

Plan de Pruebas Unitarias

Sistema Kairos Mix

Tienda de frutos secos “Kairos de Dios”

Proyecto: Kairos Mix
Codigo ECS: PP
Version: 1.0.0
Fecha: 14 de enero de 2026
ID Proyecto: 27837_G6_ADS

Equipo de Desarrollo:

Caetano Flores
Jordan Guaman
Anthony Morales
Leonardo Narvaez

Enero 2026

Índice

| | |
|--|----------|
| 1. Introduccion | 2 |
| 1.1. Proposito | 2 |
| 1.2. Alcance | 2 |
| 1.3. Documentos de Referencia | 2 |
| 2. Objetivos de las Pruebas Unitarias | 2 |
| 2.1. Objetivo General | 2 |
| 2.2. Objetivos Especificos | 2 |
| 2.3. Criterios de Exito | 2 |
| 3. Estrategia de Pruebas Unitarias | 3 |
| 3.1. Tipo de Prueba | 3 |
| 3.2. Enfoque | 3 |
| 3.3. Herramientas | 3 |
| 3.4. Responsables | 3 |
| 4. Casos de Prueba Unitarios | 3 |
| 4.1. Modulo: ProductoController | 3 |
| 4.2. Modulo: NutricionalService (Componente Critico) | 3 |
| 4.3. Modulo: PedidoController | 4 |
| 4.4. Modulo: LogObserver | 4 |
| 5. Criterios de Aceptacion | 4 |
| 5.1. Criterios de Entrada | 4 |
| 5.2. Criterios de Salida | 4 |
| 6. Conclusiones | 4 |

1. Introduccion

1.1. Proposito

El presente documento define el plan de pruebas unitarias del sistema Kairos Mix, un aplicativo web desarrollado para gestionar ventas, inventario y pedidos de la tienda de frutos secos “Kairos de Dios”, cuyo principal diferenciador es el cálculo nutricional automático de mezclas personalizadas.

1.2. Alcance

Este plan se limita exclusivamente a la validación de componentes individuales del sistema mediante pruebas unitarias, asegurando que cada módulo funcione de forma correcta e independiente antes de su integración con otros componentes.

1.3. Documentos de Referencia

- SRS - Especificación de Requisitos de Software
- DCD - Diagrama de Clases de Diseño
- ECU - Especificación de Casos de Uso

2. Objetivos de las Pruebas Unitarias

2.1. Objetivo General

Verificar el correcto funcionamiento de los componentes individuales del sistema Kairos Mix, validando la lógica interna de cada módulo de forma aislada.

2.2. Objetivos Especificos

- Validar la lógica de negocio de los controladores principales
- Garantizar la precisión del componente crítico NutricionalService
- Comprobar la correcta gestión de datos en operaciones CRUD
- Verificar el funcionamiento del patrón Observer a nivel de componente

2.3. Criterios de Exito

- Cobertura de código igual o superior al 80 %
- Todos los casos de prueba unitarios ejecutados exitosamente
- Ausencia de defectos críticos en componentes evaluados

3. Estrategia de Pruebas Unitarias

3.1. Tipo de Prueba

Pruebas Unitarias

3.2. Enfoque

Cada módulo del sistema será probado de manera aislada, utilizando datos controlados y simulando dependencias externas cuando sea necesario.

3.3. Herramientas

- Backend: Node.js con Jest y Mocha
- Simulación de dependencias: Mocks y Stubs

3.4. Responsables

Cada desarrollador es responsable de implementar y ejecutar las pruebas unitarias de los módulos que desarrolló.

4. Casos de Prueba Unitarios

4.1. Modulo: ProductoController

| ID | Descripcion | Resultado Esperado |
|-----------|----------------------------------|-------------------------------|
| TC-PR-U01 | Crear producto con datos válidos | Producto creado correctamente |
| TC-PR-U02 | Validar campos obligatorios | Error de validación |
| TC-PR-U03 | Actualizar precio de producto | Precio actualizado |
| TC-PR-U04 | Eliminar producto existente | Producto eliminado |
| TC-PR-U05 | Consultar stock | Stock devuelto correctamente |

Cuadro 1: Casos de prueba unitarios - ProductoController

4.2. Modulo: NutricionalService (Componente Critico)

| ID | Descripcion | Resultado Esperado |
|-----------|-------------------------------|----------------------------|
| TC-NS-U01 | Calcular calorías totales | Resultado exacto |
| TC-NS-U02 | Calcular proteínas ponderadas | Valor correcto |
| TC-NS-U03 | Calcular precio total | Cálculo correcto |
| TC-NS-U04 | Validar suma de porcentajes | Error si suma \neq 100 % |

| | | |
|-----------|-------------------|-----------------------|
| TC-NS-U05 | Precisión decimal | Dos decimales exactos |
|-----------|-------------------|-----------------------|

Cuadro 2: Casos de prueba unitarios - NutricionalService

4.3. Modulo: PedidoController

| ID | Descripcion | Resultado Esperado |
|-----------|-------------------------|-------------------------|
| TC-PE-U01 | Crear pedido | Estado inicial correcto |
| TC-PE-U02 | Cambiar estado válido | Transición exitosa |
| TC-PE-U03 | Cambiar estado inválido | Error de validación |
| TC-PE-U04 | Calcular total | Total correcto |

Cuadro 3: Casos de prueba unitarios - PedidoController

4.4. Modulo: LogObserver

| ID | Descripcion | Resultado Esperado |
|-----------|----------------------------|---------------------------|
| TC-LO-U01 | Registrar creación | Log generado |
| TC-LO-U02 | Registrar cambio de estado | Log con valores correctos |
| TC-LO-U03 | Validar formato del log | Estructura JSON válida |

Cuadro 4: Casos de prueba unitarios - LogObserver

5. Criterios de Aceptacion

5.1. Criterios de Entrada

- Código fuente disponible
- Módulos implementados
- Herramientas de pruebas configuradas

5.2. Criterios de Salida

- Casos de prueba unitarios ejecutados correctamente
- Cobertura de código alcanzada
- Defectos críticos corregidos

6. Conclusiones

El plan de pruebas unitarias permitió validar de forma efectiva la correcta implementación de los componentes individuales del sistema Kairos Mix. Los resultados obtenidos

aseguran que la lógica de negocio, en especial el cálculo nutricional, cumple con los requisitos definidos y proporciona una base sólida para las siguientes etapas del proceso de pruebas.

Caetano Flores
Lider de Pruebas

Jordan Guaman
Tester Backend

Anthony Morales
Tester Backend

Leonardo Narvaez
Tester Backend