## Plan de pruebas de la aplicación FranquiciasUC

Los niveles de prueba que se van a aplicar son los siguientes:

* Pruebas de aceptación. Las pruebas de aceptación se definirán siguiendo una estrategia basada en casos de uso y se ejecutarán de forma manual.
* Pruebas de integración. La estrategia para la def­­inición del orden de las pruebas de integración será jerárquica. Se probará:
  + La integración entre la capa de negocio y la de persistencia. En este caso, para la definición de los casos de prueba se utilizará técnica de prueba de métodos y caja negra y se utilizará JUnit.
  + La integración entre las tres capas. En este caso, para la definición de los casos de prueba se utilizarán técnica de casos de uso y se utilizarán JUnit y FEST.
* Pruebas unitarias. Se utilizará técnica de prueba de métodos y de caja negra (partición equivalente y AVL) para la definición de los casos de prueba de cada método de cada clase o componente. Será necesaria la utilización de JUnit, Mockito y FEST.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente.

**PRUEBAS DE ACEPTACIÓN**

En base a los casos de uso se identifican los siguientes escenarios (suponemos que la interfaz por construcción no permite dejar ningún campo requerido en blanco, para evitar casos de prueba referidos a información no disponible):

A1. CU: Alta Nueva Tienda

a. Alta válida

b. Alta no válida (tienda repetida)

c. Error en acceso a base de datos

A2. CU: Baja Tienda

a. Baja válida

b. Baja no válida (tienda existente con empleados)

c. Baja no válida (tienda no existente)

d. Error en acceso a base de datos

A3. CU: Consulta Tiendas

1. Consulta válida
2. Error en acceso a base de datos.

A4. CU: Consulta Tienda

a. Consulta válida (tienda existente)

b. Consulta no válida (tienda no existente)

c. Error en acceso a base de datos

A5. CU: Nuevo Empleado

a. Alta válida

b. Alta no válida (tienda no existente)

c. Alta no válida (empleado ya existente)

d. Error en acceso a base de datos

A7. CU: Mover Empleado

a. Movimiento válido

b. Movimiento no válido (empleado no existente)

c. Movimiento no válido (tienda origen no existente)

d. Movimiento no válido (tienda destino no existente)

e. Movimiento no válido (empleado no existe en tienda origen)

f. Error en acceso a base de datos

A8. CU: Registrar Alta Médica

a. Registro válido

b. Registro no válido (empleado no existe)

c. Registro no válido (empleado ya de alta)

d. Error en acceso a base de datos

A9. CU: Registrar Baja Médica

a. Registro válido

b. Registro no válido (empleado no existe)

c. Registro no válido (empleado ya de baja)

d. Error en acceso a base de datos

Los casos de prueba definidos para cada uno de estos escenarios, suponiendo que el estado inicial del sistema es el indicado en el Anexo 1, son los que se muestran en la Tabla 1. Los casos de prueba deberían ser ejecutados en el orden indicado, para que el estado final del sistema sea igual al inicial.

Tabla 1. Casos de prueba de aceptación

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Resultado** |
| A1.a | Tienda(TiendaD, DireccionD) | Mensaje éxito |
| A1.b | Tienda(TiendaA, DireccionX) | Mensaje error tienda repetida |
| A1.c | Tienda(TiendaF, DireccionF) | Mensaje error acceso BBDD |
| A2.a | TiendaD | Mensaje éxito |
| A2.b | TiendaA | Mensaje error tienda con empleados |
| A2.c | TiendaN | Mensaje error tienda no existe |
| A2.d | TiendaB | Mensaje error acceso BBDD |
| A3.a |  | Listado de tiendas (nombre) |
| A3.b |  | Mensaje error acceso BBDD |
| A4.a | TiendaA | Listado de empleados (nombre), total mensual sueldos y dirección |
| A4.b | TiendaN | “Tienda no existe” en campo dirección (resto de campos vacíos) |
| A4.c | TiendaA | Mensaje error acceso BBDD |
| A5.a | Tienda B, Empleado(22222222A, Pepe, ENCARGADO) | Mensaje éxito |
| A5.b | TiendaN, Empleado(22222222B, Juan, ENCARGADO) | Mensaje error tienda no existe |
| A5.c | Tienda B, Empleado(22222222A, Pepe, ENCARGADO) | Mensaje error empleado existe |
| A5.d | Tienda B, Empleado(22222222B, Pepe, ENCARGADO) | Mensaje error acceso BBDD |
| A7.a | TiendaA, 1111111A, TiendaB  TiendaB, 1111111A, TiendaA  (para dejarlo en el estado inicial) | Mensaje éxito |
| A7.b | TiendaA, 33333333A, TiendaB | Mensaje error empleado no existe |
| A7.c | TiendaN, 11111111A, TiendaB | Mensaje error tienda no existe |
| A7.d | TiendaA, 11111111A, TiendaN | Mensaje error tienda no existe |
| A7.e | TiendaA, 11111111D, TiendaB | Mensaje error empleado no en tienda |
| A7.f | TiendaA, 1111111A, TiendaB | Mensaje error acceso BBDD |
| A8.a | 11111111A | Mensaje éxito |
| A8.b | 33333333A | Mensaje error empleado no existe |
| A8.c | 11111111B | Mensaje error empleado ya de baja |
| A8.d | 11111111B | Mensaje error acceso BBDD |
| A9.a | 11111111B | Mensaje éxito |
| A9.b | 33333333A | Mensaje error empleado no existe |
| A9.c | 11111111A | Mensaje error empleado ya de alta |
| A9.d | 11111111A | Mensaje error acceso BBDD |

**PRUEBAS UNITARIAS**

**Pruebas unitarias de las clases de dominio**

Deberían probarse los métodos de las clases Empleado y Tienda. A modo de ejemplo, la Tabla 2 muestra los casos de prueba válidos para el método sueldoBruto de la clase Empleado. La definición completa de estas pruebas y su ejecución se realizó en la práctica 3 de Ingeniería del Software II en el curso anterior.

Tabla 2. Casos de prueba válidos para el método sueldoBruto de la clase Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UD.1a | (ENCARGADO, hoy, True) | 1500 |
| UD.1b | (VENDEDOR, hoy-2años, False) | 1500 |
| UD.1c | (AUXILIAR, hoy-5años, False) | 1000 |
| UD.1d | (ENCARGADO, hoy-5años-1día, False) | 2050 |
| UD.1e | (VENDEDOR, hoy-7años,True) | 1162.5 |
| UD.1f | (AUXILIAR, hoy-10años, False) | 1050 |
| UD.1g | (ENCARGADO, hoy-10años-1día, False) | 2100 |
| UD.1h | (VENDEDOR, hoy-15años, False) | 1600 |
| UD.1i | (AUXILIAR, hoy-20años, True) | 825 |
| UD.1j | (ENCARGADO, hoy-20años-1día, False) | 2200 |
| UD.1l | (VENDEDOR, hoy-50años, False) | 1700 |

**Pruebas unitarias de la capa de persistencia**

Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los que se exponen a continuación. Por similitud con las pruebas de aceptación e integración, los casos de prueba definidos suponen un estado de la base de datos como la indicada en el Anexo 1.

Casos de prueba para EmpleadosDAO (interfaz IEmpleadosDAO)

* Método crearEmpleado (Empleado e): Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UP.1a | Empleado(22222222A, Pepe, ENCARGADO) | Empleado(22222222A, Pepe, hoy, ENCARGADO, True) |
| UP.1b | Empleado(22222222A, Juan, ENCARGADO) | Null |
| UP.1d | Empleado(22222222B, Pepe, ENCARGADO)  (Error en BBDD) | DataAccessException |

* Método eliminarEmpleado(dni:String, nombre:String):Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UP.2a | 22222222A | Empleado(22222222A, Pepe, hoy, ENCARGADO, true) |
| UP.2b | 33333333A | Null |
| UP.2c | 22222222A(Error en BBDD) | DataAccessException |

* Método empleado(dni:String):Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UP.3a | 11111111A | Empleado(11111111A, Juan, 2002-01-15, ENCARGADO, False) |
| UP.3b | 33333333A | Null |
| UP.3c | 22222222A(Error en BBDD) | DataAccessException |

* Método empleados(): List<Empleado>

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UP.4a |  | Listado con 4 empleados |
| UP.4b | (Error en BBDD) | DataAccessException |

* Método modificarEmpleado(e:Empleado):Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UP.5a | Empleado(11111111A, Juan, 2002-01-15, Encargado, True) | Empleado(11111111A, Juan, 2002-01-15, Encargado, True) |
| UP.5b | Empleado(33333333A, Juan, 2002-01-15, Encargado, True) | Null |
| UP.5c | (Error en BBDD) | DataAccessException |

Casos de prueba para TiendasDAO (interfaz ITiendasDAO)

Deberían incluirse similares casos de prueba para este componente (no incluidos por simplicidad).

**Pruebas unitarias de la capa de negocio**

Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mock para las interfaces ITiendasDAO e IEmpleadosDAO. Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los que se exponen a continuación.

Prueba de GestionEmpleados (interfaces IGestionEmpleados e IGestionAltasBajas)

* Método nuevoEmpleado(e:Empleado, nombre:String):Empleado

Conceptualmente se trata de los mismos casos de aceptación del caso de uso 5.

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UB.1a | Empleado(22222222A, Pepe, ENCARGADO), TiendaB | Empleado(22222222A, Pepe, ENCARGADO) |
| UB.1b | Empleado(22222222B, Juan, ENCARGADO), TiendaN | Null |

* Método empleado(dni:String):Empleado

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UB.3a | 11111111ª | Empleado(11111111A, Juan, 2002-01-15, ENCARGADO, False) |
| UB.3b | 33333333ª | Null |
| UB.3c | 22222222A(Error en BBDD) | DataAccessException |

* Método trasladarEmpleado(dni:String, actual:String, destino:String): boolean

Conceptualmente se trata de los mismos casos de aceptación del caso de uso 7.

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UB.4a | 1111111A, TiendaA, TiendaB | True |
| UB.4b | 33333333A, TiendaA, TiendaB | False |
| UB.4c | 11111111A, TiendaN, TiendaB | False |
| UB.4d | 11111111A, TiendaA, TiendaN | False |
| UB.4e | 11111111D, TiendaA, TiendaB | OperacionNoValidaException |
| UB.4f | 1111111A, TiendaA, TiendaB  (Error en BBDD) | DataAccessException |

* Método bajaMedica(dni:String):boolean

Conceptualmente se trata de los mismos casos de aceptación del caso de uso 8.

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UB.6A | 11111111ª | True |
| UB.6b | 33333333ª | False |
| UB.6c | 11111111B | False |
| UB.6d | 11111111B  (Error en BBDD) | DataAccessException |

* Método altaMedica (dni:String):boolean

Conceptualmente se trata de los mismos casos identificados para el caso de uso 9.

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UB.7a | 11111111B | True |
| UB.7b | 33333333ª | False |
| UB.7c | 11111111ª | False |
| UB.7d | 11111111A  (Error en BBDD) | DataAccessException |

Prueba de GestionTiendas (interfa IGestionTiendas)

Habría que definir las pruebas para los métodos de este componente del mismo modo (no incluidos por simplicidad).

**Pruebas unitarias de la capa de presentación**

Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mocks para las interfaces IGestionTiendas, IGestionEmpleados, IGestionAltasBajas, ITiendasDAO e IEmpleadosDAO.

En este caso se aplica la técnica basada en casos de uso para la definición de las pruebas a realizar. Los casos de prueba definidos serán los mismos que los de las pruebas de aceptación (renombrados como UP.Xy) pero automatizados a través de JUnit y FEST.

**PRUEBAS DE INTEGRACIÓN**

El orden de las pruebas y los casos de prueba a realizar serían los siguientes:

1. Capa de negocio con capa de persistencia:
   1. GestionEmpleados con TiendasDAO y EmpleadosDAO.
   2. GestionTiendas con TiendasDAO.

Se usarían los mismos casos de prueba definidos como UB.x en la sección de pruebas unitarias de la capa de negocio, aquí renombrados como IB.x.

1. Capa de presentación con capa negocio y capa persistencia:
   1. VistaGerente con GestionEmpleados, GestionTiendas, TiendasDAO y EmpleadosDAO.
   2. VistaRRHH con GestionEmpleados, GestionTiendas, TiendasDAO y EmpleadosDAO.

Los casos de prueba coincidirían con las pruebas de aceptación, aunque en este caso se automatizarían utilizando la librería FEST y se renombraría como IPX.y.

ANEXO 1. IMPLEMENTACION DE LA CAPA DE PERSISTENCIA

La capa de persistencia, capa DAO, está implementada usando H2, una base de datos implementada en Java que puede configurarse para ser usada como base de datos en memoria, siendo ese el modo en que se ha utilizado en esta aplicación. Esto implica que los datos solo existen durante el tiempo de vida de la aplicación.

El estado inicial de la base de datos cuando se lanza la aplicación, que es necesario conocer para la realización de la fase de verificación y validación, es el que se muestra en la Tabla 1.

Tabla 1. Estado inicial de la base de datos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tiendas | | | | | | | |
| ID | Nombre | Dirección | Empleados | | | | |
| DNI | Nombre | Baja | Fecha | Categoría |
| 1 | Tienda A | Dirección A | 11111111A | Juan Pérez | False | 2002-01-15 | ENCARGADO |
| 11111111B | María Rodríguez | True | 2016-05-20 | VENDEDOR |
| 11111111C | Luis Martínez | False | 2022-05-21 | AUXILIAR |
| 2 | Tienda B | Dirección B | 11111111D | Lucía Ibáñez | False | 2010-06-01 | ENCARGADO |
| 3 | Tienda C | Dirección C |  |  |  |  |  |

*Patricia López Martínez*