



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS

6NM60 Ingeniería de Pruebas

Proyecto, desarrollo de una página de e-commerce.

Plan ACS/SQA

Alumnos:

García Méndez Juan Carlos

Conde Basilio Leonardo

Ramos Velázquez Felipe

Villaseñor Trejo Javier Enrique

Docente:

Cruz Martínez Ramón



Fecha de entrega: 09 de mayo del 2025

Contenido

| | |
|--|----------|
| 1. Introducción | 3 |
| 2. Objetivos de Calidad | 3 |
| 3. Alcance..... | 4 |
| 4. Estándares y Normativas..... | 4 |
| 5. Metodología..... | 5 |
| 6. Roles y Responsabilidades..... | 5 |
| 7. Herramientas y Tecnologías | 6 |
| 8. Procesos de Calidad..... | 6 |
| 9. Gestión de Riesgos | 7 |
| 10. Métricas y KPIs | 7 |
| 11. Auditorías de Calidad | 8 |
| 12. Capacitación | 8 |
| 13. Documentación | 8 |
| 14. Plan de Mejora Continua..... | 9 |
| 15. Conclusión | 9 |

1. Introducción

Este documento define las estrategias, procesos y estándares para garantizar la calidad del software en el desarrollo y mantenimiento de **ClothesShop**, una plataforma de e-commerce especializada en moda. El plan se alinea con los requerimientos funcionales (CRUD del carrito, simulación de cobro, historial de pedidos) y no funcionales (rendimiento, seguridad, usabilidad), asegurando cumplimiento con normativas internacionales (GDPR, PCI-DSS) y satisfacción del usuario final.

2. Objetivos de Calidad

- **Funcionalidad:**
 - Garantizar operaciones CRUD completas en el carrito (agregar, eliminar, modificar).
 - Asegurar simulación de cobro realista con Stripe Sandbox (100% cobertura de casos de prueba).
- **Rendimiento:**
 - Tiempo de carga de catálogo ≤ 2 segundos (optimización de imágenes WebP).
 - Soporte para 500 transacciones simultáneas sin degradación.
- **Usabilidad:**
 - Puntuación $\geq 90/100$ en pruebas de usabilidad (métricas Lighthouse).
 - Reducción del 30% en la tasa de abandono de carritos.
- **Seguridad:**
 - Cero vulnerabilidades críticas (OWASP Top 10).
 - Encriptación AES-256 para datos sensibles.

3. Alcance

- **Módulos Cubiertos:**
 - Gestión de productos (catálogo, filtros, búsqueda con Algolia).
 - Carrito de compras (CRUD, persistencia en localStorage/Firebase).
 - Checkout simulado (Stripe Sandbox, generación de tickets).
 - Autenticación (JWT, OAuth con Google/Facebook).
 - Historial de pedidos (descarga de PDF, reordenamiento).
- **Exclusiones:**
 - Integración con redes sociales (planeada para v3.0).
 - Sistema de recomendaciones basado en IA (en desarrollo).

4. Estándares y Normativas

- **Desarrollo:**
 - Código: ES6+ (JavaScript), PEP8 (Python), PSR (PHP).
 - Accesibilidad: WCAG 2.1 Nivel AA.
- **Pruebas:**
 - Cobertura $\geq 80\%$ (Jest, Cypress).
 - Pruebas de carga con JMeter (1000 usuarios concurrentes).
- **Documentación:**
 - Manual de usuario en Markdown y PDF.
 - Diagramas UML (Casos de Uso, Secuencia, Componentes).

5. Metodología

- **Modelo Híbrido (Scrum + Kanban):**
 - **Sprints de 2 semanas** con entregables priorizados mediante *MoSCoW*.
 - **Daily Standups** para sincronizar avances (Zoom/Slack).
 - **Tableros Jira:** *Backlog, En Progreso, Revisión, Done*.
- **Control de Versiones:**
 - **GitFlow** con ramas: main, develop, feature/carrito-crud.
 - **Convención de Commits:** <tipo>(<ámbito>): <descripción> (ej: feat(cart): add delete item function).

6. Roles y Responsabilidades

| Rol | Responsabilidades |
|-------------------------------|--|
| Líder de Calidad | Supervisar pruebas, auditorías y cumplimiento de estándares. |
| Desarrollador Frontend | Implementar componentes React (Carrito, Búsqueda) con pruebas unitarias. |
| Desarrollador Backend | Gestionar APIs (Node.js), integración con Firebase/Stripe. |
| QA Engineer | Ejecutar pruebas E2E (Cypress), reportar bugs en Jira. |
| UX Designer | Validar prototipos en Figma con usuarios reales (tests A/B). |

7. Herramientas y Tecnologías

- **Desarrollo:**
 - Frontend: React, Redux, Algolia InstantSearch.
 - Backend: Node.js, Express, Firebase.
 - DevOps: Docker, AWS EC2, GitHub Actions.
- **Pruebas:**
 - Unitarias: Jest, React Testing Library.
 - E2E: Cypress (flujos de compra).
 - Carga: JMeter, K6.
- **Monitoreo:**
 - New Relic (rendimiento), Kibana (logs), Sentry (errores).

8. Procesos de Calidad

8.1. Revisión de Código

- **Peer Review:** Dos aprobaciones requeridas para merge en main.
- **Checklist:**
 - Código documentado (JSDoc).
 - Sin deuda técnica (SonarQube).
 - Cumplimiento de convenciones (ESLint, Prettier).

8.2. Pruebas Automatizadas

- **Estrategia:**
 - **Unitarias:** Validar funciones críticas (ej: cálculo de totales).
 - **Integración:** Flujos usuario-invitado → registro → checkout.
 - **Regresión:** Ejecutar suite completa tras cada commit.

8.3. Pruebas de Usabilidad

- **Participantes:** 20 usuarios reales (10 registrados, 10 invitados).
- **Métricas:**
 - Tiempo para completar compra (objetivo: ≤ 3 minutos).
 - Satisfacción (Escala Likert: 1-5).

9. Gestión de Riesgos

| Riesgo | Probabilidad | Impacto | Mitigación |
|----------------------------------|--------------|---------|--|
| Fallo en integración Stripe | Media | Alto | Pruebas sandbox rigurosas + rollback automático. |
| Pérdida de datos en localStorage | Baja | Medio | Backup diario en Firebase para usuarios invitados. |
| Ataque DDoS | Baja | Crítico | Configuración de AWS Shield + balanceo de carga. |
| Errores de redondeo en precios | Alta | Alto | Uso de Decimal.js para cálculos monetarios. |

10. Métricas y KPIs

| Métrica | Objetivo | Herramienta de Medición |
|---------------------------|-------------------|-------------------------|
| Cobertura de pruebas | $\geq 80\%$ | Jest/Istanbul |
| Tiempo medio de respuesta | ≤ 500 ms | New Relic |
| Defectos por sprint | ≤ 5 críticos | Jira |
| CSAT (Satisfacción) | $\geq 4.5/5$ | Encuestas post-compra |

11. Auditorías de Calidad

- **Internas:** Mensuales, enfocadas en código, seguridad y UX.
 - Ejemplo: Revisión de tokens JWT para prevenir ataques CSRF.
- **Externas:** Anuales, realizadas por terceros (ej: auditoría PCI-DSS).

12. Capacitación

- **Programa de Entrenamiento:**
 - Cursos en Udemy: React Avanzado, Seguridad en Firebase.
 - Talleres prácticos: Escritura de casos de prueba en Cypress.
- **Certificaciones:**
 - AWS Certified Developer, Scrum Master (PSM I).

13. Documentación

- **Manual de Usuario:** Incluye glosario, atajos y guías paso a paso.
- **API Docs:** Swagger/OpenAPI para endpoints de pedidos y autenticación.
- **Registro de Cambios (Changelog):** Actualizado en GitHub tras cada release.

14. Plan de Mejora Continua

- **Retrospectivas:** Al final de cada sprint para identificar áreas de mejora.
- **Feedback de Usuarios:** Incorporar sugerencias en el backlog (ej: filtro por temporada).
- **Benchmarking:** Análisis trimestral vs. competidores (ej: rendimiento de carga).

15. Conclusión

Este plan ACS/SQA asegura que **ClothesShop** cumpla con los más altos estándares de calidad, desde el código hasta la experiencia del usuario. La combinación de metodologías ágiles, pruebas automatizadas y monitoreo proactivo garantiza una plataforma escalable, segura y centrada en el cliente, lista para posicionarse como líder en su nicho de mercado.