

Documento de Pruebas unitarias.

Pruebas para el manejador de sesiones.

1. Prueba de Carga de Usuarios:

Descripción: Verificar que la lista de usuarios se cargue correctamente desde el archivo.

Resultado Esperado: Se espera que la cantidad de usuarios cargados sea igual a 10.

Resultado Actual: La carga de usuarios se realizó con éxito, y la cantidad de usuarios es efectivamente 10.

2. Prueba de Revisión de Usuario Existente con Contraseña Correcta:

Descripción: Verificar que un usuario existente con contraseña correcta devuelva el rol correspondiente.

Resultado Esperado: Se espera que el usuario "usuario2" con la contraseña "contrasena2" devuelva el rol "adminLocal".

Resultado Actual: El usuario "usuario2" se autenticó correctamente, devolviendo el rol esperado.

3. Prueba de Revisión de Usuario Existente con Contraseña Incorrecta:

Descripción: Verificar que un usuario existente con contraseña incorrecta devuelva un mensaje de error.

Resultado Esperado: Se espera que el usuario "usuario4" con contraseña incorrecta devuelva el mensaje "Mi bro te equivocaste de contrasena".

Resultado Actual: El usuario "usuario4" generó el mensaje de error esperado al ingresar una contraseña incorrecta.

4. Prueba de Revisión de Usuario No Existente:

Descripción: Verificar que un usuario no existente devuelva un mensaje de error.

Resultado Esperado: Se espera que el usuario "usuarioxd" con cualquier contraseña devuelva el mensaje "Usuario no existe".

Resultado Actual: El sistema generó el mensaje esperado al intentar autenticar un usuario inexistente.

5. Prueba Adicional de Revisión de Usuario Existente con Contraseña Correcta:

Descripción: Verificar nuevamente que un usuario existente con contraseña correcta devuelva el rol correspondiente.

Resultado Esperado: Se espera que el usuario "usuario1" con la contraseña "contrasena1" devuelva el rol "cliente".

Resultado Actual: El usuario "usuario1" se autenticó con éxito, devolviendo el rol esperado.

Pruebas unitarias para la clase Inventario.

Por supuesto, aquí te proporciono un documento de pruebas de errores para el código proporcionado:

1. Prueba de Agregar Vehículo al Inventario:

Descripción: Verificar que se pueda agregar un vehículo al inventario correctamente.

Resultado Esperado: Se espera que el método agregarVehiculo retorne true después de agregar un vehículo al inventario.

Resultado Actual: La prueba pasa. El método agregarVehiculo retorna true correctamente al agregar un vehículo al inventario.

2. Prueba de Eliminar Vehículo del Inventario:

Descripción: Verificar que se pueda eliminar un vehículo del inventario correctamente.

Resultado Esperado: Se espera que el método eliminarVehiculo retorne true después de eliminar un vehículo existente del inventario y que la cantidad de vehículos sea 0.

Resultado Actual: La prueba pasa. El método eliminarVehiculo retorna true después de eliminar un vehículo del inventario, y la cantidad de vehículos después de eliminar es 0.

Prueba de Agregar Cliente al Inventario:

Descripción: Verificar que se pueda agregar un cliente al inventario correctamente.

Resultado Esperado: Se espera que el método agregarCliente agregue al cliente al inventario y retorne true.

Resultado Actual: La prueba pasa. El método agregarCliente agrega el cliente correctamente y retorna true.

Detalles de la prueba:

Se crea un inventario vacío y se verifica que la lista de clientes esté inicialmente vacía.

Se crean una licencia de conducción y una tarjeta de crédito para el cliente.

Se agrega un cliente al inventario con datos específicos.

Se verifica que la lista de clientes ahora contenga exactamente un cliente.

Se obtiene el cliente recién agregado.

Se verifican todos los atributos del cliente para asegurarse de que coincidan con los valores proporcionados.

Se verifica que el método agregarCliente retorne true, indicando una adición exitosa al inventario.

Prueba de Agregar Sede al Inventario:

Descripción: Verificar que se pueda agregar una sede al inventario correctamente.

Resultado Esperado: Se espera que el método agregarSede añada una sede al inventario, asignándole un administrador local y actualizando las listas correspondientes.

Resultado Actual: La prueba pasa. El método agregarSede agrega una sede al inventario, con la información correcta y el administrador local asignado.

Prueba de Eliminar Sede del Inventario:

Descripción: Verificar que se pueda eliminar una sede del inventario correctamente.

Resultado Esperado: Se espera que el método eliminarSede elimine una sede del inventario, eliminando también la información asociada y actualizando las listas correspondientes.

Resultado Actual: La prueba pasa. El método eliminarSede elimina la sede del inventario, así como la información asociada, y las listas se actualizan correctamente.

Pruebas para la clase empresa:

Nueva reserva: (unitaria)

Descripción:

Verificar que el método nuevaReserva de la clase Empresa pueda crear una nueva reserva correctamente, asignando un vehículo disponible al cliente y generando la reserva con los parámetros proporcionados.

Resultado Esperado:

Se espera que la reserva se cree con éxito y que los valores de la reserva coincidan con los parámetros proporcionados.

Resultado Actual:

La prueba pasa. El método nuevaReserva de la clase Empresa crea la reserva correctamente, asigna un vehículo disponible, y todos los valores de la reserva coinciden con los parámetros proporcionados.

Nueva reserva (integral):

Prueba de Modificación de Seguros en una Reserva:

Descripción:

Se verifica que el método modificarSegurosReserva de la clase Empresa modifique correctamente la lista de seguros de una reserva.

Resultado Esperado:

Después de modificar los seguros de una reserva mediante el método modificarSegurosReserva, se espera que la lista de seguros de la reserva refleje los cambios realizados.

Resultado Actual:

La prueba confirma que el método modificarSegurosReserva realiza las modificaciones adecuadas en la lista de seguros de la reserva.

Pruebas para la clase Tarifa:

Prueba Crear Tarifa Sin Descuento:

Descripción: Verificar que se pueda crear una tarifa sin descuento y que los valores se configuren correctamente.

Resultado Esperado: Se espera que los valores de días de renta, número de conductores extra, tarifa diaria normal, precio de seguros y precio total sean configurados correctamente.

Prueba Crear Tarifa Con Descuento:

Descripción: Verificar que se pueda crear una tarifa con descuento y que los valores se configuren correctamente.

Resultado Esperado: Se espera que los valores de días de renta, número de conductores extra, tarifa diaria normal, precio de seguros y precio total sean configurados correctamente, y que el precio total refleje el descuento.

Prueba Modificar Días de Renta:

Descripción: Verificar que se pueda modificar correctamente la cantidad de días de renta y que el precio total se ajuste en consecuencia.

Resultado Esperado: Se espera que el número de días de renta se actualice correctamente y que el precio total refleje este cambio.

Prueba Modificar Número de Conductores Extra:

Descripción: Verificar que se pueda modificar correctamente el número de conductores extra y que el precio total se ajuste en consecuencia.

Resultado Esperado: Se espera que el número de conductores extra se actualice correctamente y que el precio total refleje este cambio.

Pruebas clase seguro:

Prueba Crear Seguro:

Descripción: Verificar que se pueda crear un seguro correctamente y que los valores se configuren adecuadamente.

Resultado Esperado: Se espera que el nombre y la tarifa diaria del seguro se configuren correctamente durante la creación.

Resultado Actual: La prueba pasa. El seguro se crea satisfactoriamente con el nombre y la tarifa diaria configurados según lo especificado.

Prueba Modificar Nombre de Seguro:

Descripción: Verificar que se pueda modificar el nombre del seguro correctamente.

Resultado Esperado: Se espera que el nombre del seguro se actualice correctamente.

Resultado Actual: La prueba pasa. Después de modificar el nombre del seguro, se verifica que el cambio se refleje correctamente.

Prueba Modificar Tarifa Diaria de Seguro:

Descripción: Verificar que se pueda modificar la tarifa diaria del seguro correctamente.

Resultado Esperado: Se espera que la tarifa diaria del seguro se actualice correctamente.

Resultado Actual: La prueba pasa. Después de modificar la tarifa diaria del seguro, se verifica que el cambio se refleje correctamente.

Prueba Acceder a Nombre de Seguro:

Descripción: Verificar que se pueda acceder correctamente al nombre del seguro.

Resultado Esperado: Se espera que el nombre del seguro se pueda obtener correctamente.

Resultado Actual: La prueba pasa. Se accede al nombre del seguro y se verifica que se obtenga el valor esperado.

Prueba Acceder a Tarifa Diaria de Seguro:

Descripción: Verificar que se pueda acceder correctamente a la tarifa diaria del seguro.

Resultado Esperado: Se espera que la tarifa diaria del seguro se pueda obtener correctamente.

Resultado Actual: La prueba pasa. Se accede a la tarifa diaria del seguro y se verifica que se obtenga el valor esperado.

Pruebas de Carga.

Carga de Categorías

Se verificó que el sistema carga correctamente las categorías desde el archivo "archivoCategorias.txt". La lista de categorías no está vacía después de la carga.

Carga de Administradores Locales

Se confirmó que el sistema carga adecuadamente los administradores locales desde el archivo "archivoAdminLocal.txt". La lista de administradores locales no está vacía después de la carga.

Carga de Productos

Se constató que el sistema carga los productos correctamente desde el archivo "archivoProductos.txt". La lista de productos no está vacía después de la carga.

Carga de Usuarios

Se comprobó que el sistema realiza la carga de usuarios de manera adecuada desde el archivo "archivoUsuarios.txt". La lista de usuarios no está vacía después de la carga.

Carga de Ventas

Se verificó que el sistema efectúa la carga de ventas de forma exitosa desde el archivo "archivoVentas.txt". La lista de ventas no está vacía después de la carga.

Casos de Excepción

Archivo Inexistente

Se confirmó que el sistema maneja correctamente la excepción cuando se intenta cargar un archivo inexistente. El sistema arroja una excepción de tipo `IOException`.

Formato de Archivo Inválido

Se validó que el sistema gestiona adecuadamente la excepción cuando se intenta cargar un archivo con formato inválido. El sistema arroja una excepción de tipo `NumberFormatException`.