

ANEXO: TRABAJO PRÁCTICO FINAL

TALLER DE PROGRAMACIÓN I

Grupo N°3:

- **Cármenes, Santiago**
- **García, Federico**
- **Orte, Bautista**

Anexo Caja negra	2
Modelo Negocios	3
Modelo Datos	31
Anexo Test Integración	48

Anexo Caja negra

Para los test unitarios se van a considerar los siguientes escenarios. Vamos a enumerarlos con un identificador único porque se harán referencia a ellos a lo largo de todo el documento. De todas maneras, para cada método vamos a aclarar con qué escenarios fueron testeados, ya que estos no aplican para todos los casos.

Escenario	Descripción
1	La agencia tiene clientes
2	La agencia no tiene clientes
3	Agencia tiene clientes y en estado de búsqueda Laboral, se pueden crear y eliminar Tickets
4	Agencia tiene clientes y se encuentra en estado de contratación, se generaron listas de postulantes
5	Empleador tiene ticket1
6	Empleador tiene ticket2
7	Empleador tiene ticket3

Ticket1 = {"ticket.estudios=Primario", "ticket.experiencia=Exp_nada",
"ticket.Jornada=Jornada_media", "ticket.Locacion=Home_office", "ticket.Puesto=Junior",
0<ticket.Remuneracion<V1}

Ticket2= {"ticket.estudios=Secundario", "ticket.experiencia=Exp_media",
"ticket.Jornada=Jornada_completa", "ticket.Locacion=Presencial", "ticket.Puesto=Senior",
V1<ticket.Remuneracion<V2}

Ticket3= {"ticket.estudios=Terciario", "ticket.experiencia=Exp_mucha",
"ticket.Jornada=Jornada_extendida", "ticket.Locacion=Indistinto", "ticket.Puesto=Management",
ticket.Remuneracion<V2}

Modelo Negocios

- void cerrarSesion()

Numero escenario	Descripción
1	La agencia tiene clientes
2	La agencia no tiene clientes

Se va a considerar nada más el caso en el que la agencia tiene clientes, ya que no tiene sentido un logout en el caso de que no haya clientes registrados.

Tabla de Particiones en Clases de Equivalencia (Esc. 1)

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
-	1	si	cumple contrato	

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	tipoUsuario = -1	1

Batería de pruebas (Esc. 2)

El que cierra sesión es el administrador.

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	tipoUsuario = -1	1

- void crearTicketEmpleado(String locacion, int remuneracion, String jornada, String puesto, String experiencia, String estudios, Cliente cliente)

Numero escenario	Descripción
3	Agencia tiene clientes y en estado de búsqueda Laboral, se pueden crear y eliminar Tickets
4	Agencia tiene clientes y se encuentra en estado de contratación, se generaron listas de postulantes

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
locación	= "Home_Office" = "Presencial" = "Cualquiera"	si	cumple contrato	1
remuneración	remuneración > 0	si	cumple contrato	2
jornada	= "Jornada_Completa" = "Jornada_Extendida" = "Jornada_Media"	si	cumple contrato	3
puesto	= "Junior" = "Senior" = "managment"	si	cumple contrato	4
experiencia	= "nada" = "media" "muchacha"	si	cumple contrato	5
estudios	= "primario" = "secundario" = "terciario"	si	cumple contrato	6
cliente	Cliente es un empleado	si	cumple contrato	7
cliente	Cliente no es empleado	no	no cumple contrato	8

Batería de pruebas (Esc. 3)

Número de prueba	Datos de entrada	Valor	Salida Esperada	Clases que abarca
1	<i>locacion</i>	<i>"presencial"</i>	<i>Crea un nuevo ticket para el empleador</i>	1,2,3,4,5,6,7
	<i>remuneracion</i>	<i>70000</i>		
	<i>jornada</i>	<i>"completa"</i>		
	<i>puesto</i>	<i>"junior"</i>		
	<i>experiencia</i>	<i>"nada"</i>		
	<i>estudios</i>	<i>"secundario"</i>		
	<i>cliente</i>	<i>Empleado1</i>		
2	<i>locacion</i>	<i>"presencial"</i>	<i>Elimina el ticket y crea uno nuevo para el empleador</i>	1,2,3,4,5,6,7
	<i>remuneracion</i>	<i>70000</i>		
	<i>jornada</i>	<i>"completa"</i>		
	<i>puesto</i>	<i>"junior"</i>		
	<i>experiencia</i>	<i>"nada"</i>		
	<i>estudios</i>	<i>"secundario"</i>		
	<i>cliente</i>	<i>Empleado2</i>		

Aclaración: Empleado1 no posee ticket y Empleado2 si posee.

Batería de pruebas (Esc.4)

Número de prueba	Datos de entrada	Valor	Salida esperada	Clases que abarca
1	locacion	"presencial"	<i>ImposibleModificarTicketsException</i>	1,2,3,4,5,6,7
	remuneracion	70000		
	jornada	"completa"		
	puesto	"junior"		
	experiencia	"nada"		
	estudios	"secundario"		
	cliente	Empleado1		

Aclaración: Para el esc. 2 no se toman más casos de prueba ya que la salida esperada en este escenario no va a depender del cliente

- **void crearTicketEmpleador(String locacion, int remuneracion, String jornada, String puesto, String experiencia, String estudios, Cliente cliente)**

Aclaración: Empleador1 no posee ticket y Empleador2 si posee.

Numero escenario	Descripción
3	Agencia tiene clientes y en estado de búsqueda Laboral, se pueden crear y eliminar Tickets
4	Agencia tiene clientes y se encuentra en estado de contratación, se generaron listas de postulantes

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
locación	=“Home_Office” =“Presencial” =“Cualquiera”	si	cumple contrato	1
remuneración	remuneración>0	si	cumple contrato	2
jornada	=“Jornada_Completa” =“Jornada_Extendida” =“Jornada_Media”	si	cumple contrato	3
puesto	=“Junior” =“Senior” = “managment”	si	cumple contrato	4
experiencia	=“nada” =“media” “muchacha”	si	cumple contrato	5
estudios	=“primario” =“secundario” =“terciario”	si	cumple contrato	6
cliente	Cliente es un empleador	si	cumple contrato	7
cliente	Cliente no es empleador	no	no cumple contrato	8

Batería de pruebas (Esc. 3)

Número de prueba	Datos de entrada	Valor	Salida Esperada	Clases que abarca
1	<i>locacion</i>	<i>"presencial"</i>	<i>Crea un nuevo ticket para el empleador</i>	1,2,3,4,5,6,7
	<i>remuneracion</i>	<i>70000</i>		
	<i>jornada</i>	<i>"completa"</i>		
	<i>puesto</i>	<i>"junior"</i>		
	<i>experiencia</i>	<i>"nada"</i>		
	<i>estudios</i>	<i>"secundario"</i>		
	<i>cliente</i>	<i>Empleador1</i>		
2	<i>locacion</i>	<i>"presencial"</i>	<i>Elimina el ticket y crea uno nuevo para el empleador</i>	1,2,3,4,5,6,7
	<i>remuneracion</i>	<i>70000</i>		
	<i>jornada</i>	<i>"completa"</i>		
	<i>puesto</i>	<i>"junior"</i>		
	<i>experiencia</i>	<i>"nada"</i>		
	<i>estudios</i>	<i>"secundario"</i>		
	<i>cliente</i>	<i>Empleador2</i>		

Batería de pruebas (Esc.4)

Número de prueba	Datos de entrada	Valor	Salida esperada	Clases que abarca
1	<i>locacion</i>	<i>“presencial”</i>	<i>ImposibleModificarTicketsException</i>	1,2,3,4,5,6,7
	<i>remuneracion</i>	<i>70000</i>		
	<i>jornada</i>	<i>“completa”</i>		
	<i>puesto</i>	<i>“junior”</i>		
	<i>experiencia</i>	<i>“nada”</i>		
	<i>estudios</i>	<i>“secundario”</i>		
	<i>cliente</i>	<i>Empleado1</i>		

Aclaración: Para el esc. 2 no se toman más casos de prueba ya que la salida esperada en este escenario no va a depender del cliente

- **eliminarTicket()**

Numero escenario	Descripción
3	Agencia tiene clientes y en estado de búsqueda Laboral, se pueden crear y eliminar Tickets
4	Agencia tiene clientes y se encuentra en estado de contratación, se generaron listas de postulantes

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
-	-	si	apllca	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	Se elimina el ticket del usuario logueado	1

Batería de pruebas (Esc. 4)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	ImposibleModificarTicketsException	1

- **Usuario login(String nombreUsuario, String pass)**

Numero escenario	Descripción
1	Hay clientes en el sistema
2	No hay clientes en el sistema

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
nombreUsuario	nombre de usuario valido	si	cumple contrato	1
nombreUsuario	nombre de usuario nulo	no	no cumple contrato	2
pass	pass valida	si	cumple contrato	3
pass	pass nulo	no	no cumple contrato	4

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	usuario	= "baucho"	Se retorna un objeto Usuario con el objeto usuario El tipoUsuario se pone en 0	1,3
	contraseña	= "123"		
2	usuario	= "santi"	Se retorna un objeto Usuario con el objeto usuario El tipoUsuario se pone en 1	1,3
	contraseña	= "456"		
3	usuario	= "admin"	Se retorna un objeto Usuario con el objeto usuario El tipoUsuario se pone en 2	1,3
	contraseña	= "admin"		
4	usuario	= "baucho"	ContraException	1,3
	contraseña	= "789"		
5	usuario	= "mati"	NombreUsuarioException	1,3
	contraseña	= "123"		

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	usuario	= "admin"	Se retorna un objeto Usuario con el objeto usuario El tipoUsuario se pone en 2	1,4
	contraseña	= "admin"		
2	usuario	= "admin"	ContraException	1,5
	contraseña	= "123"		
3	usuario	= "santi"	NombreUsuarioException	2,4
	contraseña	= "123"		

- void gatillarRonda()

Numero escenario	Descripción
3	Agencia tiene clientes y en estado de búsqueda Laboral, se pueden crear y eliminar Tickets
4	Agencia tiene clientes y se encuentra en estado de contratación, se generaron listas de postulantes

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
-	-	Si	Cumple contrato	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	se generan nuevos postulantes para los trabajos disponibles y se calculan los premios o castigos se invierte el valor de estadoContratacion	1

Batería de pruebas (Esc. 4)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	-	-	Se buscan matcheos entre empleadores y empleados. Se penaliza a los empleadores que tienen un ticket asociado pero no fueron elegidos Se eliminan todas las conexiones entre candidatos y puestos de trabajo luego de la ronda de contratacion se invierte el valor de estadoContratacion	1

- **void match(Empleador empleador, EmpleadoPretenso empleado)**

Numero escenario	Descripción
1	Hay clientes en el sistema

Tabla de Particiones en Clases de Equivalencia.

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
empleador	empleador válido	si	cumple contrato	1
empleado	empleado válido	si	cumple contrato	2

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	empleador	empleador	Crea una nueva instancia de Contratación, Incrementa el puntaje del empleado en 10 puntos y del empleador en 50 puntos. Calcula las comisiones para el empleado y el empleador. Empleado y empleador ya no están disponibles para matchear	1,2
	empleado	empleado		

- **Cliente registroEmpleado(String nombreUsuario, String pass, String nombreReal, String apellido, String telefono, int edad)**

Numero escenario	Descripción
1	Hay clientes en el sistema
2	No hay clientes en el sistema

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
nombreUsuario	!="admin" & !="santi" & !="fede" & !="baucho" & !=null	si	cumple contrato	1
nombreUsuario	="admin" ="santi" ="fede" ="baucho"	si	cumple contrato	2
nombreUsuario	=null	si	cumple contrato	3
pass	!=null	si	cumple contrato	4
pass	=null	si	cumple contrato	5
nombreReal	!=null	si	cumple contrato	6
nombreReal	=null	si	cumple contrato	7
apellido	!=null	si	cumple contrato	8
apellido	=null	si	cumple contrato	9

telefono	!=null	si	cumple contrato	10
telefono	=null	si	cumple contrato	11
edad	cualquier entero	si	cumple contrato	12

Batería de pruebas (Esc. 1)

Número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	nombreUsuario	=“pepito”	objeto Cliente que es el empleado registrado	1,4,6,8,10,12
	pass	=“789”		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
2	nombreUsuario	=“fedede”	NewRegisterException	2,4,6,8,10,12
	pass	=“345”		
	nombreReal	=“Federico”		
	apellido	=“Garcia”		
	telefono	=“22321”		
	edad	=23		
3	nombreUsuario	=null	ImposibleCrearUsuarioException	3,4,6,8,10,12
	pass	=“789”		

	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
4	nombreUsuario	=“pepito”	ImposibleCrearU suarioException	1,5,6,8,10,12
	pass	=null		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
5	nombreUsuario	=“pepito”	ImposibleCrearU suarioException	1,4,7,8,10,12
	pass	=“789”		
	nombreReal	=null		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
6	nombreUsuario	=“pepito”	ImposibleCrearU suarioException	1,4,6,9,10,12
	pass	=“789”		
	nombreReal	=“Jose”		
	apellido	=null		
	telefono	=“+5492234562170”		

	edad	=45		
7	nombreUsuario	=“pepito”	ImposibleCrearU suarioException	1,4,6,8,11,12
	pass	=“789”		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=null		
	edad	=45		

Batería de pruebas (Esc. 2)

Número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	nombreUsuario	=“pepito”	objeto Cliente que es el empleado registrado	1,4,6,8,10,12
	pass	=“789”		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
2	nombreUsuario	=null	ImposibleCrearUsuarioException	3,4,6,8,10,12
	pass	=“789”		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
3	nombreUsuario	=“pepito”	ImposibleCrearUsuarioException	1,5,6,8,10,12
	pass	=null		
	nombreReal	=“Jose”		
	apellido	=“Perez”		
	telefono	=“+5492234562170”		
	edad	=45		
4	nombreUsuario	=“pepito”		1,4,7,8,10,12

	pass	= "789"	ImposibleCrearUsuarioException	
	nombreReal	= null		
	apellido	= "Perez"		
	telefono	= "+5492234562170"		
	edad	= 45		
5	nombreUsuario	= "pepito"	ImposibleCrearUsuarioException	1,4,6,9,10,12
	pass	= "789"		
	nombreReal	= "Jose"		
	apellido	= null		
	telefono	= "+5492234562170"		
	edad	= 45		
6	nombreUsuario	= "pepito"	ImposibleCrearUsuarioException	1,4,6,8,11,12
	pass	= "789"		
	nombreReal	= "Jose"		
	apellido	= "Perez"		
	telefono	= null		
	edad	= 45		

- **Cliente registroEmpleador(String nombreUsuario, String pass, String nombreReal, String telefono, String tipoPersona, String rubro)**

Numero escenario	Descripción
1	Hay clientes en el sistema
2	No hay clientes en el sistema

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
nombreUsuario	!="admin" & != "santi" & != "fede" & != "baucho" & != null	si	cumple contrato	1
nombreUsuario	="admin" ="santi" ="fede" ="baucho"	si	cumple contrato	2
nombreUsuario	=null	si	cumple contrato	3
pass	!=null	si	cumple contrato	4
pass	=null	si	cumple contrato	5
nombreReal	!=null	si	cumple contrato	6
nombreReal	=null	si	cumple contrato	7
telefono	!=null	si	cumple contrato	8
telefono	=null	si	cumple contrato	9
tipoPersona	="Juridica" ="Fisica"	si	cumple contrato	10

tipoPersona	=null (!="Juridica" & != "Fisica")	si	cumple contrato	11
rubro	="Salud" ="Comercio Local" = "Comercio Internacional"	si	cumple contrato	12
rubro	=null	si	cumple contrato	13
rubro	!="Salud" & !="Comercio Local" & !="Comercio Internacional"	si	cumple contrato	14

Batería de pruebas (Esc. 1)

Número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	nombreUsuario	="pepito"	objeto Cliente que es el empleado registrado	1,4,6,8,10,12
	pass	="789"		
	nombreReal	="Jose"		
	telefono	=" +5492234562170"		
	tipoPersona	="Fisica"		
	rubro	="Salud"		
2	nombreUsuario	="santi"	NewRegisterException	2,4,6,8,10,12
	pass	="456"		
	nombreReal	="Santiago"		
	telefono	="43234"		
	tipoPersona	="Juridica"		
	rubro	="Salud"		

3	nombreUsuario	=null	ImposibleCrearU suarioException	3,4,6,8,1012
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
4	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,5,6,8,10,12
	pass	=null		
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
5	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,4,7,8,10,12
	pass	= "789"		
	nombreReal	=null		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
6	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,4,6,9,10,12
	pass	= "789"		
	nombreReal	= "Jose"		

	telefono	=null		
	tipoPersona	="Fisica"		
	rubro	="Salud"		
7	nombreUsuario	="pepito"	ImposibleCrearU suarioException	1,4,6,8,11,12
	pass	="789"		
	nombreReal	="Jose"		
	telefono	=" +5492234562170"		
	tipoPersona	=null		
	rubro	="Salud"		
8	nombreUsuario	="pepito"	ImposibleCrearU suarioException	1,4,6,8,10,13
	pass	="789"		
	nombreReal	="Jose"		
	telefono	=" +5492234562170"		
	tipoPersona	=null		
	rubro	="CualquierCosa"		
9	nombreUsuario	="pepito"	ImposibleCrearU suarioException	1,4,6,8,10,13
	pass	="789"		
	nombreReal	="Jose"		

	telefono	="+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	=null		
10	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,4,6,8,10,14
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	="+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "CualquierCosa"		

Batería de pruebas (Esc. 2)

Número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	nombreUsuario	= "pepito"	objeto Cliente que es el empleado registrado	1,4,6,8,10,12
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	="+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
2	nombreUsuario	=null	ImposibleCrearU suarioException	3,4,6,8,10,12
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	="+5492234562170"		

	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
3	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,5,6,8,10,12
	pass	= null		
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
4	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,4,7,8,10,12
	pass	= "789"		
	nombreReal	= null		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
5	nombreUsuario	= "pepito"	ImposibleCrearU suarioException	1,4,6,9,10,12
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	= null		
	tipoPersona	= "Fisica"		
	rubro	= "Salud"		
6	nombreUsuario	= "pepito"		1,4,6,8,11,12

	pass	= "789"	ImposibleCrearUsuarioException	
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= null		
	rubro	= "Salud"		
7	nombreUsuario	= "pepito"	ImposibleCrearUsuarioException	1,4,6,8,10,13
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= null		
8	nombreUsuario	= "pepito"	ImposibleCrearUsuarioException	1,4,6,8,10,14
	pass	= "789"		
	nombreReal	= "Jose"		
	telefono	= "+5492234562170"		
	tipoPersona	= "Fisica"		
	rubro	= "CualquierCosa"		

- **void setLimitesRemuneracion(int limiteInferior, int limiteSuperior)**

Numero escenario	Descripción
1	Hay clientes en el sistema
2	No hay clientes en el sistema

Se va a testear en ambos escenarios, aunque no deberían haber salidas distintas.

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
<i>LimiteSuperior</i>	<i>Cualquier número entero</i>	<i>Si</i>	<i>Aplica contrato</i>	<i>1</i>
<i>LimiteInferior</i>	<i>Cualquier número entero</i>	<i>Si</i>	<i>Aplica contrato</i>	<i>2</i>

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	<i>limiteInferior</i>	<i>-2000</i>	<i>LimiteInferiorRemuneracionInvalidaException</i>	1,2
	<i>limiteInferior</i>	<i>5000</i>		
2	<i>limiteInferior</i>	<i>6000</i>	<i>LimiteSuperiorRemuneracionInvalidaException</i>	1,2
	<i>limiteSuperior</i>	<i>2000</i>		
3	<i>limiteInferior</i>	<i>2000</i>	<i>Establece los límites de remuneración</i>	1,2
	<i>limiteSuperior</i>	<i>6000</i>		

Batería de pruebas (Esc 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	<i>limiteInferior</i>	-2000	<i>LimiteInferiorRemuneracion/ nvalidaException</i>	1,2
	<i>limiteInferior</i>	5000		
2	<i>limiteInferior</i>	6000	<i>LimiteSuperiorRemuneracion InvalidaException</i>	1,2
	<i>limiteSuperior</i>	2000		
3	<i>limiteInferior</i>	2000	<i>Establece los límites de remuneración</i>	1,2
	<i>limiteSuperior</i>	6000		

Modelo Datos

- **Empleador Empleador(String usserName, String password, String realName, String telefono, String rubro, String tipoPersona)**

Numero escenario	Descripción
Escenario único	Constructor empleador

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
usserName	usserName!= null	si	cumple contrato	1
password	password!= null	si	cumple contrato	2
realName	realName!= null	si	cumple contrato	3
telefono	telefono!= null	si	cumple contrato	4
rubro	rubro=="Salud" rubro=="Comercio_Local" rubro == "Comercio_Internacional	si	cumple contrato	5
tipoPersona	tipoPersona=="Juridica tipoPersona=="Fisica"	si	cumple contrato	6
usserName	usserName== null	no	no cumple contrato	7
password	password== null	no	no cumple contrato	8
realName	realName== null	no	no cumple contrato	9

telefono	telefono== null	no	no cumple contrato	10
rubro	rubro!="Salud" & rubro!= "Comercio_Local" & rubro != "Comercio_Internacional	no	no cumple contrato	11
tipoPersona	tipoPersona!="Juridica & tipoPersona!="Fisica"	no	no cumple contrato	12

Batería de pruebas (Esc. único)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	usserName	usserName=="Lio"	objeto Empleador	1,2,3,4,5 ,6
	password	password=="123"		
	realName	realName== "Lionel"		
	telefono	telefono=="5492236"		
	rubro	rubro=="Salud"		
	tipoPersona	tipoPersona=="Fisica"		

- **EmpleadoPretenso EmpleadoPretenso(String usserName, String password, String realName, String telefono, String apellido, int edad)**

Numero escenario	Descripción
Escenario único	Constructor empleado

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
usserName	usserName!=null	si	cumple contrato	1
password	password!=null	si	cumple contrato	2
realName	realName!=null	si	cumple contrato	3
telefono	telefono!=null	si	cumple contrato	4
apellido	apellido!=null	si	cumple contrato	5
edad	edad>0	si	cumple contrato	6
usserName	usserName==null	no	no cumple contrato	7
password	password==null	no	no cumple contrato	8
realName	realName==null	no	no cumple contrato	9
telefono	telefono==null	no	no cumple contrato	10

apellido	apellido==null	no	no cumple contrato	11
edad	edad>0	no	no cumple contrato	12

Batería de pruebas (Esc. Único)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	usserName	usserName=="Marto"	objeto EmpleadoPretenso	1,2,3,4,5,6
	password	password=="123"		
	realName	realName== "Martin"		
	telefono	telefono=="2236"		
	apellido	apellido=="Gomez"		
	edad	26		

- **double getComparacionEstudios(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de Particiones de Clases de equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	estudios="Primario" ="Secundario" ="Terciario"	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.estudios="Primario"	1	1
2	otro	otro.estudios="Secundario"	1,5	1
3	otro	otro.estudios="Terciario"	2	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.estudios="Primario"	-0,5	1
2	otro	otro.estudios="Secundario"	1	1
3	otro	otro.estudios="Terciario"	1,5	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.estudios="Primario"	-2	1
2	otro	otro.estudios="Secundario"	-1,5	1
3	otro	otro.estudios="Terciario"	1	1

- **double getComparacionExperiencia(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de particiones de clases de equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	experiencia="nada" ="media" ="muchacha"	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.experiencia="nada"	1	1
2	otro	otro.experiencia="media"	1,5	1
3	otro	otro.experiencia="muchacha"	2	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.experiencia="nada"	-0,5	1
2	otro	otro.experiencia="media"	1	1
3	otro	otro.experiencia="muchacha"	1.5	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.experiencia="nada"	-2	1
2	otro	otro.experiencia="media"	-1,5	1
3	otro	otro.experiencia="mucha"	1	1

- **double getComparacionJornada(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	jornada="Media" ="Completa" ="Extendida"	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.jornada="media"	1	1
2	otro	otro.jornada="completa"	-0,5	1
3	otro	otro.jornada="extendida"	-1	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.jornada="media"	-0,5	1
2	otro	otro.jornada="completa"	1	1
3	otro	otro.jornada="extendida"	-0,5	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.jornada="media"	-1	1
2	otro	otro.jornada="completa"	1	1
3	otro	otro.jornada="extendida"	1	1

- **double getComparacionLocacion(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de Particiones en Clases de Equivalencia (Esc. 1)

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	locacion="Home_Office" ="Presencial" ="indistinto"	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Locacion= Home_Office	1	1
2	otro	otro.Locacion= Presencial	-1	1
3	otro	otro.Locacion= Indistinto	1	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Locacion= Home_Office	-1	1
2	otro	otro.Locacion= Presencial	1	1
3	otro	otro.Locacion= Indistinto	-1	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Locacion= Home_Office	1	1
2	otro	otro.Locacion= Presencial	-1	1
3	otro	otro.Locacion= Indistinto	1	1

- **double getComparacionPuesto(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de Particiones en Clases de Equivalencia (Esc. 1)

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	otro.puesto="Junior" ="Senior" ="Management"	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Puesto=Junior	1	1
2	otro	otro.Puesto=Senior	-0,5	1
3	otro	otro.Puesto=Management	-1	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Puesto=Junior	-0,5	1
2	otro	otro.Puesto=Senior	1	1
3	otro	otro.Puesto=Management	-0,5	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	otro.Puesto=Junior	-1	1
2	otro	otro.Puesto=Senior	1	1
3	otro	otro.Puesto=Management	1	1

- **double getComparacionRemuneracion(Ticket otro)**

Para todas las comparaciones de tickets vemos siempre que el actual es empleador y el otro es empleado.

Numero escenario	Descripción
5	El empleador tiene asociado el Ticket1
6	El empleador tiene asociado el Ticket2
7	El empleador tiene asociado el Ticket3

Tabla de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	Remuneracion $\in R$	si	cumple contrato	1

Batería de pruebas (Esc. 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	$0 < \text{otro.Remuneracion} < V_1$	1	1
2	otro	$V_1 < \text{otro.Remuneracion} < V_2$	-0,5	1
3	otro	$V_2 < \text{otro.Remuneracion}$	-1	1

Batería de pruebas (Esc. 2)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	$0 < \text{otro.Remuneracion} < V_1$	1	1
2	otro	$V_1 < \text{otro.Remuneracion} < V_2$	1	1
3	otro	$V_2 < \text{otro.Remuneracion}$	-0,5	1

Batería de pruebas (Esc. 3)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	0<otro.Remuneracion<V1	1	1
2	otro	V1<otro.Remuneracion<V2	1	1
3	otro	V2<otro.Remuneracion	1	1

- **double getComparacionTotal(Ticket otro)**

Numero escenario	Descripción
6	Ticket actual: Ticket del escenario 2

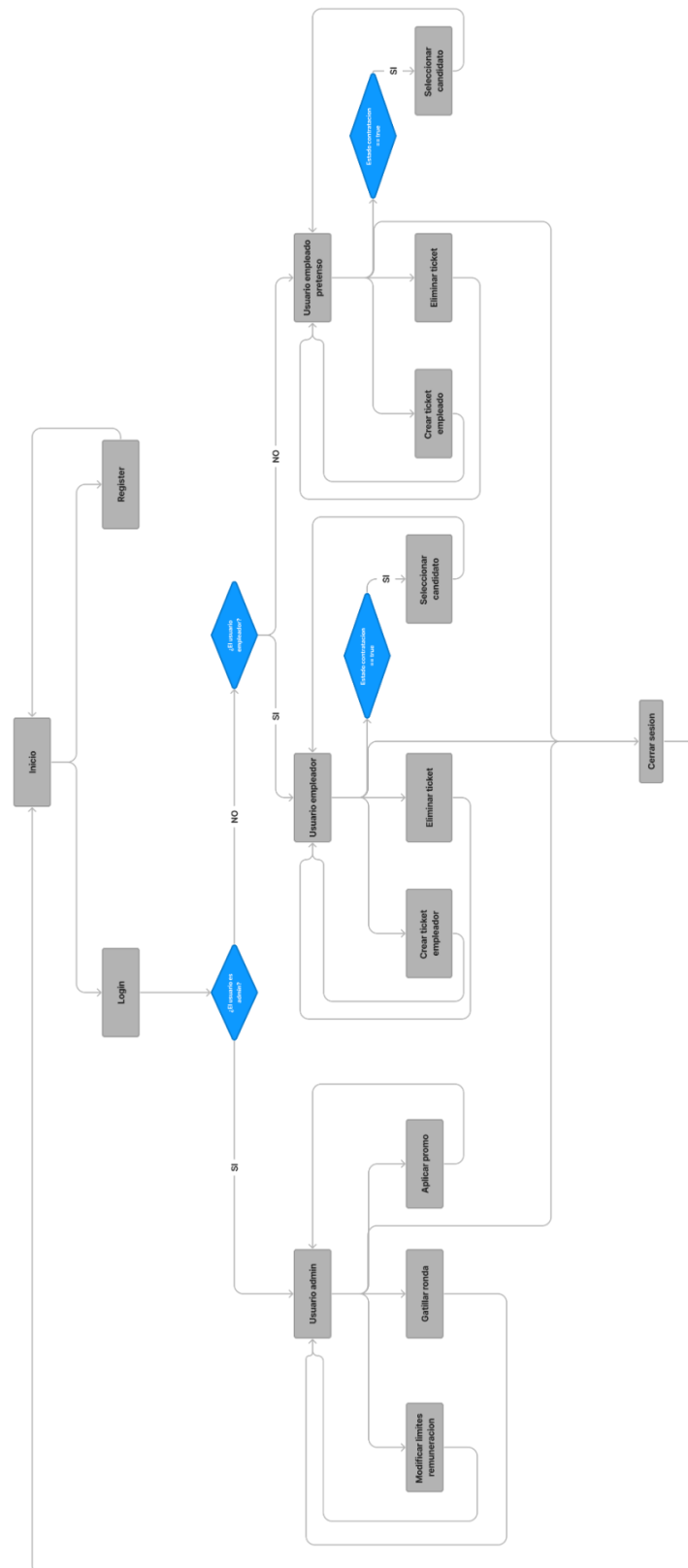
Tablas de Particiones en Clases de Equivalencia

Dato de entrada	Clases de equivalencia	Aplica?	Motivo	Identificador de clase de equivalencia
otro	Ticket	si	cumple contrato	1

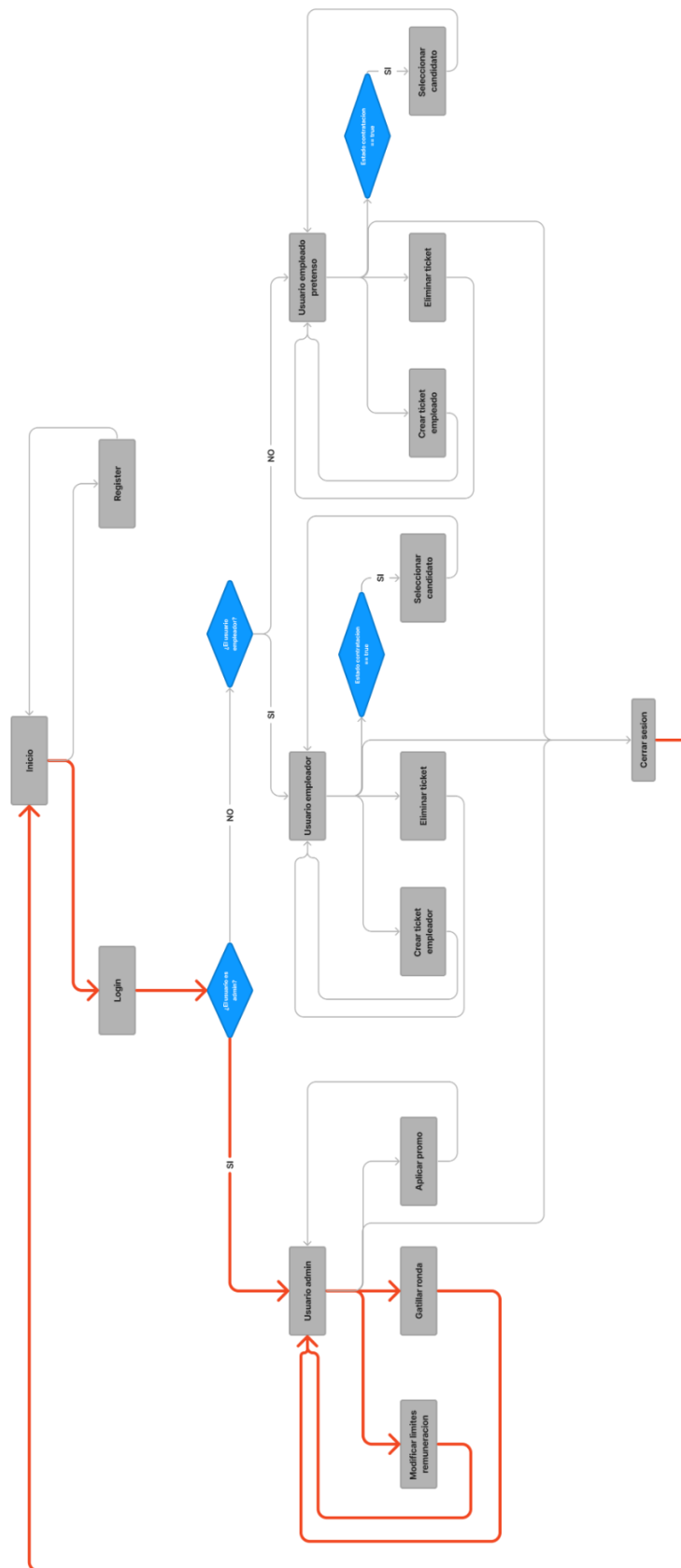
Batería de pruebas (Esc 1)

número de prueba	Datos de entrada	Valor	Salida Esperada	clases que abarca
1	otro	{“ticket.estudios=Primario”, “ticket.experiencia=Exp_nada”, “ticket.Jornada=Jornada_media”, “ticket.Locacion=Home_office”, “ticket.Puesto=Junior”, 0<ticket.Remuneracion<V1}	-2	1

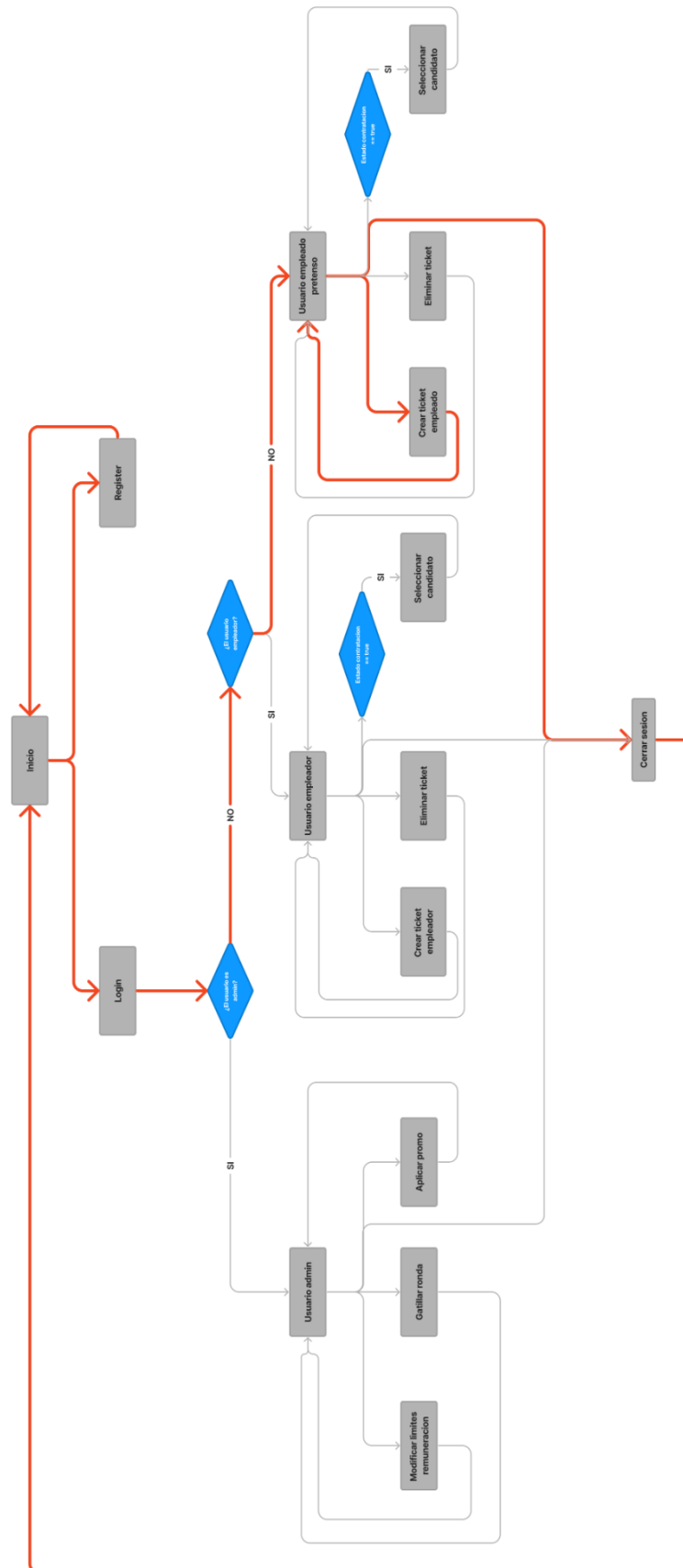
Anexo Test Integración



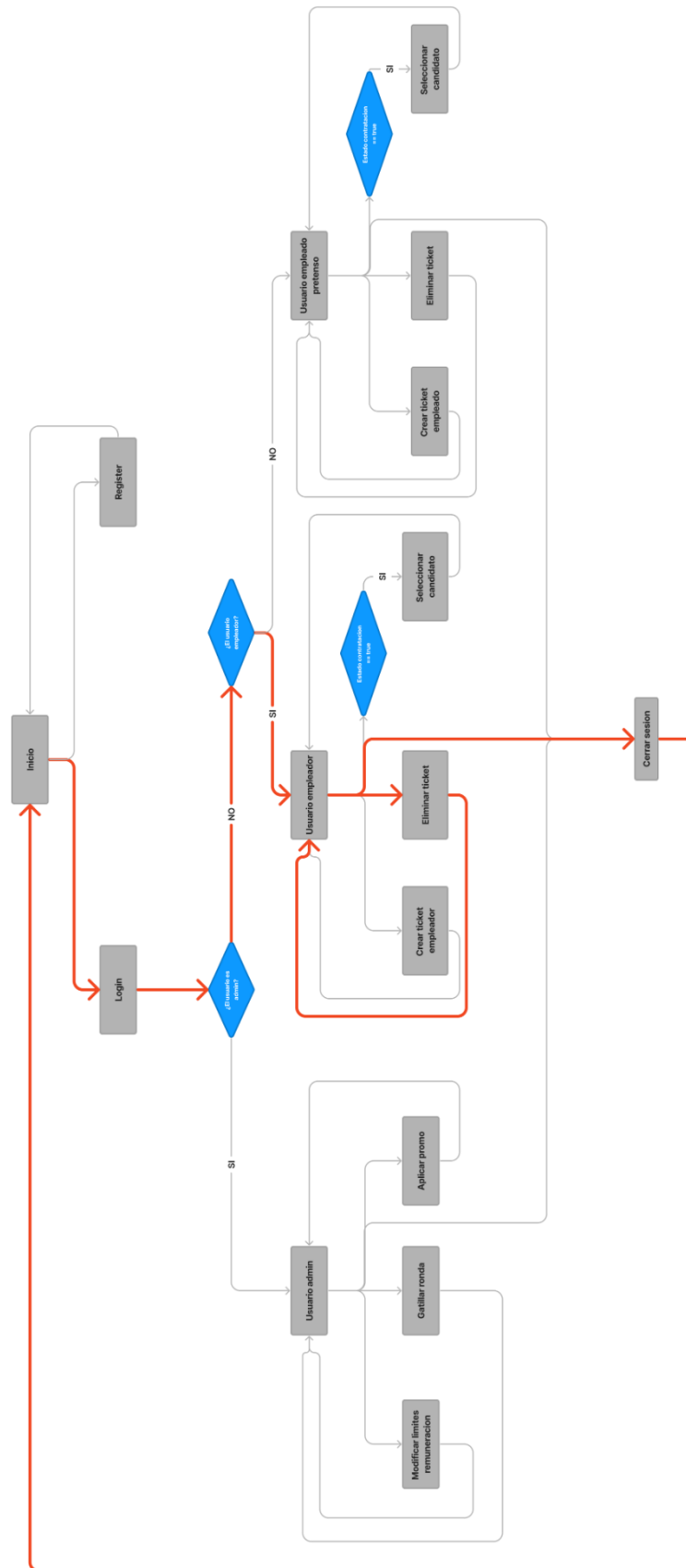
Caso 1



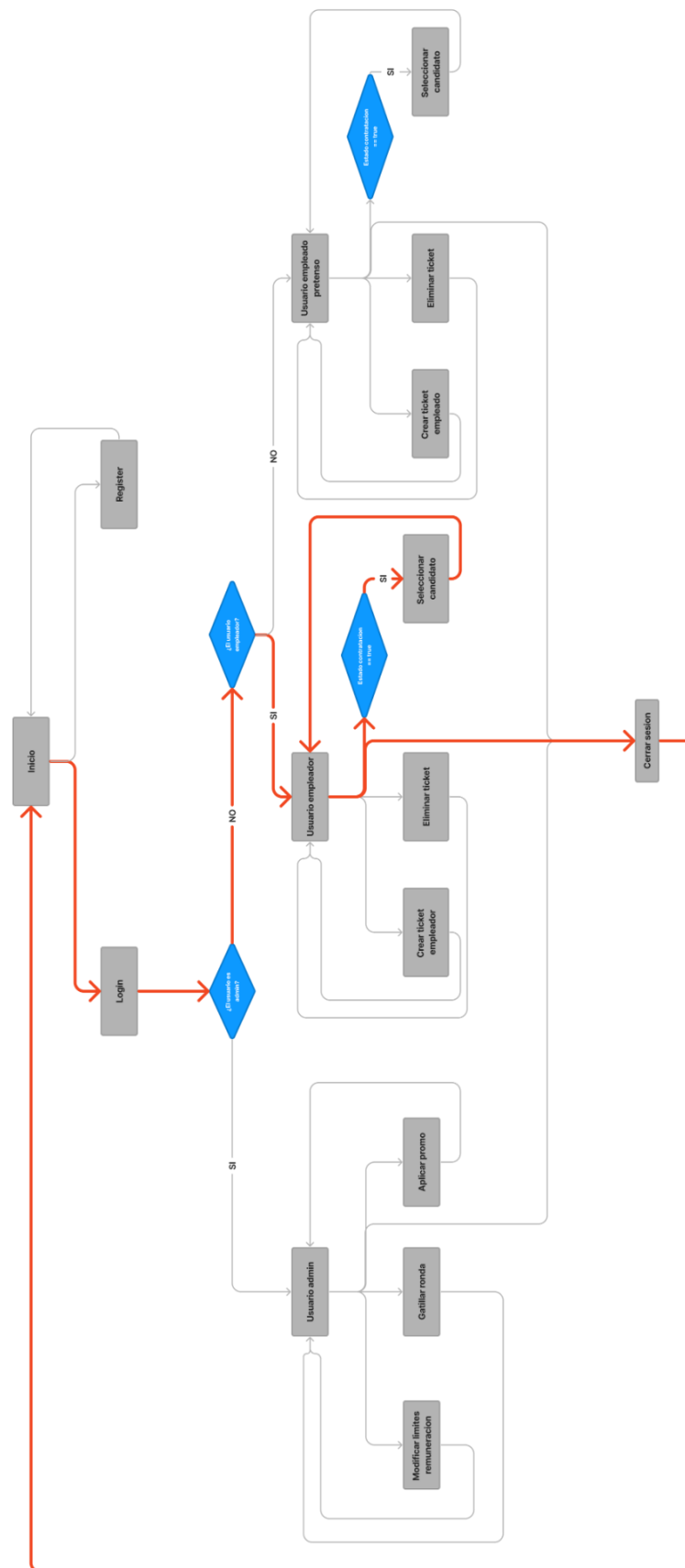
Caso 2



Caso 3



Caso 4



Caso 5

