



**UNIVERSIDAD  
DE LA SERENA**  
CHILE

---

# PLAN DE PRUEBAS

## “InventarisPro” v0.1

---

**Equipo de desarrollo:**

Antony Rodriguez.  
Roberto Contreras.  
Sebastian Morgado.

**Profesor:**

Guillermo Leyton.

**Asignatura:**

Programación Avanzada

## Ficha del Documento

Nombre del proyecto	Curso
Inventaris Pro	Programación Avanzada

Versión	Equipo de desarrollo	Fecha
0.1	Sebastian Morgado Roberto Contreras Antony Rodriguez	27/11/2021

Testers	Nombre	Cargo
T01	Javier Garcia Victor Maldonado	SQA
T02	Sebastian Morgado Roberto Contreras Antony Rodriguez	Equipo de desarrollo

# Índice

<b>Ficha del Documento</b>	<b>2</b>
<b>Índice</b>	<b>3</b>
<b>Clases del sistema</b>	<b>5</b>
<b>Pruebas</b>	<b>6</b>

# Propósito

El objetivo de este documento es presentar las pruebas realizadas sobre el sistema (funciones,clases) que ahora se pueden aplicar, de esta manera podremos evaluar cómo se comportan los diferentes componentes del prototipo actual bajo condiciones inesperadas. El objetivo básico de estas pruebas es encontrar tantos errores como sea posible y minimizarlos antes de que se distribuyan a los clientes. Se especifica el tipo de prueba a aplicar, su prioridad, sus valores de entrada así como sus eventos esperados y finalmente se detalla cómo se puede determinar demostrar que se realiza el evento esperado.

# Clases del sistema

En el siguiente apartado se presentarán las clases del prototipo actual a los cuales se le harán pruebas.

- Clase **[C1]**: ProductoController
  - Módulo **[C1.M1]**: ProductoRepository
    - Función **[C1.M1.F1]**: Add
    - Función **[C1.M1.F2]**: Update
    - Función **[C1.M1.F3]**: GetById
    - Función **[C1.M1.F4]**: Delete
- Clase **[C2]**: VentaController
  - Módulo **[C2.M1]**: VentaRepository
    - Función **[C2.M1.F1]**: Add
    - Función **[C2.M1.F2]**: GetById
    - Función **[C2.M1.F3]**: Delete
- Clase **[C3]**: TrabajadorController
  - Módulo **[C3.M1]**: TrabajadorRepository
    - Función **[C3.M1.F1]**: GetByEmail
    - Función **[C3.M1.F2]**: Validar
    - Función **[C3.M1.F3]**: Registrar

# Pruebas

En este apartado se presentará cada uno de los componentes y las pruebas a realizar de cada uno, detallando lo siguiente:

- Nombre del método
- Descripción de la prueba
- Método de la prueba (Caja negra / Caja blanca)
- Prioridad de la prueba (Alta / Media / Baja)
- Datos de entrada
- Pasos que sigue la función.
- Salida esperada
- Entradas no válidas

<b>[C1.M1.F1]</b>	<b>Nombre:</b> Add()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	----------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si el objeto ingresado contiene atributos válidos para su creación.

Entradas: Objeto ProductoViewModel

Pasos:

1. Ingresar valores en el formulario, posteriormente enviarlos.
2. Se crea un Objeto ProductoViewModel
3. Se ejecuta la función Add()
4. Verifica si los valores ingresados son válidos.
5. Agrega el producto en la base de datos.

Salida esperada: Retorna un número mayor a cero si la ejecución fue exitosa, un cero si no se agregó el producto, y un número menor a cero si falló la ejecución de la consulta.

Entradas no válidas:

- Objeto ProductoViewModel nulo
  - Atributos requeridos nulo o vacíos:
    - IdCategoria (String)
    - IdNegocio (String)
    - Nombre (String)
    - stock (Integer)
    - preciocosto (Integer)
    - precioventa (Integer)
    - unidad de medida (String)
    - rut (String)

<b>[C1.M1.F2]</b>	<b>Nombre:</b> Update()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	-------------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si el objeto ingresado contiene atributos válidos para su edición.

Entradas:

- Identificador del producto a editar
- Objeto tipo ProductoViewModel

Pasos:

1. Ingresar valores en el formulario, posteriormente enviarlos.
2. Se ejecuta la función Update()
3. Se obtiene un Objeto ProductoViewModel de entrada
4. Se obtiene el identificador(String) del producto que se está editando.
5. Se ejecuta la función GetById que verifica que el producto exista.
6. Verifica si los valores(atributos del producto) ingresados son válidos.
7. Actualiza el producto en la base de datos.

Salida esperada: Retorna un número mayor a cero si la ejecución fue exitosa, un cero si no se agregó el producto, y un número menor a cero si falló la ejecución de la consulta.

Entradas no válidas:

- Atributos requeridos nulo o vacíos:
  - IdCategoria (String)
  - Nombre (String)
  - stock (Integer)
  - preciocosto (Integer)
  - precioventa (Integer)
  - unidad de medida (String)
  - rut (String)

<b>[C1.M1.F3]</b>	<b>Nombre:</b> GetById	<b>Prioridad:</b> Alta	<b>Método de prueba:</b> Caja Negra
-------------------	------------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si se obtiene el producto correspondiente al identificador

Entradas: Identificador del producto (String).

Pasos:

1. Se ejecuta GetById()
2. Se obtiene un identificador (String) de entrada
3. Se obtiene rut del usuario con la función GetCurrentTrabajador()
4. Se obtiene el producto perteneciente al rut del usuario.

Salida esperada: Retorna un objeto de tipo Producto.

Entradas no válidas:

- Identificador (String) nulo o vacío.

<b>[C1.M1.F4]</b>	<b>Nombre:</b> Delete	<b>Prioridad:</b> Alta	<b>Método de prueba:</b> Caja Negra
-------------------	-----------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si el id del producto a eliminar es válido.

Entradas: Objeto tipo Producto.

Pasos:

1. El usuario selecciona el producto a eliminar.
2. Se ejecuta la función Delete().
3. Se obtiene un objeto(Producto) de entrada
4. Se obtiene rut del usuario con la función GetCurrentTrabajador()
5. Se elimina el producto perteneciente al rut del usuario.

Salida esperada: Retorna un número mayor a cero si la ejecución fue exitosa, un cero si no eliminó el producto, y un número menor a cero si falló la ejecución de la consulta.

Entradas no válidas: Objeto Producto nulo o vacío.

<b>[C2.M1.F1]</b>	<b>Nombre:</b> Add()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	----------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si se obtienen atributos válidos para la creación de la venta.

Entradas: Importe total de la venta (Int)

Pasos:

1. Obtiene un valor entero.
2. Se crea un Objeto Venta.
3. Se ejecuta la función Add()
4. Verifica si los valores obtenidos son válidos.
5. Agrega la venta en la base de datos.

Salida esperada: Retorna una lista de tipo Venta (List<Venta>), que muestra en la página todas las acciones asociadas al usuario.

Entradas no válidas: Rut o Identificador de Negocio inexistente.



<b>[C2.M1.F2]</b>	<b>Nombre:</b> GetById	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	------------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si se obtiene la venta correspondiente al identificador.

Entradas: Identificador de la venta (String).

Pasos:

1. Se ejecuta GetById().
2. Se obtiene identificador.
3. Se obtiene el rut del trabajador con GetCurrentTrabajador().
4. Verifica si los valores son válidos.

Salida esperada: Retorna un objeto de tipo Venta.

Entradas no válidas: Identificador (String) nulo o vacío.

<b>[C2.M1.F3]</b>	<b>Nombre:</b> Delete()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	-------------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si se obtiene el id de la venta a eliminar.

Entradas: Objeto tipo Venta

Pasos:

1. Se obtiene el rut del trabajador con GetCurrentTrabajador().
2. Se obtiene el identificador de la venta.

Salida esperada: Retorna un número mayor a cero si la ejecución fue exitosa, un cero si no se eliminó, y un número menor a cero si falló la ejecución de la consulta.

Entradas no válidas: Objeto tipo Venta nulo o vacío.

<b>[C3.M1.F1]</b>	<b>Nombre:</b> Registrar()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	----------------------------	------------------------	--

Descripción: Esta prueba consiste en verificar si los datos ingresados (String) son válidos para el registro del usuario.

Entradas: Valor(String) introducido por el usuario.

Pasos:

1. Ingresar valores en el formulario.
2. Verificar si los valores ingresados son válidos.
3. Crear un negocio.
4. Encriptar la contraseña.
5. Crear el trabajador.

Salida esperada: Retorna vacío.

Entradas no válidas:

- Campos vacíos.
- Nombre de usuario de más de 25 caracteres.
- Apellido de usuario de más de 25 caracteres.
- Correo de usuario de más de 320 caracteres.
- Teléfono de usuario de más de 9 caracteres.
- Ingresar un rut superior al valor máximo de 10 caracteres.
- Ingresar un nombre de negocio superior al valor máximo del tipo String .

<b>[C3.M1.F2]</b>	<b>Nombre:</b> Validar()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	--------------------------	------------------------	--

Descripción: Este método sirve para validar los datos del usuario.

Entradas: El objeto TrabajadorLoginViewModel, el cuál contiene 2 miembros, los cuales son el Correo (String) y la Contraseña (String).

Pasos:

1. Ingresar valores en el formulario.
2. El servidor recibe los datos, y verifica si son válidos.
  - a. Si los datos son inválidos (ej: texto vacío), retorna la misma página (Inicio Sesión), con los datos anteriormente ingresados, y un mensaje de error.
- 3.

Salida esperada: Retorna un objeto TrabajadorCookie (el cual contiene el rut, correo, nombre y rol del trabajador).

Entradas no válidas:

- Campos vacíos.
- Correo inexistente.
- Contraseña incorrecta.

<b>[C3.M1.F3]</b>	<b>Nombre:</b> GetByEmail()	<b>Prioridad:</b> Alto	<b>Método de prueba:</b> Caja Negra
-------------------	-----------------------------	------------------------	--

Descripción: Este método sirve para obtener los datos del trabajador con su email.

Entradas: Un string que contenga el email.

Pasos:

1. Se le pasa un string que tiene el email. Las funciones que lo llaman antes verifican si el email es válido.

Salida esperada: Retorna los datos del trabajador.

Entradas no válidas:

- String nulo.
- String vacío.