

Nuestro objetivo es determinar los elementos necesarios para generar nuestro caso de prueba, que en todo nuestro proceso de testing serán de *Caja Negra*, que consiste en someter al sistema a una serie de datos de entrada y observar las salidas que genera.

En primer lugar, hablaremos del Caso de Uso 1 que pertenece a este módulo: Autenticación.

Es muy importante determinar el dominio de las entradas y las salidas. Vamos a usar la técnica de *Clases o particiones de equivalencia*, que se basa en dividir el dominio de entrada de cada parámetro en conjuntos distintos:

PARÁMETROS	CLASES DE EQUIVALENCIA
Usuario	Cadena 9 caracteres (No hay clases)
Contraseña	Cadena (No hay clases)

A continuación, dedicamos una parte a la *Conjetura de errores*, lo que consiste en proponer valores para los que se intuye que el sistema puede mostrar un comportamiento erróneo.

En el caso de Autenticación, las conjeturas serían que el String Contraseña fuera demasiado largos y el String Usuario no sea de 9 caracteres, de los cuales, o que alguno de ellos sea null.

En general, los casos de prueba constan de tres partes fundamentales:

- 1- Situación inicial: especificación de qué se requiere para la ejecución de la prueba.
- 2- Operaciones: ejecutar todas las posibilidades que se han implementado.
- 3- Oráculo: se encarga de comparar el resultado obtenido tras la ejecución del caso de prueba con el resultado que se esperaba.

En Autenticación, inicialmente tenemos una base de datos con un conjunto de tuplas DNI-Nombre-Apellido1-Apellido2-Password.

No es necesario eliminar todos los registros de la base de datos de prueba, ya que el caso de prueba no está orientado a la Base de Datos.

Para la implementación de las operaciones generamos todas las posibles situaciones que pueden generarse en el código.

Concretamente son: introducir correctamente tanto usuario como contraseña, en cuyo caso el programa mostrará la VentanaPrincipal con todas sus funcionalidades; en el caso en que alguno o ambos parámetros sean incorrectos se le notificará al usuario mediante un mensaje y se limpiarán los diversos campos; si algún campo se deja vacío también se notificará.

Además a este módulo pertenece el cdu2: Añadir Producto.

PARÁMETROS	CLASES DE EQUIVALENCIA
Tipo de producto	[Perecedero, No Perecedero]
Categoría	[Alimentación, Cosméticos, Limpieza]
Producto	[]
Unidades	[0,baseDedatos.getProductos.size()]

Conjetura de errores: Que el usuario no rellene alguno de los campos.

En AñadirProducto, inicialmente tenemos una base de datos con un conjunto de tuplas TipoDeProducto-Categoría-Producto-Unidades.

No es necesario eliminar todos los registros de la base de datos de prueba, ya que el caso de prueba no está orientado a la Base de Datos.

En cuanto a las operaciones diferenciamos las siguientes: introducir correctamente todos los campos (Tipo de producto, categoría, producto y unidades), en cuyo caso el programa añadirá un producto a la base de datos; en el caso en que alguno o ambos parámetros estén vacíos se le notificará al usuario mediante un mensaje.