

**[stock-manager]**  
**Modelo de Casos de Prueba**  
**Versión [1.0]**

**Contenido**

**1. CASOS Y PROCEDIMIENTOS DE PRUEBAS UNITARIAS .....2**

1.1. COMPONENTE: REGISTRO DE COMPRAS DE ACCIONES – VERSIÓN 1.0 .....2

1.1.1. *Requerimientos Funcionales.....2*

1.1.2. *Requerimientos No funcionales .....2*

1.2. COMPONENTE 2: VISUALIZACIÓN DEL HISTORIAL DE COMPRAS – VERSIÓN 1.0 .....2

1.2.1. *Requerimientos Funcionales.....3*

1.2.2. *Requerimientos No funcionales .....3*

## 1. Casos y procedimientos de Pruebas Unitarias

En esta sección se detallan los casos y procedimientos de pruebas unitarias realizados para la aplicación web de gestión de acciones de empresas. Las pruebas aseguran el cumplimiento de los requerimientos funcionales y no funcionales, basados en las historias de usuario y criterios de aceptación.

### 1.1. Componente: Registro de Compras de Acciones – Versión 1.0

#### 1.1.1. Requerimientos Funcionales

- Registro de compras de acciones con validación de datos.
- Persistencia de datos en la base de datos.
- Gestión de errores con mensajes claros.

##### 1.1.1.1. Funcionalidad 1: Registro de compras

Entrada	Salida esperada
Fecha válida (aaaa-mm-dd), símbolo de empresa válido, cantidad positiva, valor positivo.	Mensaje de éxito y registro guardado en la base de datos.
Fecha inválida (2024/11/30), símbolo inválido, cantidad negativa, valor negativo.	Mensaje de error indicando el campo inválido.

#### Procedimiento de prueba

- 1) Enviar una solicitud POST a la ruta / con datos válidos.
- 2) Verificar que la respuesta contiene el mensaje Registro exitoso.
- 3) Enviar una solicitud POST con:
  - Fecha en formato inválido.
  - Símbolo de empresa no existente.
  - Cantidad de acciones negativa.
  - Valor de compra negativo.
- 4) Verificar que cada solicitud retorna un mensaje de error apropiado.

##### 1.1.1.2. Funcionalidad 2: Prevención de duplicados

Entrada	Salida esperada
Registro con fecha, empresa, cantidad y valor idénticos a un registro existente.	Mensaje de error indicando duplicación.

#### Procedimiento de prueba

- 1) Enviar dos solicitudes POST consecutivas con los mismos datos.
- 2) Verificar que la primera solicitud guarda el registro y la segunda retorna un error.

#### 1.1.2. Requerimientos No funcionales

Requerimientos a verificar:

- Manejo adecuado de errores y mensajes claros.
- Persistencia de datos en la base de datos.

### 1.2. Componente 2: Visualización del Historial de Compras – Versión 1.0

### 1.2.1. Requerimientos Funcionales

- Visualización del historial completo de compras.
- Filtrado del historial por rango de fechas.
- Cálculo de valores asociados: ganancia/pérdida, porcentajes y valores actuales.

#### 1.2.1.1. Funcionalidad 1: Historial Completo

Entrada	Salida esperada
Solicitud GET sin parámetros.	Lista completa de compras almacenadas.

##### Procedimiento de prueba

- 1) Registrar varias compras con diferentes fechas y valores.
- 2) Realizar una solicitud GET a la ruta /historial.
- 3) Verificar que la respuesta incluye todos los registros almacenados.

#### 1.2.1.2. Funcionalidad 2: Filtrado por fecha

Entrada	Salida esperada
Rango de fechas válido.	Lista filtrada por rango.
Rango sin registros.	Mensaje indicando que no hay datos.

##### Procedimiento de prueba

- 1) Registrar varias compras con diferentes fechas.
- 2) Realizar una solicitud GET a /historial con un rango de fechas.
- 3) Verificar que la respuesta incluye solo los registros dentro del rango.

#### 1.2.1.3. Funcionalidad 3: Cálculo de Valores

Entrada	Salida esperada
Registro con cantidad y precio actual.	Cálculo correcto de valores totales, ganancia/pérdida y porcentajes.

##### Procedimiento de prueba

- 1) Registrar compras con diferentes cantidades y precios.
- 2) Realizar una solicitud GET a /historial.
- 3) Verificar que los cálculos en los resultados son correctos.

### 1.2.2. Requerimientos No funcionales

Requerimientos a verificar:

- Respuesta rápida para solicitudes de historial.
- Cálculos precisos y consistentes.

#### 1.2.2.1. Requerimiento no Funcional 1: Respuesta rápida

Entorno:

Base de datos con más de 100 registros.

Entrada	Salida esperada
Solicitud GET con rango amplio.	Respuesta en menos de 1 segundo.

### **Procedimiento de prueba**

- 1) Insertar más de 100 registros en la base de datos.
- 2) Realizar solicitudes GET y medir el tiempo de respuesta.

#### *1.2.2.2. Requerimiento no funcional 2: Consistencia en cálculos*

*Entorno:*

Base de datos con registros variados.

<b>Entrada</b>	<b>Salida esperada</b>
Registros con diferentes precios y cantidades.	Resultados consistentes en todos los cálculos.

### **Procedimiento de prueba**

- 1) Registrar compras con combinaciones variadas de precios y cantidades.
- 2) Realizar solicitudes GET y validar los resultados con cálculos manuales.

### **Conclusión**

Se completaron con éxito las pruebas unitarias planificadas, cumpliendo con los requisitos funcionales y no funcionales establecidos en las historias de usuario. La aplicación es robusta, maneja errores adecuadamente y asegura la persistencia y precisión de los datos.