

Plan de Pruebas

StockManager

Collareda Agustín y Frey Hugo

Índice

Introducción.....	3
Alcance del Plan de Pruebas.....	3
Estrategia de Pruebas.....	3
Casos de Prueba.....	4
CU02: CrearMueble.....	4
CP02_1 - Crear mueble válido.....	4
CP02_2 - Crear mueble faltan campos.....	4
CP02_3 - Crear mueble con strings.....	4
CU03: ModificarMueble.....	4
CP03_1 - Actualizar mueble válido.....	4
CP03_2 - Actualizar mueble inválido.....	4
CU04: EliminarMueble.....	5
CP04_1 - Eliminar mueble válido.....	5
CP04_2 - Eliminar mueble inválido.....	5
Pruebas adicionales.....	5
CP00_1 - Obtener mueble válido.....	5
CP00_2 - Obtener mueble no encontrado.....	5
CP00_3 - Obtener mueble ID inválido.....	5
Seguimiento y Control.....	6

Plan de Pruebas

Introducción

El presente documento tiene como objetivo definir el Plan de Pruebas para el sistema StockManager, en el marco del proceso de verificación y validación del producto. Este plan establece el enfoque general que se seguirá para garantizar que el software cumpla con los requisitos funcionales y no funcionales definidos, y que se comporte de manera confiable bajo condiciones normales y excepcionales.

El propósito de este plan es proporcionar una visión clara y estructurada de las actividades de prueba a realizar, los recursos necesarios, los criterios de entrada y salida, los tipos de pruebas aplicables.

Alcance del Plan de Pruebas

Este plan tiene como alcance todos los Casos de Uso elaborados del sistema StockManager.

Se excluyen de este plan módulos que aún se encuentren en fase de diseño o bajo desarrollo, así como funcionalidades que dependen de sistemas externos no disponibles durante la ejecución de las pruebas.

Estrategia de Pruebas

Se llevarán a cabo pruebas unitarias, orientadas a verificar el correcto funcionamiento de los métodos o funciones de forma aislada. Estas pruebas serán implementadas por los desarrolladores utilizando un marco de pruebas automatizado con PHPUnit.

En este caso como se utiliza una base de datos, se utilizara el concepto del Mock para probar su funcionalidad. Un mock es un objeto creado artificialmente que reemplaza a un componente real durante una prueba para simular su comportamiento, permitiendo verificar la lógica interna del módulo que se está probando sin depender de factores externos.

Las técnicas utilizadas serán de tipo de caja blanca y negra para las pruebas de casos de uso.

Casos de Prueba

CU02: CrearMueble

CP02_1 - Crear mueble válido

Este tiene como objetivo de prueba crear un mueble con datos correctos.

La **entrada** seria: nombre="Silla", peso=10, ancho=50, alto=100, largo=60

La **salida esperada** seria: true y se crea el mueble exitosamente.

CP02_2 - Crear mueble faltan campos

Este tiene como objetivo de prueba crear un mueble sin campos requeridos.

La **entrada** seria: nombre="Mesa", peso=15

La **salida esperada** seria: lanzar una excepción por datos incompletos (InvalidArgumentException).

CP02_3 - Crear mueble con strings

Este tiene como objetivo de prueba crear un mueble con datos no numéricos en las dimensiones.

La **entrada** seria: nombre="Mesa", peso=15, ancho="diez", alto="veinte", largo="quince"

La **salida esperada** seria: lanzar una excepción por datos incompletos (InvalidArgumentException).

CU03: ModificarMueble

CP03_1 - Actualizar mueble válido

Este tiene como objetivo de prueba modificar un mueble con datos correctos.

La **entrada** seria: id_mueble=1, nombre="Silla", peso=10, ancho=50, alto=100, largo=60

La **salida esperada** seria: true y el mueble se actualiza correctamente.

CP03_2 - Actualizar mueble inválido

Este tiene como objetivo de prueba modificar mueble con campos incompletos.

La **entrada** seria: id_mueble=1, nombre="Silla", peso=10

La **salida esperada** seria: lanzar una excepción por datos incompletos (InvalidArgumentException).

CU04: EliminarMueble

CP04_1 - Eliminar mueble válido

Este tiene como objetivo de prueba eliminar un mueble con ID existente.

La **entrada** seria: id_mueble=5

La **salida esperada** seria: true y el mueble se elimina correctamente.

CP04_2 - Eliminar mueble inválido

Este tiene como objetivo de prueba eliminar un mueble con ID inexistente.

La **entrada** seria: id_mueble=5

La **salida esperada** seria: false ya que no existe un mueble.

Pruebas adicionales

CP00_1 - Obtener mueble válido

Este tiene como objetivo de prueba obtener un mueble con ID válido.

La **entrada** seria: id_mueble=5

La **salida esperada** seria: objeto de tipo Mueble con los datos correspondientes a ese ID.

CP00_2 - Obtener mueble no encontrado

Este tiene como objetivo de prueba obtener un mueble con un ID inexistente.

La **entrada** seria: id_mueble=5

La **salida esperada** seria: lanzar la excepción (RunTimeException) si no se encuentra el mueble.

CP00_3 - Obtener mueble ID inválido

Este tiene como objetivo de prueba enviar ID no numérico al método.

La **entrada** seria: id_mueble="cinco"

La **salida esperada** seria: lanzar la excepción (RunTimeException) por ID invalido.

Seguimiento y Control

El seguimiento y control del proceso de pruebas se realizará mediante el registro y análisis sistemático de los resultados obtenidos en cada uno de los casos de prueba definidos. Para ello, se obtendrá un artefacto generado por PHPUnit el cual genera una página web que muestra el nombre de las pruebas junto con su resultado en verde si no encontró errores y en rojo si encontró errores.

Los avances se revisarán de forma periódica para garantizar el cumplimiento de la ejecución de pruebas. Esto se dispondrá en la sección de pruebas del repositorio donde se cuenta con todas las pruebas ejecutadas por el GitHub Actions.