

Plan de Pruebas

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

1.1. Proposito

Este documento define el plan de pruebas para el sistema Kairos Mix, un aplicativo web diseñado para digitalizar los procesos de venta, gestion de inventario y pedidos de la tienda de frutos secos "Kairos de Dios", con un diferenciador competitivo: el diseñador de mezclas personalizadas con calculo nutricional automatico.

1.2. Alcance

El plan de pruebas abarca:

- Pruebas unitarias de componentes individuales
- Pruebas de integracion entre capas del sistema
- Pruebas funcionales de casos de uso
- Pruebas del diferenciador: NutricionalService
- Pruebas de seguridad y auditoria (LogObserver)

1.3. Documentos de Referencia

- SRS - Especificacion de Requisitos de Software (G6_SRS_KairosMix.pdf)
- DCD - Diagrama de Clases de Diseño (ClassDiagKairo_G6_V2.pdf)
- ECU - Especificacion de Casos de Uso (G6_CasosDeUso.pdf)
- DAS - Descripcion de Arquitectura (Arquitectura_G6_V2.pdf)

2. Objetivos de las Pruebas

2.1. Objetivos Generales

1. Verificar que todos los requisitos funcionales estan correctamente implementados
2. Validar la precision del algoritmo de calculo nutricional (componente critico)
3. Garantizar la integridad de los datos en todas las operaciones CRUD
4. Asegurar el correcto funcionamiento del patron Observer para auditoria
5. Validar la comunicacion entre las 3 capas del sistema

2.2. Criterios de Exito

- Cobertura de codigo $\geq 80\%$
- 0 defectos criticos en produccion
- Precision del calculo nutricional: 100% exactitud
- Todos los casos de prueba de alta prioridad ejecutados exitosamente
- Registro correcto de logs para operaciones criticas

3. Estrategia de Pruebas

3.1. Niveles de Prueba

3.1.1. Pruebas Unitarias

Objetivo: Validar componentes individuales de forma aislada.

Herramientas:

- Backend (Node.js): Jest / Mocha
- Frontend (React): Jest + React Testing Library

Responsable: Cada desarrollador prueba sus propios componentes.

3.1.2. Pruebas de Integracion

Objetivo: Verificar la comunicacion entre modulos y capas.

Enfoque: Pruebas de API REST, integracion con base de datos MySQL.

3.1.3. Pruebas Funcionales

Objetivo: Validar que el sistema cumple con los casos de uso definidos.

Basadas en: Especificacion de Casos de Uso (ECU).

3.2. Tipos de Prueba

Tipo	Nivel	Descripcion
Funcionales	Unitaria	Validacion de logica de negocio
Integracion	Integracion	Comunicacion entre capas
Regresion	Unitaria/Integracion	Verificar que nuevos cambios no rompen funcionalidad existente
Seguridad	Unitaria	Validacion de entradas, registro de logs
Performance	Sistema	Tiempo de respuesta del calculo nutricional

Cuadro 1: Tipos de prueba a ejecutar

4. Casos de Prueba

4.1. Modulo: ProductoController

ID	Descripcion	Entrada	Resultado Esperado
TC-PR-001	Crear producto con datos validos	nombre, categoria, precio, stock	Producto creado, status 201
TC-PR-002	Crear producto sin nombre	categoria, precio, stock	Error de validacion, status 400
TC-PR-003	Actualizar precio de producto	idProducto, nuevo-Precio	Precio actualizado correctamente
TC-PR-004	Eliminar producto existente	idProducto	Producto eliminado, status 200
TC-PR-005	Consultar stock de producto	idProducto	Valor de stock correcto

Cuadro 2: Casos de prueba - ProductoController

4.2. Modulo: NutricionalService (CRITICO)

ID	Descripcion	Entrada	Resultado Esperado
TC-NS-001	Calcular calorías totales de mezcla	ingredientes con valores nutricionales, proporciones	Suma exacta de calorías
TC-NS-002	Calcular proteínas basadas en proporciones	50 % Almendras (20g prot), 50 % Nueces (15g prot)	17.5g proteínas totales
TC-NS-003	Calcular precio total de mezcla	Ingrediente A (\$10, 40 %), Ingrediente B (\$15, 60 %)	\$13.00 total
TC-NS-004	Validar suma de porcentajes = 100 %	proporciones: 30 %, 40 %, 20 %	Error: suma != 100 %
TC-NS-005	Calcular grasas de mezcla compleja	3 ingredientes con diferentes valores	Calculo correcto ponderado
TC-NS-006	Calcular carbohidratos	ingredientes, proporciones	Suma ponderada correcta
TC-NS-007	Verificar precisión decimal	valores con decimales	Precisión hasta 2 decimales

Cuadro 3: Casos de prueba - NutricionalService (COMPONENTE CRITICO)

4.3. Modulo: PedidoController

ID	Descripcion	Entrada	Resultado Esperado
TC-PE-001	Crear pedido con estado inicial	cliente, productos	Estado = "Por preparar"
TC-PE-002	Transicion valida de estados	estado actual: "Por preparar", nuevo: "Preparado"	Cambio exitoso
TC-PE-003	Transicion invalida de estados	estado actual: ".Entregado", nuevo: "Por preparar"	Error: transicion no permitida
TC-PE-004	Calcular total del pedido	lista de productos con precios	Suma correcta del total
TC-PE-005	Generar reporte de pedido	idPedido	Reporte con datos completos

Cuadro 4: Casos de prueba - PedidoController

4.4. Modulo: MezclaController

ID	Descripcion	Entrada	Resultado Esperado
TC-MZ-001	Crear mezcla personalizada	nombre, ingredientes, proporciones	Mezcla creada
TC-MZ-002	Agregar ingrediente a mezcla	idMezcla, idIngrediente, proporcion	Ingrediente agregado
TC-MZ-003	Invocar calculo nutricional	idMezcla	Valores nutricionales calculados
TC-MZ-004	Guardar mezcla como pedido	idMezcla, idCliente	Pedido creado desde mezcla

Cuadro 5: Casos de prueba - MezclaController

4.5. Modulo: LogObserver (Patron Observer)

ID	Descripcion	Entrada	Resultado Esperado
TC-LO-001	Registrar creacion de producto	ProductoController.create()	Log generado con timestamp
TC-LO-002	Registrar cambio de estado de pedido	PedidoController.cambiarEstado()	Log con estado anterior y nuevo
TC-LO-003	Registrar eliminacion de producto	ProductoController.eliminar()	Log con datos del producto eliminado

ID	Descripcion	Entrada	Resultado Esperado
TC-LO-004	Verificar formato de log	cualquier operacion critica	JSON con entidad, operacion, usuario, timestamp

Cuadro 6: Casos de prueba - LogObserver

4.6. Frontend: Componentes React

ID	Descripcion	Entrada	Resultado Esperado
TC-UI-001	Renderizar lista de productos	array de productos	Productos mostrados en pantalla
TC-UI-002	Validar formulario de producto	campos vacios	Mensajes de error visibles
TC-UI-003	Diseñador de mezclas - agregar ingrediente	click en ingrediente	Ingrediente aparece en lista de seleccion
TC-UI-004	Mostrar valores nutricionales	mezcla calculada	Tabla nutricional visible
TC-UI-005	Cambiar estado de pedido	click en boton de estado	Estado actualizado en UI

Cuadro 7: Casos de prueba - Componentes Frontend

5. Priorizacion de Casos de Prueba

5.1. Prioridad Critica

- **TC-NS-001 a TC-NS-007:** Calculo nutricional (diferenciador del sistema)
- **TC-PR-001:** Crear producto
- **TC-PE-001, TC-PE-002:** Gestion de pedidos y estados
- **TC-LO-001 a TC-LO-004:** Auditoria y seguridad

5.2. Prioridad Alta

- **TC-MZ-001 a TC-MZ-004:** Diseñador de mezclas
- **TC-PR-003, TC-PR-005:** Operaciones CRUD de productos
- **TC-UI-003, TC-UI-004:** UI del diseñador de mezclas

5.3. Prioridad Media

- **TC-PR-002, TC-PR-004:** Validaciones y eliminaciones
- **TC-PE-003, TC-PE-004, TC-PE-005:** Funciones adicionales de pedidos
- **TC-UI-001, TC-UI-002, TC-UI-005:** Componentes UI generales

6. Recursos y Responsabilidades

6.1. Equipo de Pruebas

Rol	Responsable
Lider de Pruebas	Caetano Flores
Tester Backend	Jordan Guaman
Tester Frontend	Anthony Morales
Tester de Integracion	Leonardo Narvaez

Cuadro 8: Equipo de pruebas

6.2. Entorno de Pruebas

- **Sistema Operativo:** Windows
- **Backend:** Node.js v18+
- **Frontend:** React 18+
- **Base de Datos:** MySQL 8.0 (instancia de prueba)
- **Framework de Pruebas:** Jest, Mocha, React Testing Library
- **CI/CD:** GitHub Actions (opcional)

7. Cronograma de Pruebas

Fase	Duracion	Actividades
Preparacion	Semana 1	Configuracion de entorno, preparacion de datos
Pruebas Unitarias	Semana 2-3	Ejecucion de TC-PR, TC-NS, TC-PE, TC-MZ
Pruebas de Integracion	Semana 4	Pruebas de API y base de datos
Pruebas Funcionales	Semana 5	Validacion de casos de uso completos
Pruebas de Regresion	Semana 6	Re-ejecucion de casos criticos
Reporte Final	Semana 6	Documentacion de resultados

Cuadro 9: Cronograma de pruebas

8. Criterios de Aceptacion

8.1. Criterios de Entrada

- Codigo fuente completo y desplegado en entorno de pruebas
- Base de datos configurada con datos de prueba
- Documentacion de diseño disponible (DCD, DAS, ECU)
- Herramientas de prueba instaladas y configuradas

8.2. Criterios de Salida

- Todos los casos de prueba de prioridad critica ejecutados exitosamente
- 0 defectos criticos abiertos
- Cobertura de codigo $\geq 80\%$
- Reporte de pruebas completo y aprobado
- Precision del NutricionalService validada al 100 %

9. Gestion de Defectos

9.1. Clasificacion de Severidad

Severidad	Descripcion
Critico	Sistema no funcional, calculo nutricional incorrecto, perdida de datos
Alta	Funcionalidad principal afectada, work-around complicado
Media	Funcionalidad secundaria afectada, work-around disponible
Baja	Problemas cosmeticos o de UI

Cuadro 10: Niveles de severidad

9.2. Proceso de Reporte

1. Deteccion del defecto durante prueba
2. Registro en sistema de seguimiento (carpeta 1.8 Reportes de errores)
3. Asignacion a desarrollador responsable
4. Correccion del defecto
5. Re-prueba y verificacion
6. Cierre del defecto

10. Entregables

1. Plan de Pruebas (este documento)
2. Casos de Prueba Detallados (ECP)
3. Scripts de prueba automatizadas
4. Reporte de Ejecucion de Pruebas
5. Reporte de Defectos
6. Reporte de Cobertura de Codigo
7. Informe Final de Pruebas

11. Riesgos y Mitigacion

Riesgo	Impacto	Mitigacion
Error en algoritmo de calculo nutricional	Critico - Funcionalidad principal incorrecta	Pruebas exhaustivas con datos reales, validacion matematica
Falta de datos de prueba	Alto - Retraso en pruebas	Preparar dataset de prueba con anticipacion
Cambios de requisitos durante pruebas	Medio - Re-trabajo	Congelar requisitos antes de inicio de pruebas
Falta de tiempo para pruebas completas	Alto - Pruebas incompletas	Priorizar casos criticos primero

Cuadro 11: Riesgos y mitigacion

12. Conclusiones

Este plan de pruebas asegura la calidad del sistema Kairos Mix mediante:

- Cobertura exhaustiva de todos los modulos criticos
- Enfoque especial en el diferenciador: NutricionalService
- Validacion de seguridad mediante LogObserver
- Pruebas de integracion entre las 3 capas del sistema
- Proceso estructurado de gestion de defectos

La ejecucion exitosa de este plan garantizara un sistema robusto, confiable y listo para produccion.

Caetano Flores
Lider de Pruebas

Jordan Guaman
Tester Backend

Anthony Morales
Tester Frontend

Leonardo Narvaez
Tester de Integracion