULL | Si | Cumple contrato | 5 |
| | nombreReal == NULL | Si | Cumple contrato | 6 |
| tipoPersona | tipoPersona == "FISICA" o "JURIDICA" | Si | Cumple contrato | 7 |
| | tipoPersona == cualquier otro string no nulo | Si | Cumple contrato | 8 |
| | tipoPersona == NULL | Si | Cumple contrato | 9 |
| telefono | telefono != null | Si | Cumple contrato | 10 |
| | telefono == null | Si | Cumple contrato | 11 |
| rubro | rubro = "SALUD" o "COMERCIO_LOCAL" | Si | Cumple contrato | 12 |
| | o "COMERCIO_INTERNACIONAL" | | | |
| | rubro = cualquier otro String | Si | Cumple contrato | 13 |



| | | | | |
|--|---------------|----|-----------------|----|
| | rubro == NULL | Si | Cumple contrato | 14 |
|--|---------------|----|-----------------|----|

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-----------|----------------------|-------------------|
| 1 | nombreUsuario | berni | NewRegisterException | 1,3,5,7,10,12 |
| | pass | berni1234 | | |
| | nombreReal | Bernardo | | |
| | telefono | 223123456 | | |
| | tipoPersona | "FISICA" | | |
| | rubro | "SALUD" | | |

7.1.7) **E1**

Clases de equivalencia

| public Usuario login(String nombreUsuario, String pass) throws ContraException, NombreUsuarioException | | | | |
|---|------------------------|----------|--------------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| nombreUsuario | nombreUsuario == NULL | NO | No cumple contrato | 1 |
| | nombreUsuario != NULL | SI | Cumple contrato | 2 |
| pass | pass != NULL | SI | Cumple contrato | 3 |
| | pass == NULL | NO | No cumple contrato | 4 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-----------|------------------------|-------------------|
| 1 | nombreUsuario | berni | Usuario == berni | 2,3 |
| | pass | berni1234 | tipoUsuario == 1 | |
| 2 | nombreUsuario | berni | ContraException | 2,3 |
| | pass | mati1234 | | |
| 3 | nombreUsuario | mati | NombreUsuarioException | 2,3 |
| | pass | mati1234 | | |

7.1.8) **E1**

Batería de pruebas

| public void cerrarSesion() | | | | |
|-----------------------------------|------------------|-------|-------------------|-------------------|
| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
| 1 | Ninguno | nada | tipoUsuario == -1 | |

7.1.9) **E1**

Clases de equivalencia

| public gatillarRonda() | | | | |
|-------------------------------|-----------------------------|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| estadoContratacion | estadoContratacion == true | SI | Cumple contrato | 1 |
| | estadoContratacion == false | SI | Cumple contrato | 2 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|--------------------|-----------|---|-------------------|
| 1 | estadoContratacion | VERDADERO | EstadoContratacion == false | 1 |
| | | | Castigo que corresponde | |
| | | | Empleados y empleadores sin postulantes | |
| 2 | estadoContratacion | FALSO | EstadoContratacion == true | 2 |
| | | | Premios y castigos que corresponden | |

7.1.10) **E1**

Clases de equivalencia

| public void crearTicketEmpleado(String locacion, int remuneracion, String jornada, String puesto, String experiencia, String estudios, Cliente cliente) | | | | |
|--|-------------------------|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| locacion | String no nulo ni vacio | Si | Cumple contrato | 1 |
| remuneracion | int>0 | Si | Cumple contrato | 2 |
| jornada | String no nulo ni vacio | Si | Cumple contrato | 3 |
| puesto | String no nulo ni vacio | Si | Cumple contrato | 4 |



| | | | | |
|--------------------|---------------------------------|----|--------------------|----|
| experiencia | String no nulo ni vacío | Si | Cumple contrato | 5 |
| estudios | String no nulo ni vacío | Si | Cumple contrato | 6 |
| cliente | El cliente debe ser un empleado | Si | Cumple contrato | 7 |
| | El cliente no es un empleado | No | No cumple contrato | 8 |
| estadoContratacion | FALSO | Si | Cumple contrato | 9 |
| | VERDADERO | SI | Cumple contrato | 10 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|--------------------|-------------------|------------------------------------|-------------------|
| 1 | locacion | HOME_OFFICE | Se crea un nuevo ticket | 1,2,3,4,5,6,7,9 |
| | remuneracion | 5000 | | |
| | jornada | JORNADA_MEDIA | | |
| | puesto | JUNIOR | | |
| | experiencia | EXP_MUCHA | | |
| | estudios | SECUNDARIOS | | |
| | cliente | new Empleado(...) | | |
| | estadoContratacion | FALSO | | |
| 2 | locacion | HOME_OFFICE | ImposibleModificarTicketsException | 1,2,3,4,5,6,7,10 |
| | remuneracion | 5000 | | |
| | jornada | JORNADA_MEDIA | | |

7.1.11)E1

Clases de equivalencia

| <pre>public void crearTicketEmpleado(String locacion, int remuneracion, String jornada, String puesto, String experiencia, String estudios, Cliente cliente)</pre> | | | | |
|---|------------------------|----------|--------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |



| | | | | |
|--------------------|---------------------------------|----|--------------------|----|
| locacion | String no nulo ni vacio | Si | Cumple contrato | 1 |
| remuneracion | int>0 | Si | Cumple contrato | 2 |
| jornada | String no nulo ni vacio | Si | Cumple contrato | 3 |
| puesto | String no nulo ni vacio | Si | Cumple contrato | 4 |
| experiencia | String no nulo ni vacio | Si | Cumple contrato | 5 |
| estudios | String no nulo ni vacio | Si | Cumple contrato | 6 |
| cliente | El cliente debe ser un empleado | Si | Cumple contrato | 7 |
| | El cliente no es un empleado | No | No cumple contrato | 8 |
| estadoContratacion | FALSO | Si | Cumple contrato | 9 |
| | VERDADERO | Si | Cumple contrato | 10 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|--------------------|-------------------|------------------------------------|-------------------|
| 1 | locacion | HOME_OFFICE | Se crea un nuevo ticket | 1,2,3,4,5,6,7, 10 |
| | remuneracion | 5000 | | |
| | jornada | JORNADA_MEDIA | | |
| | puesto | JUNIOR | | |
| | experiencia | EXP_MUCHA | | |
| | estudios | SECUNDARIOS | | |
| | cliente | new Empleado(...) | | |
| | estadoContratacion | FALSO | | |
| 2 | locacion | HOME_OFFICE | ImposibleModificarTicketsException | 1,2,3,4,5,6,7, 9 |
| | remuneracion | 5000 | | |
| | jornada | JORNADA_MEDIA | | |

7.1.12) **E1**

Clases de equivalencia

| public void eliminarTicket() | | | | |
|-------------------------------------|-------------------------|----------|--------------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| usuarioLogueado | hay un usuario logueado | Si | Cumple contrato | 1 |
| | no hay usuario logueado | No | No cumple contrato | 2 |
| estadoContratacion | VERDADERO | Si | Cumple contrato | 3 |
| | FALSO | Si | Cumple contrato | 4 |

Batería de pruebas

| Número | Datos de entrada | Valor | Salida esperada | Clases que |
|--------|------------------|-------|-----------------|------------|
|--------|------------------|-------|-----------------|------------|



| de prueba | | | | abarca |
|-----------|--------------------|-----------|--|--------|
| 1 | estadoContratacion | VERDADERO | ImposibleModificarTicketException | 1,3 |
| 2 | estadoContratacion | FALSO | UsuarioLogueado.getTicket==NULL usuarioLogueado.getPuntaje no se modifica | 1,4 |

7.2.1) E2=ESCENARIO LISTAS VACÍAS

Clases de equivalencia

| <pre>public Cliente registroEmpleador(String nombreUsuario, String pass, String nombreReal, String apellido, String telefono, int edad)</pre> | | | | |
|---|--|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| nombreUsuario | nombreUsuario != NULL | Si | Cumple contrato | 1 |
| | nombreUsuario == NULL | Si | Cumple contrato | 2 |
| pass | pass != NULL | Si | Cumple contrato | 3 |
| | pass == NULL | Si | Cumple contrato | 4 |
| nombreReal | nombreReal != NULL | Si | Cumple contrato | 5 |
| | nombreReal == NULL | Si | Cumple contrato | 6 |
| tipoPersona | tipoPersona == "FISICA" o "JURIDICA" | Si | Cumple contrato | 7 |
| | tipoPersona == cualquier otro string no nulo | Si | Cumple contrato | 8 |
| | tipoPersona == NULL | Si | Cumple contrato | 9 |
| telefono | telefono != null | Si | Cumple contrato | 10 |
| | telefono == null | Si | Cumple contrato | 11 |
| rubro | rubro = "SALUD" o "COMERCIO_LOCAL" | Si | Cumple contrato | 12 |
| | o "COMERCIO_INTERNACIONAL" | | | |
| | rubro = cualquier otro String | Si | Cumple contrato | 13 |
| | rubro == NULL | Si | Cumple contrato | 14 |



Batería de pruebas

| | Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|---------|------------------|------------------|--------------|------------------------------------|-------------------|
| RECORTE | 1 | nombreUsuario | berni | Cliente con los parametros pasados | 1,3,5,7,10,12 |
| | | pass | berni1234 | | |
| | | nombreReal | Bernardo | | |
| | | telefono | 223123456 | | |
| | | tipoPersona | "FISICA" | | |
| | | rubro | "SALUD" | | |
| | 2 | nombreUsuario | berni | ImposibleCrearEmpleadoException | |
| | | pass | berni1234 | | |
| | | nombreReal | null | | |
| | | telefono | 223123456 | | |
| | | tipoPersona | "FISICA" | | |
| | | rubro | "VERDULERIA" | | |
| | 3 | nombreUsuario | berni | ImposibleCrearEmpleadoException | |
| | | pass | berni1234 | | |
| | | nombreReal | Bernardo | | |
| | | telefono | 223123456 | | |
| | | tipoPersona | "FISICO" | | |
| | | rubro | "SALUD" | | |

7.2.2)E2

Clases de equivalencia

| public Cliente registroEmpleado(String nombreUsuario, String pass, String nombreReal, String apellido, String telefono, int edad) | | | | |
|--|------------------------|----------|-----------------|---------------|
| Dato de entrada | Clases de equivalencia | Aplica ? | Motivo | Identificador |
| nombreUsuario | nombreUsuario != NULL | Si | Cumple contrato | 1 |
| | nombreUsuario == NULL | Si | Cumple contrato | 2 |
| pass | pass != NULL | Si | Cumple contrato | 3 |
| | pass == NULL | Si | Cumple contrato | 4 |
| nombreReal | nombreReal != NULL | Si | Cumple contrato | 5 |
| | nombreReal == NULL | Si | Cumple contrato | 6 |



| | | | | |
|----------|--|----|-----------------|----|
| apellido | apellido != null | Si | Cumple contrato | 7 |
| | apellido == null | Si | Cumple contrato | 8 |
| telefono | telefono != null | Si | Cumple contrato | 9 |
| | telefono == null | Si | Cumple contrato | 10 |
| edad | entero sin restricciones por el contrato | Si | Cumple contrato | 11 |

Batería de pruebas

| Número de prueba | Datos de entrada | Valor | Salida esperada | Clases que abarca |
|------------------|------------------|-----------|--|-------------------|
| 1 | nombreUsuario | berni | Cliente con todos los parámetros pasados | 1,3,5,7,9,11 |
| | pass | berni1234 | | |
| | nombreReal | Bernardo | | |
| | apellido | Porfilio | | |
| | telefono | 223123456 | | |
| | edad | 22 | | |

Errores encontrados en el test de Caja Negra:

testModeloNegocio [5 FAILURES]:

TestAgenciaListasConDatos:

testRegistroEmpleadorNewRegisterException

Failure en Agencia.registroEmpleador. Lanza "ImposibleCrearEmpleadorException", cuando debería lanzar "NewRegisterException".

testGatillarRondaEstadoContratacionFalse

Failure en Agencia.gatillarRonda: "No se aplicó el castigo al empleado".

testGatillarRondaEstadoContratacionTrue

Failure en Agencia.gatillarRonda: "No se aplicó el castigo por no ser elegido en la ronda".

testLoginNombreUsuarioException

Failure en Agencia.login. Lanza "ContraException", cuando debería lanzar "NombreUsuarioException".

testCalculaPremiosCastigosAsignacionesTestCastigos

Failure en Agencia.calculaPremiosAsignacionesCastigos: "El puntaje del empleado no es el correcto"



testModeloDatos [13 FAILURES]

TestEmpleadoPretense

testCalculaComisionSeniorPuntajeMaximo

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionJuniorPuntajeMaximo

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionSeniorPuntaje

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionJunior

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionManagment

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionJuniorPuntaje

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionSenior

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

testCalculaComisionManagmentPuntajeMaximo

Failure en calculaComision : “El cálculo de la comisión es incorrecto”

TestEmpleador:

testConstructor

Failure al recuperar el realName : “El real name no es el esperado expected <[Matias]> but was <[2235825715]>. En este caso no podemos saber si el error se encuentra en el constructor o en el getRealName

TestTicketCasos1:

testComparacionTotal

Failure en getComparacionTotal : “La comparación total no fue la esperada”.

testComparacion12

Failure en getComparacionRemuneracion: “La comparación de remuneraciones no fue la esperada”.

TestTicketCasos3:

testComparacionJornada33

Failure en getComparacionJornada: “La comparación de jornadas no fue la esperada”.

testComparacionPuesto33

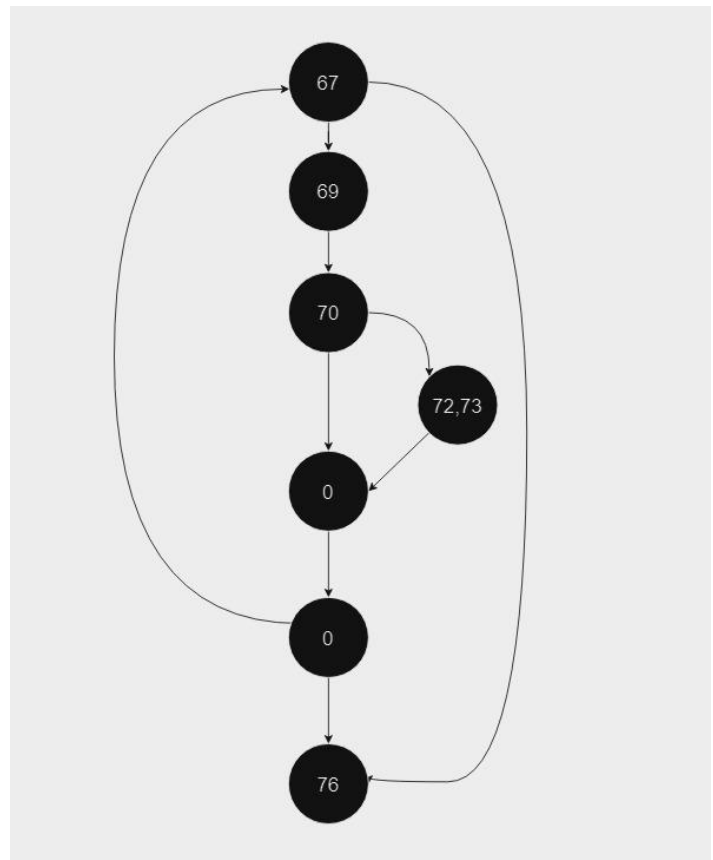
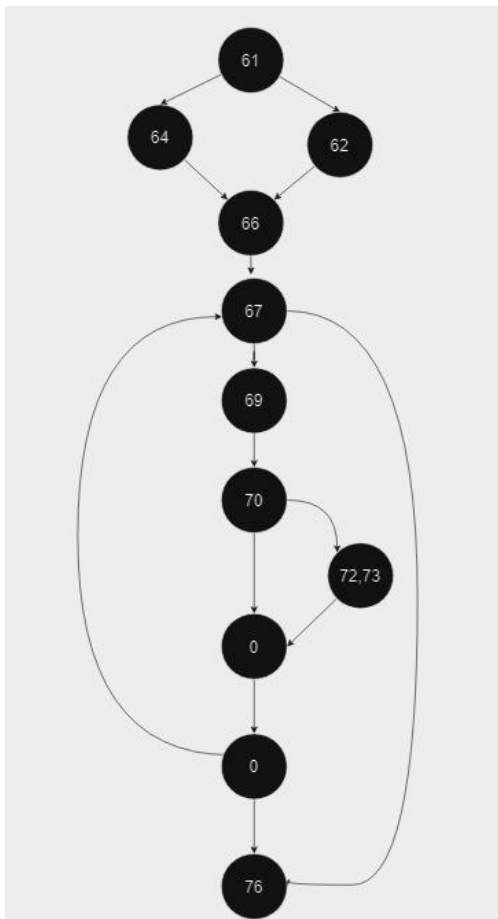
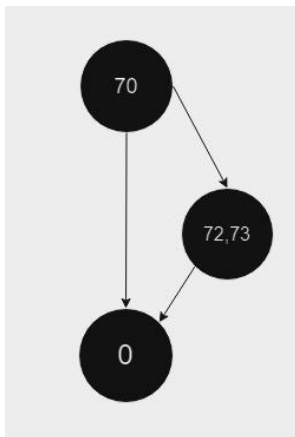


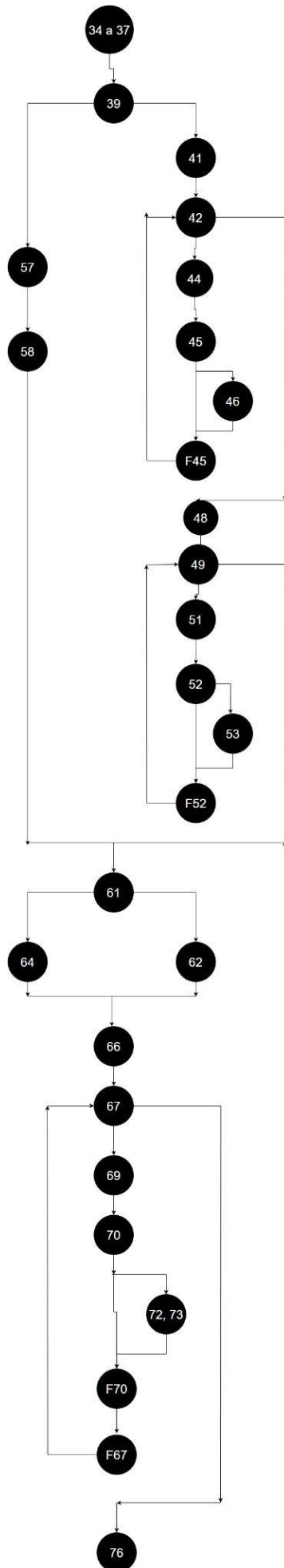
Failure en getComparacionPuesto: “La comparación de puestos no fue la esperada”.

Caja Blanca:

Grafo de control: representa un programa de modo que se observan con claridad los diferentes puntos donde existen cambios de dirección, permitiendo analizar los diversos caminos que llevan del inicio al fin del mismo.

Para realizar este grafo implementamos el **método ascendente**, que consiste en comenzar por instrucciones que están dentro de las iteraciones o condiciones más profundamente anidadas e ir avanzando hacia el exterior. Si se identifica un condicional, se reemplaza por la estructura correspondiente. En forma similar se aplica a iteraciones y case.







Complejidad ciclomática: Cota superior de número de pruebas

$$V(G) = \text{Número de Condiciones} + 1 \rightarrow V(G) = 8 + 1 = 9.$$

Para la determinación de caminos utilizamos el **método simplificado**, que consiste en seleccionar el camino más corto de principio a fin y luego ir buscando segmentos no recorridos hasta completar el número de caminos necesarios.

Este método tiende a recorrer primero los caminos de excepción, los errores, las salidas de emergencia.

| Definición de caminos: MÉTODO SIMPLIFICADO. | | |
|---|---|--------------|
| 1 | 34 a 37, 39, 57, 58, 61, 64, 66, 67, 76 | alcanzable |
| 2 | 34 a 37, 39, 57, 58, 61, 62, 66, 67, 76 | INALCANZABLE |
| 3 | 34 a 37, 39, 57, 58, 61, 62, 66, 67, 69, 70, 67, 76 | INALCANZABLE |
| 4 | 34 a 37, 39, 57, 58, 61, 62, 66, 67, 69, 70, 72, 73, 67, 76 | alcanzable |
| 5 | 34 a 37, 39, 41, 42, 48, 49, 61, 64, 66, 67, 76 | alcanzable |
| 6 | 34 a 37, 39, 41, 42, 44, 45, 42, 48, 49, 61, 64, 66, 67, 76 | alcanzable |
| 7 | 34 a 37, 39, 41, 42, 44, 45, 46, 48, 49, 61, 62, 66, 67, 69, 70, 72, 73, 67, 76 | alcanzable |
| 8 | 34 a 37, 39, 41, 42, 48, 49, 51, 52, 49, 61, 64, 66, 67, 69, 70, 72, 73, 67, 76 | alcanzable |
| 9 | 34 a 37, 39, 41, 42, 48, 49, 51, 52, 53, 49, 61, 64, 66, 67, 69, 70, 72, 73, 67, 76 | alcanzable |



| Diseño de casos de prueba | | | | |
|---------------------------|-----------------------|----------------------------|--|--|
| Id | Objetivo de la prueba | Datos de entrada | Procedimiento | Salida Esperada |
| T1 | Verificar camino 1 | promoPorListaDePostulantes | promoPorListaDePostulantes = false | Devolverá null |
| | | HashMap empleadores | empleadores vacío | |
| | | HashMap empleados | empleados vacío | |
| T2 | Camino 2 inalcanzable | | | |
| T3 | Camino 3 inalcanzable | | | |
| T4 | Verificar camino 4 | promoPorListaDePostulantes | promoPorListaDePostulantes = false | Devuelve el cliente beneficiado de Empleadores |
| | | HashMap empleadores | empleadores con empleadores | |
| | | HashMap empleados | empleados sin empleados | |
| T5 | Verificar camino 5 | promoPorListaDePostulantes | promoPorListaDePostulantes = true | Devolverá null |
| | | HashMap empleadores | empleadores vacío | |
| | | HashMap empleados | empleados vacío | |
| T6 | Verificar camino 6 | promoPorListaDePostulantes | promoPorListaDePostulantes = true | Devolverá null |
| | | HashMap empleadores | it.Empleadores con un empleador sin lista de postulantes | |
| | | HashMap empleados | empleados vacío | |
| T7 | Verificar camino 7 | promoPorListaDePostulantes | promoPorListaDePostulantes = true | Devuelve el cliente beneficiado de Empleadores |
| | | HashMap empleadores | it.Empleadores con empleador | |
| | | HashMap empleados | empleador con lista de postulantes empleados vacío | |
| T8 | Verificar camino 8 | promoPorListaDePostulantes | promoPorListaDePostulantes = true | Devuelve el cliente beneficiado de Empleados |
| | | HashMap empleadores | it.Empleadores vacío | |
| | | HashMap empleados | it.Empleados con empleados. empleado sin lista de postulantes clientes=empleados | |
| | Verificar camino 9 | promoPorListaDePostulantes | promoPorListaDePostulantes = true | Devuelve el cliente beneficiado de |



| | | | | |
|----|--|---------------------|--|-----------|
| T9 | | HashMap empleadores | it.Empleadores vacío | Empleados |
| | | | it.Empleados con empleados. | |
| | | | Empleado con lista de postulantes suma al contador | |
| | | | clientes=empleados | |
| | | HashMap empleados | actualiza puntaje máximo y cliente beneficiado | |

Al diseñar los casos de prueba, siguiendo los caminos determinados por el método simplificado, logramos ejecutar los test de cobertura con los distintos valores de entrada propuestos en cada caso, obteniendo así un 100% de cobertura en el código a testear con Caja Blanca, junto con los valores esperados de salida sin ningún error.

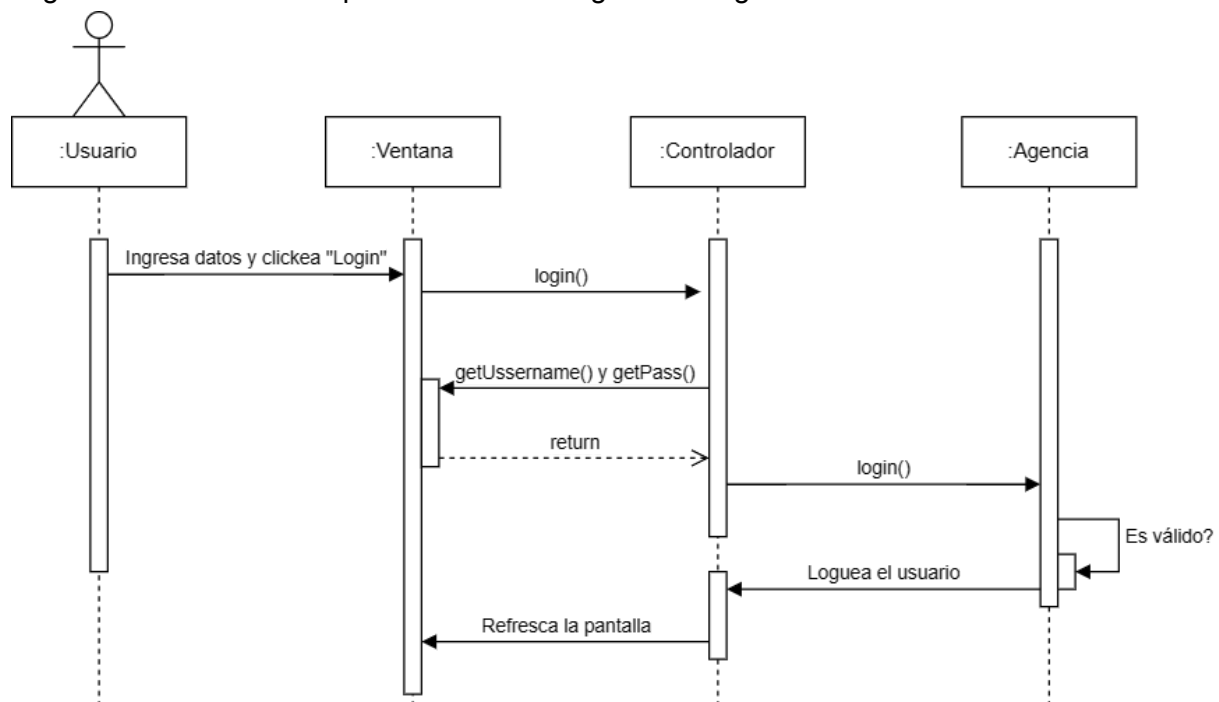
Test de integración:

Dado que en el test de Caja Negra testeamos todos los métodos, cuando llegamos al momento del test de integración (orientado a objetos), los casos de uso y casos de prueba se repetían, y por ende no los realizamos todos.

Nos limitamos a lo siguiente:

En el paquete “testIntegracion”, hay una sola clase, con únicamente 3 métodos, los cuales utilizan mocks.

Seguimos el caso de uso presentado en el siguiente diagrama de secuencia:





Y en base a eso, los siguientes casos de prueba:

| Elemento | Estados |
|-------------|---------------------|
| Controlador | Existe, inexistente |
| Ventana | Existe, inexistente |
| Agencia | Existe, inexistente |

Se consideran las variables username y password:

| | | |
|------|---------|--|
| Mati | loki123 | Nombre y contraseña correctas, de un usuario registrado. |
| Mati | 123456 | Nombre registrado, contraseña incorrecta. |
| Fran | loki123 | Nombre no registrado. |

| Entrada | Condiciones de entrada | Salida esperada |
|---------------|---|--|
| Mati, loki123 | Controlador inexistente | Falla de ejecución |
| Mati, loki123 | Ventana inexistente | Falla de ejecución |
| Mati, loki123 | Agencia inexistente | Usuario inexistente |
| Fran, loki123 | Controlador, ventana y agencia existen. | Mensaje de usuario inexistente |
| Mati, 123456 | Controlador, ventana y agencia existen. | Mensaje de password erróneo |
| Mati, loki123 | Controlador, ventana y agencia existen. | Login exitoso del usuario. Refresca pantalla y muestra panel de cliente. |

Luego, para el resto de los casos de prueba de test de integración, no codeamos, ya que en el código de caja negra se incluyen todos los test. Sin embargo, a continuación se presenta un caso de uso:

Caso de uso:

Un usuario registra un EmpleadoPretense y un Empleador en el sistema. Luego loguea con cada uno de ellos y crea un Ticket. Luego loguea como Administrador y gatilla una ronda.



| Elemento | Estado |
|-------------|---------------------|
| Agencia | Existe, inexistente |
| Controlador | Existe, inexistente |
| Ventana | Existe, inexistente |

Para simplificar el registro de Empleado y Empleador, se utilizan las variables username y password.

| | | | |
|-----------|-------|-----------|--|
| Empleado | Mati | loki123 | Nombre y contraseña válidos, registra usuario. |
| Empleado | Mati | | Falta contraseña. |
| Empleado | | fran123 | Falta nombre. |
| Empleador | Pedro | 123456 | Nombre y contraseña válidos, registra usuario. |
| Empleador | Berni | | Falta contraseña. |
| Empleador | | berni1234 | Falta nombre. |

Para el ticket utilizamos las variables remuneración y locación.

| | | |
|------|-------------|----------------------------------|
| 2000 | Home office | Remuneración y locación válidas. |
|------|-------------|----------------------------------|

Casos de prueba:

| Entrada | Condiciones de entrada | Salida esperada |
|---|---|---|
| Empleado Mati, loki123 | Controlador inexistente | Falla de ejecución. |
| Empleado Mati, loki123 | Ventana inexistente | Falla de ejecución. |
| Empleado Mati, loki123 | Agencia inexistente | Falla al registrar. |
| Empleado(Mati, loki123). Empleador(Pedro, 123456) TicketEmpleado(1500, Home office) TicketEmpleador(2000, Home office) | Controlador, ventana y agencia existen. | Ambos clientes registrados. Ambos ticket creados. Al gatillar ronda se crea la lista de candidatos. |



| | | |
|-------------------------|---|------------------------------|
| Empleado(Mati,). | Controlador, ventana y agencia existen. | Mensaje de password erróneo. |
| Empleador(Pedro,) | Controlador, ventana y agencia existen. | Mensaje de password erróneo. |
| Empleado(, fran123) | Controlador, ventana y agencia existen. | Mensaje de usuario erróneo. |
| Empleador(, berni1234) | Controlador, ventana y agencia existen. | Mensaje de usuario erróneo. |

Test de persistencia:

Para la prueba de la persistencia, pensamos en 3 casos, la persistencia sin que exista el archivo, la persistencia con archivo pero solo cargando los datos de límite superior e inferior y la persistencia con listas de empleados/empleadores llenas.

Para el primer caso, verificamos que al utilizar el método de la Agencia llamado guardarAgencia(); se cree el archivo .xml correctamente. Además de verificar que sea imposible despersistir sin que exista el archivo, es decir que el método cargarAgencia(); retorne false.

Para el segundo caso, creamos una agencia, le seteamos los límites inferior y superior y luego persistimos la agencia. Cuando despersistimos para verificar el funcionamiento de la persistencia descubrimos que el límite inferior no se persiste correctamente, por lo que puede ocasionar fallas futuras.

En el último caso cargamos datos en los HashMap de empleados y empleadores y verificamos que los datos en este caso se persisten correctamente.

Test GUI:

Para el test de GUI, utilizamos la clase Robot de java la cual interactúa con la ventana simulando eventos de mouse y teclado, lo cual permite simular un usuario humano, y automatizar las diferentes pruebas que deseemos hacer.

En este caso testeamos los 4 paneles que tiene disponible el sistema. Empezando por el panel de login. En el mismo testeamos que el botón de login se encuentre habilitado sólo si los JText son distintos de vacío. También que se invoque al método login al clicar el botón, y que se devuelva un mensaje en caso de alguna falla al loguear. +

Luego testeamos el panel de registro, al cual accedemos clickeando el botón "Registrar".

En este panel, por lógica, tuvimos que realizar recortes a la hora de las pruebas. Ya que si pretendíamos abarcar todas las posibles combinaciones del panel esto sería muy tedioso, ya que por ejemplo, existen en el panel 7 JTextField, que pueden ser llenados todos o algunos, y en distintos órdenes, por lo cual sería muy complicado testear todo. Al igual que en el panel anterior, se testeó la habilitación/deshabilitación de los JButton y el funcionamiento de los mismos.

El tercer y cuarto panel testeado fue el panel Admin y el panel de Cliente, a los cuales accedimos logueado el usuario correspondiente. En estos casos testeamos de la misma manera, realizando quizá menos recortes que en el panel de registro.



Para el uso de la clase Robot utilizamos la clase TestUtils provista en un ejemplo de la cátedra.

Los errores o fallas encontrados mediante este método de testeo de interfaces gráficas fueron:

1. No registra al empleado en la agencia.
2. Permite registrar un empleado/empleador poniendo una contraseña distinta en “confirmar contraseña”.
3. No devuelve error al registrar un empleador con un usserName igual a uno ya existente.
4. El botón eliminar ticket permanece deshabilitado.

Conclusiones:

Como grupo creemos que realizar el testeo sobre este sistema es una forma muy buena de aprender y relacionarnos con el testing. Ya que es un sistema con varias funcionalidades, distintas capas y que nos permitió realizar testing de distintas naturalezas.

En el caso del test de caja negra, aprendimos a testear sin conocer nada acerca del código fuente del sistema, pudimos verificar la verdadera importancia que tienen los contratos de los métodos que habíamos comenzado a ver en Programación III.

En cuanto a caja blanca, realizamos la construcción de grafos y caminos, para luego desde la IDE trabajar con un test de cobertura, teniendo el código a disposición pudimos identificar los caminos alcanzables e inalcanzables, y escogimos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa obteniendo un 100% de cobertura del mismo..

Lo que más entretenido nos resultó del trabajo en general fue el testeo de GUI, algo que nunca habíamos hecho y nos llamó la atención.

En cuanto a la experiencia en general, no la consideramos tan difícil, si un poco tediosa pero entendemos que es parte del testing. Quizá lo más dificultoso es reconocer los casos de prueba, escenarios, etc. pero no tanto la parte de codificarlos una vez que fueron identificados.