

Plan de Pruebas

“PentaDocs”

Versión	Fecha	Descripción	Elaboradores
1.0	23/05/2025	Primera versión	Condori Gonzales, Jean Espinoza Villanueva, Alexis Murillo Castillo, Alexander Salas Sotillo, Santiago Salazar Zuñiga, Camila

Controlador:

- Guillermo Enrique Calderón Ruiz

Índice general

1. Introducción	3
2. Alcance	4
2.1. Alcance del Plan de Pruebas	5
2.1.1. Tipos de Pruebas	5
2.1.2. Niveles de Prueba	6
2.1.3. Exclusiones	6
2.2. Gestión del alcance	6
3. Requisitos	8
3.1. Enfoque basado en requisitos	9
3.2. Tipos de requisitos considerados	9
3.3. Criterios de aceptación	9
3.4. Trazabilidad de requisitos	10
3.4.1. Matriz de Trazabilidad de Requisitos	10
3.5. Gestión de requisitos	12
4. Cronograma	14
4.1. Actividades	15
4.2. Gestión de cronograma	16
5. Costos	18
5.1. Introducción	18
5.2. Metodología de Costeo	18
5.3. Detalle Individual de Costos	19
5.3.1. Recursos Humanos (Costos Unitarios)	19
5.3.2. Infraestructura (Costos Unitarios)	19
5.4. Generalización para el Equipo	19
5.4.1. Personal (Equipo Completo)	19
5.4.2. Infraestructura (Consolidado)	20
5.5. Presupuesto Consolidado	20
5.6. Distribución por Actividad	21
5.7. Gestión de costos	21
6. Recursos	23
6.1. Recursos humanos	24
6.2. Recursos físicos	24
6.3. Gestión de recursos	25

7. Diseño de casos de prueba	27
7.1. Pruebas unitarias	28
7.2. Pruebas de integración	29
7.3. Pruebas de validación	32
7.4. Pruebas de sistema	33

Capítulo 1

Introducción

Este documento es el plan de pruebas para el sistema de descargas de contenido **QuickContentMedia**, desarrollado por la empresa **PentaDocs**. El objetivo principal de este documento es organizar, estructurar y garantizar la correcta verificación y validación del sistema mediante la planificación y ejecución de distintos tipos de pruebas.

Este plan se ha elaborado para establecer con claridad los criterios, técnicas, recursos y responsabilidades necesarias para evaluar que el software cumpla con los requisitos funcionales y no funcionales definidos. Sirve como una guía detallada para asegurar que los componentes del sistema funcionen correctamente tanto de forma aislada como integrados, reduciendo el riesgo de errores en producción y asegurando una experiencia satisfactoria para el usuario final.

El documento está compuesto por apartados clave como el alcance del plan de pruebas, donde se definen los niveles de prueba a utilizar y sus exclusiones; un cronograma detallado de actividades; una estimación de costos asociados; los recursos humanos y físicos involucrados; y finalmente, el diseño de los casos de prueba.

En conjunto, este plan de pruebas busca ser un instrumento de control y mejora continua durante el proceso de verificación y validación del sistema, facilitando la toma de decisiones, el aseguramiento de la calidad y el cumplimiento de los objetivos del proyecto.

Capítulo 2

Alcance

Esta sección define el alcance del Plan de Pruebas del sistema **QuickContentMedia**, estableciendo los límites y objetivos de las actividades de verificación y validación. El propósito principal es asegurar que los componentes desarrollados del sistema funcionen correctamente, cumplan con los requisitos especificados y estén listos para su despliegue.

Se especifican los tipos de pruebas a realizar, los niveles en los que serán aplicadas, así como las funcionalidades incluidas y aquellas que quedarán fuera del proceso de prueba. Este enfoque permite enfocar los esfuerzos del equipo de aseguramiento de calidad en los elementos críticos del sistema, garantizando una cobertura adecuada y eficiente de los escenarios más relevantes desde la perspectiva técnica y del usuario final.

2.1. Alcance del Plan de Pruebas

2.1.1. Tipos de Pruebas

■ Pruebas Unitarias

- **Definición:** Prueba individual de cada función/método identificado en:
 - Métodos de clases en diagramas de diseño (Sección 4.2 Doc. Ejecución)
- **Ejemplo:**
 - Función `validarUsername()` en `gestorUsuario` (G-001) para RU-001
 - Función `aplicarDescuento()` en `gestorCompra` (G-002) para RU-011

■ Pruebas de Integración

- **Definición:** Verificar interacción entre funciones según:
 - Diagramas de secuencia (Sección 4.1 Doc. Ejecución)
 - Flujos entre gestores (Ej: G-001 → G-004 → G-005)
- **Ejemplo:**
 - Integración `gestorUsuario` + `gestorPago` para RF-004 (Recarga saldo)
 - Flujo completo de compra (G-006 → G-002 → G-005)

■ Pruebas de Validación

- **Definición:** Demostración al cliente (Guillermo Calderón) de que el sistema cumple los requisitos.
- **Metodología:**
 - Sesiones guiadas usando mockups (Sección 3.3.2)
 - Checklist basado en Tabla 3.1 (Requisitos usuario)

■ Pruebas de Sistema

- **Definición:** Verificar comportamiento en entorno real:
 - Hardware: Según especificaciones en Sección 6.2 (i5/Ryzen 5, 8GB RAM)
 - Software: PostgreSQL 17.4 + Railway (Sección 6.2)
- **Enfoque:**
 - Rendimiento con 5-20 usuarios (RNF-004)
 - Compatibilidad navegadores (RNF-003)
- **Participantes:**
 - 1 Tester Senior (5h totales)
 - DevOps (soporte técnico para configuración)

2.1.2. Niveles de Prueba

- **Pruebas Alfa:**
 - Realizadas por: Equipo de testing (Tabla 6.4 - Roles)
 - Ambiente: Desarrollo (Railway - Sección 5.1 Doc. Ejecución)
 - Cobertura: 100 % funciones críticas (RF con prioridad Alta)
- **Pruebas Beta:**
 - Participantes: 3 clientes reales (según perfil 2.2.1)
 - Casos: Flujos principales (CU-001 a CU-005)
 - Duración: 1 semana (alineado con Cronograma - Tabla 4.1)

2.1.3. Exclusiones

- No se probará en hardware inferior a lo especificado (i3 no cubierto)
- No se validarán navegadores fuera de los listados en RNF-003
- No se incluyen pruebas de estrés más allá de 20 usuarios (límite RNF-004)

2.2. Gestión del alcance

La gestión de cambios en el alcance de pruebas es fundamental para asegurar que las actividades de verificación y validación del sistema **QuickContentMedia** se desarrollen según lo planeado, permitiendo identificar a tiempo modificaciones que puedan afectar los objetivos acordados. Se realizarán revisiones periódicas del alcance definido, evaluando su cumplimiento y las posibles desviaciones.

- **Monitoreo y Detección de Cambios en el Alcance:** Se llevarán a cabo revisiones periódicas del alcance establecido, detectando adiciones, eliminaciones o modificaciones en los requisitos a validar.
- **Comparación y Evaluación del Cambio:** Se contrastarán los requisitos originales con los propuestos. Si se detectan discrepancias, se documentarán las causas (ej: nuevos requerimientos del cliente, cambios técnicos, etc.).
- **Acciones Correctivas:** Se tomarán medidas como reasignación de recursos o ajustes en los casos de prueba para adaptarse a los cambios.
- **Acciones Preventivas:** Se reforzarán prácticas como la revisión temprana de requisitos para reducir futuros impactos.
- **Actualización del Alcance:** Cualquier modificación deberá ser aprobada por todo el equipo y el cliente. Una vez aprobada, se reflejará el cambio en la documentación y se notificará a todos los involucrados.

Tabla 2.1 – Ejemplo de registro de cambios en el alcance de pruebas de **QuickContentMedia**

Código	Fecha	Revi- sor(es)	Requi- sito Afecta- do	Causa del Cambio	Acciones propues- tas	Estado	Fecha de reso- lución
SCA-01	30/05/25	Alexis Espi- noza	RF-005	Nuevo re- querimiento del cliente	Añadir 3 casos de prueba	Aproba- do	01/06/25
SCA-02	02/06/25	Santiago Salas	RNF- 003	Limitación técnica	Excluir prueba en Edge Le- gacy	En eva- luación	-

Capítulo 3

Requisitos

Esta sección detalla cómo se verificará el cumplimiento de los requisitos del sistema **Quick-ContentMedia** a través de actividades de prueba. Los requisitos considerados incluyen aspectos funcionales, no funcionales y de calidad, todos documentados en la etapa previa del proyecto. Las pruebas se han diseñado para cubrir estos requisitos mediante criterios de aceptación claros y una matriz de trazabilidad que permita garantizar una cobertura completa.

3.1. Enfoque basado en requisitos

Las pruebas se han diseñado considerando los requisitos del proyecto, los cuales se validarán para asegurar que:

- Las funcionalidades del sistema se comporten como se espera.
- Las condiciones de uso (rendimiento, seguridad, etc.) se cumplan en los entornos previstos.
- El producto entregado cumpla con los criterios de calidad establecidos.

3.2. Tipos de requisitos considerados

Los tipos de requisitos considerados se muestran en la tabla 3.1.

Tabla 3.1 – Clasificación de requisitos y su estrategia de prueba

Tipo de requisito	Descripción	Estrategia de prueba
Funcionales	Especifican las funciones, procesos o comportamientos que el sistema debe realizar (ej. procesar un pago, mostrar un listado)	Se verificarán mediante pruebas funcionales y pruebas de validación para asegurar que cada función opera según lo especificado en la documentación de requisitos.
No funcionales	Relacionados con las cualidades del sistema y las condiciones bajo las que opera (ej. rendimiento, seguridad, usabilidad, mantenibilidad)	Se verificarán mediante pruebas de rendimiento, pruebas de seguridad, pruebas de usabilidad, etc., según aplique a cada requisito no funcional específico.
De Calidad	Incluyen criterios o condiciones que permiten evaluar si un entregable es aceptable en términos de funcionalidad o desempeño esperado.	Verificación contra métricas y condiciones predefinidas (por ejemplo: tiempos, experiencia de usuario).

3.3. Criterios de aceptación

Cada requisito será considerado cumplido si:

- El caso de prueba asociado ha sido ejecutado con resultado exitoso.
- El comportamiento observado cumple los valores o límites establecidos en los criterios definidos.
- Se han documentado las evidencias correspondientes.

3.4. Trazabilidad de requisitos

Para garantizar que todos los requisitos sean adecuadamente cubiertos por los casos de prueba, se emplea una **Matriz de Trazabilidad de Requisitos**, la cual vincula los requisitos funcionales y no funcionales con los casos de prueba diseñados y con los criterios de aceptación correspondientes.

Esta matriz permite:

- Verificar la cobertura completa de los requisitos.
- Monitorear el estado de cumplimiento de cada requisito durante las pruebas.
- Identificar rápidamente el impacto en pruebas cuando se actualice un requisito.

3.4.1. Matriz de Trazabilidad de Requisitos

La Tabla 3.2 contiene la **Matriz de Trazabilidad**, donde cada requisito tiene un identificador, su descripción, el tipo de requisito, los casos de prueba asociados, los criterios de aceptación y su prioridad.

Tabla 3.2 – Matriz de trazabilidad de requisitos para el plan de pruebas del sistema **QuickContentMedia**

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RF-001	Registro y autenticación de usuarios con validación de username y redirección por rol.	Funcional		Acceso exitoso según tipo de usuario; datos almacenados correctamente.	Alta
RF-002	Visualización de contenidos por tipo, con filtros por autor y categoría.	Funcional		Tarjetas cargadas correctamente; filtros aplicados correctamente.	Alta
RF-003	Configuración de cuenta: ver perfil, historial, cambio de contraseña y eliminación.	Funcional		Información mostrada correctamente; operaciones realizadas con éxito.	Media
RF-004	Recarga de saldo mediante pasarela de pago, con opciones predefinidas.	Funcional		Saldo actualizado correctamente; mensaje de éxito/error mostrado.	Alta
RF-005	Gestión del carrito: agregar/quitar productos, aplicar descuentos, regalar, pagar.	Funcional	PU-001, PI-001, PV-002	Compra realizada correctamente; descuentos y regalos aplicados.	Alta

Continúa en la siguiente página

Tabla 3.2 – continuación desde la página anterior

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RF-006	Visualización, descarga y calificación de contenidos adquiridos.	Funcional		Descarga funcional; calificación registrada una sola vez.	Alta
RF-007	Rankings semanales de contenidos y semestrales de clientes.	Funcional		Rankings generados y actualizados correctamente.	Media
RF-008	Gestión de clientes: visualizar, buscar, modificar saldo.	Funcional	PU-002, PI-002, PV-001	Lista de clientes mostrada; saldo modificado correctamente.	Alta
RF-009	Gestión de promociones: ver, buscar, agregar, editar y eliminar promociones.	Funcional		Promoción válida; no se permite duplicidad de contenidos.	Alta
RF-010	Gestión de contenidos: visualizar, buscar, agregar, editar, eliminar, capturar metadatos.	Funcional		Contenido cargado y editado correctamente; metadatos identificados.	Alta
RF-011	Gestión jerárquica de categorías: visualizar estructura, agregar y renombrar categorías.	Funcional		Estructura coherente; renombrado sin errores.	Alta
RNF-001	Seguridad: control de acceso por roles y políticas de contraseña seguras.	No funcional	PS-002	Accesos restringidos correctamente; contraseña verificada para cambios.	Alta
RNF-002	Usabilidad: interfaz coherente y aprendizaje en menos de 5 minutos.	No funcional		Flujo intuitivo validado por pruebas; tareas completadas en tiempo.	Alta

Continúa en la siguiente página

Tabla 3.2 – continuación desde la página anterior

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RNF-003	Portabilidad: funcionamiento en múltiples navegadores y SO.	No funcional	PS-001	Pruebas cruzadas exitosas; compatibilidad verificada.	Alta
RNF-004	Escalabilidad: mantener rendimiento con crecimiento de usuarios y contenidos.	No funcional		Operaciones rápidas (10 ms – 1s); sin degradación perceptible.	Alta
RNF-005	Mantenibilidad: actualizaciones aplicadas en máximo 3 días.	No funcional		Tareas de mantenimiento registradas y aplicadas sin errores.	Alta

3.5. Gestión de requisitos

La gestión de cambios en los requisitos es esencial para preservar la validez del *Plan de Pruebas* del sistema **QuickContentMedia**. Cualquier modificación en los requisitos funcionales o no funcionales puede afectar la cobertura de los casos de prueba, los criterios de aceptación y el cronograma de ejecución; por ello, se establece el siguiente flujo de control:

- **Solicitud formal de cambio:** Toda propuesta de adición, eliminación o ajuste de un requisito debe presentarse por escrito utilizando el formato de la Tabla 3.3. La solicitud debe describir con claridad el cambio propuesto, su justificación y el posible impacto en las pruebas.
- **Evaluación del impacto:** El equipo de pruebas analiza la solicitud y determina:
 - Requisitos y casos de prueba afectados.
 - Necesidad de crear, modificar o retirar pruebas.
 - Repercusiones en cronograma y costos.
- **Aprobación:** El cambio se aprueba por consenso del equipo de proyecto y se actualizan la *Matriz de Trazabilidad de Requisitos*, el cronograma y el presupuesto, según corresponda.
- **Actualización de la documentación:** Una vez aprobado, se refleja el cambio en todos los documentos relevantes (requisitos, plan de pruebas, casos de prueba, cronograma y costos) y se notifica al equipo completo.

La Tabla 3.3 muestra el formato que se usará para registrar y hacer seguimiento a cada solicitud de cambio de requisitos.

Tabla 3.3 – Ejemplo de solicitud de cambio de requisitos

Campo	Descripción
Código	SCR-002
Fecha de solicitud	25/06/2025
Solicitado por	Camila Salazar
Requisito(s) afectado(s)	RNF-002 (Usabilidad)
Descripción del cambio	Añadir soporte para Tema Oscuro (Dark Mode) en todas las interfaces del sistema, accesible desde el menú de configuración de usuario.
Justificación	Dark Mode mejora la accesibilidad y la experiencia de usuario, y se alinea con las tendencias actuales de UI.
Impacto en pruebas	<ul style="list-style-type: none">■ Crear dos nuevos casos de prueba de validación para comprobar el cambio de tema.■ Incremento de 6 horas de esfuerzo (3 h senior tester , 3 h junior tester).
Estado	Pendiente
Fecha de resolución	—

Capítulo 4

Cronograma

Esta sección se enfocará en definir, secuenciar, estimar, desarrollar y controlar el tiempo necesario para ejecutar las pruebas del proyecto **QuickContentMedia** de manera eficiente y dentro de los plazos establecidos. Se definen las actividades relacionadas con los diferentes tipos de pruebas, incluyendo pruebas unitarias, de integración, de validación y de sistema, descomponiéndolas en tareas específicas y manejables. Se tiene en cuenta la secuenciación de las pruebas, identificando dependencias entre ellas y considerando los entregables parciales del desarrollo. Asimismo, se realiza la estimación del tiempo requerido para cada tipo de prueba en función del avance del proyecto y de los recursos asignados. Finalmente, se construye el cronograma de pruebas, analizando su duración, orden de ejecución y criterios de finalización, con el objetivo de garantizar la calidad y funcionamiento correcto del sistema.

4.1. Actividades

Las actividades se definen en la siguiente tabla, esta contiene el código, la relación con la actividad anterior, fechas, recursos y costo de cada actividad. En la tabla 4.1 se detallan las actividades para las pruebas del sistema **QuickContentMedia**.

Tabla 4.1 – Cronograma de actividades para las pruebas del sistema **QuickContentMedia**.

Código	Actividad	Tarea	Inicio	Final	Recursos	Costo (S/)
AP-01	Diseño de casos de prueba	—	19/05/25	24/05/25	Recursos humanos: Alexander Murillo, Alexis Espinoza, Santiago Salas.	5,400.00
AP-02	Pruebas unitarias	PU Módulo administrador	26/05/25	31/05/25	Recursos humanos: Alexis Espinoza, Santiago Salas. Recursos físicos: Pycharm	5,400.00
AP-03	Pruebas unitarias	PU Módulo cliente	02/06/25	07/06/25	Recursos humanos: Camila Salazar, Jean Condori. Recursos físicos: Pycharm	2,700.00
AP-04	Pruebas de integración	PI Módulo administrador	02/06/25	07/06/25	Recursos humanos: Alexis Espinoza, Camila Salazar. Recursos físicos: Pycharm	2,700.00
AP-05	Pruebas de integración	PI Módulo cliente	09/06/25	14/06/25	Recursos humanos: Alexis Espinoza, Camila Salazar. Recursos físicos: Pycharm	5,400.00
AP-06	Pruebas de validación	—	16/06/25	21/06/25	Recursos humanos: Camila Salazar, Jean Condori. Recursos físicos: Selenium Grid, Heroku.	2,973.75
AP-07	Pruebas de sistema	—	16/06/25	21/06/25	Recursos humanos: Alexis Espinoza. Recursos físicos: Selenium Grid, Heroku, JMeter Cloud.	3,521.75

El diagrama de Gantt correspondiente a las pruebas del sistema **QuickContentMedia** se presenta en la Figura 4.1.

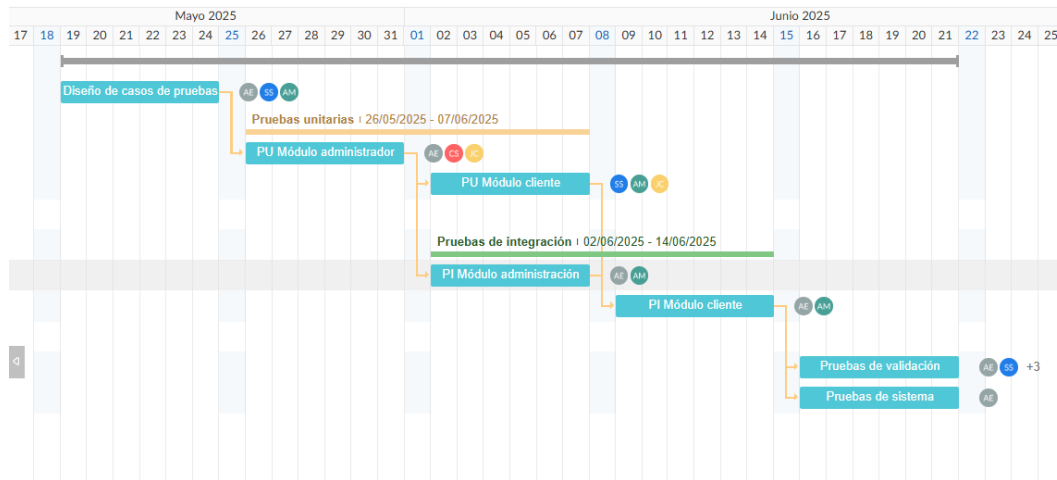


Figura 4.1 – Diagrama de Gantt para las pruebas del sistema **QuickContentMedia**.

Archivo: Diagrama de Gantt (Herramienta online GanttPro)

Link: [Abrir diagrama de Gantt](#)

4.2. Gestión de cronograma

La gestión del cronograma de pruebas es fundamental para asegurar que las actividades de verificación y validación del sistema **QuickContentMedia** se desarrollen de acuerdo al calendario establecido, permitiendo identificar a tiempo desviaciones que puedan afectar la calidad o el avance del software. Se realizarán revisiones periódicas del progreso de las pruebas programadas, evaluando el cumplimiento de los tiempos, la disponibilidad de recursos y los resultados obtenidos.

- **Monitoreo y Detección de Cambios en el Cronograma de Pruebas:** Se llevarán a cabo revisiones periódicas del avance de las pruebas en cada módulo, detectando retrasos, bloqueos o desviaciones respecto al cronograma.
- **Comparación y Evaluación del Cambio:** Se contrastan los resultados esperados y reales de ejecución de los casos de prueba. Si se detectan discrepancias o reprogramaciones necesarias, se documentan las causas (por ejemplo, errores en el software, problemas de entorno, indisponibilidad de testers, etc.) usando el formato de la tabla 4.2.
- **Acciones Correctivas:** Se tomarán medidas como reasignación de pruebas entre miembros del equipo o ajustes en el orden de ejecución para mitigar retrasos.
- **Acciones Preventivas:** Se reforzarán prácticas como realizar pruebas paralelas en módulos independientes con el fin de reducir futuros impactos.
- **Actualización del Cronograma de Pruebas:** Cualquier modificación deberá ser aprobada por todo el equipo. Una vez aprobada, se reflejará el cambio en el cronograma y se notificará a todos los integrantes.

La Tabla 4.2 muestra el formato que se utilizará para documentar las desviaciones detectadas durante la ejecución del plan de pruebas.

Tabla 4.2 – Ejemplo de registro de desviaciones detectadas durante la ejecución del plan de pruebas de **QuickContentMedia**.

Código	Fecha	Revisor(es)	Actividad Afectada	Causa de la Desviación	Acciones propuestas	Estado	Fecha de resolución
SCC-01	28/05/25	Alexis Espinoza	AP-02	Error en la ejecución de casos por cambios recientes	Ajustar scripts de prueba y extender 1 día la ejecución	En evaluación	-

Capítulo 5

Costos

5.1. Introducción

En este capítulo se presenta el desglose financiero completo para las actividades de verificación y validación del sistema **QuickContentMedia** (5 semanas). El documento establece:

- **Moneda:** Todos los valores están expresados en Soles (PEN) con redondeo a dos decimales
- **Exactitud:** Margen de variación del $\pm 8\%$ según lo establecido en la sección 5.1 del Plan de Trabajo
- **Herramientas principales:**
 - Pycharm (IDE)
 - Git (control de versiones)
 - Selenium WebDriver (automatización de pruebas UI)
 - JMeter (pruebas de carga y rendimiento)
 - Heroku (entorno de pruebas en la nube)
- **Umbral de control:** Variaciones superiores al 7% requerirán revisión inmediata

5.2. Metodología de Costeo

Este presupuesto se ha elaborado mediante un proceso estructurado en tres fases:

1. **Cálculo individual:** Determinación de costos unitarios por rol y recurso
2. **Generalización:** Proyección escalada para todo el equipo de trabajo
3. **Asignación temporal:** Distribución según el cronograma de actividades

5.3. Detalle Individual de Costos

5.3.1. Recursos Humanos (Costos Unitarios)

Esta sección detalla la inversión requerida por cada miembro del equipo durante las 5 semanas de ejecución.

Tabla 5.1 – Costo Individual por Rol

Rol	Horas/Día	Días	Tarifa (S/)	Total
Tester Senior	6	30	35	6,300.00
Tester Junior	8	30	20	4,800.00

Notas:

- Días laborales calculados: 30 días (5 semanas \times 6 días hábiles)
- Diferenciación horaria basada en complejidad de tareas asignadas
- Tarifas alineadas con el mercado local para perfiles QA

5.3.2. Infraestructura (Costos Unitarios)

Desglose de plataformas tecnológicas requeridas para la ejecución de pruebas.

Tabla 5.2 – Costos Unitarios de Plataformas

Recurso	Detalle Técnico	Costo (S/)	Justificación
Heroku Dyno	Plan Professional - 2.5GB RAM	182.50	Entorno estable para pruebas E2E
Selenium Grid	BrowserStack Basic Plan	182.50	Soporte para automatización cross-browser
JMeter Cloud	20 horas de ejecución	548.00	Paquete básico para pruebas de carga

5.4. Generalización para el Equipo

5.4.1. Personal (Equipo Completo)

Proyección escalada para los 5 miembros del equipo de QA.

Tabla 5.3 – Extrapolación de Costos de Personal

Rol	Cantidad	Costo Unitario	Total
Tester Senior	2	6,300.00	12,600.00
Tester Junior	3	4,800.00	14,400.00
Total	5	-	27,000.00

5.4.2. Infraestructura (Consolidado)

Inversión total en plataformas tecnológicas compartidas.

Tabla 5.4 – Costos Consolidados de Plataformas

Concepto	Total (S/)	Distribución Temporal
Heroku (2 Dynos)	365.00	Semanas 1-5 (uso continuo)
Selenium Grid	182.50	Semanas 2-4 (AP-02 a AP-05)
JMeter Cloud	548.00	Semanas 4-5 (AP-05, AP-07)
Total	1,095.50	-

5.5. Presupuesto Consolidado

Visión global de la inversión requerida para el plan de pruebas.

Tabla 5.5 – Resumen Final del Presupuesto

Concepto	Monto (S/)	Observaciones
Recursos Humanos	27,000.00	2 seniors + 3 juniors
Infraestructura	1,095.50	Heroku + Selenium + JMeter
Subtotal	28,095.50	-
Contingencias (10 %)	2,809.55	Fondo para riesgos no previstos
Total	30,905.05	-

5.6. Distribución por Actividad

Asignación detallada de costos según el cronograma del proyecto.

Tabla 5.6 – Costos Asociados a Cada Actividad (5 semanas)

Código	Actividad	Duración	Costo (S/)	Recursos
AP-01	Diseño de casos de prueba	5 días	5,400.00	2 seniors + 1 junior
AP-02	Pruebas unitarias módulo Admin	5 días	5,400.00	1 senior + 2 juniors
AP-03	Pruebas unitarias módulo Cliente	5 días	2,700.00	1 senior + 2 juniors
AP-04	Pruebas integración Admin	5 días	2,700.00	2 seniors
AP-05	Pruebas integración Cliente	5 días	5,400.00	2 seniors
AP-06	Validación final	5 días	2,973.75	Equipo completo + Heroku + Selenium
AP-07	Pruebas de sistema	5 días	3,521.75	1 Senior + Heroku + Selenium + JMeter
Total semanas: 5 (30 días hábiles)				

5.7. Gestión de costos

La gestión de cambios en los costos es fundamental para asegurar que las actividades de verificación y validación del sistema **QuickContentMedia** se desarrollen de acuerdo con el presupuesto establecido. Este control permite identificar tempranamente desviaciones que puedan comprometer la calidad, los plazos o la sostenibilidad financiera del proyecto.

Se realizarán revisiones periódicas de las actividades de prueba programadas, verificando si el uso de recursos humanos y tecnológicos se mantiene dentro de los márgenes previstos. Cualquier diferencia deberá ser documentada, analizada y, si corresponde, corregida mediante ajustes presupuestarios justificados.

- **Monitoreo y Detección de Cambios en los Costos:** Se llevarán a cabo revisiones continuas del uso de recursos durante cada fase de pruebas (unitarias, integración, validación y sistema). Estas evaluaciones buscan detectar incrementos imprevistos en las horas hombre, licencias de software o necesidades de infraestructura adicionales.
- **Comparación y Evaluación del Cambio:** Se contrastan los costos reales ejecutados frente a los estimados. Las discrepancias deben ser analizadas considerando factores como cambios en el alcance, problemas técnicos, disponibilidad de testers, entre otros.
- **Acciones Correctivas:** Se podrán tomar medidas como redistribución de tareas, sustitución de herramientas o modificación en el orden de ejecución, para evitar que se

generen sobrecostos en las etapas subsiguientes.

- **Acciones Preventivas:** Se reforzarán prácticas como la planificación cruzada de tareas entre miembros del equipo para evitar desviaciones futuras.
- **Actualización del Presupuesto:** Toda solicitud de modificación en los costos deberá contar con la aprobación del equipo. Una vez aceptada, el nuevo presupuesto será actualizado formalmente y comunicado a todos los involucrados.

Las solicitudes de modificaciones en los costos se realizarán usando el formato de la tabla 5.7.

Tabla 5.7 – Solicitud de modificación de costos en el plan de pruebas

Campo	Descripción
Código de solicitud	SCCO-01
Fecha de solicitud	12/06/2025
Solicitado por	Alexis Espinoza
Descripción del cambio	Se solicita la adquisición de 10 horas adicionales de ejecución en JMeter Cloud para ampliar las pruebas de carga del módulo cliente.
Justificación técnica o estratégica	Durante las pruebas se detectaron escenarios no cubiertos inicialmente que requieren evaluación de rendimiento con múltiples usuarios concurrentes.
Impacto en el presupuesto	Incremento de S/ 274.00 adicionales sobre el rubro de infraestructura. Aumento del total consolidado en un 1.06 %.
Estado del cambio	Aprobado
Fecha de resolución	13/06/2025

Capítulo 6

Recursos

Esta sección detalla los recursos necesarios para la ejecución del plan de pruebas del sistema **QuickContentMedia**. Se garantiza que tanto el equipo humano como los recursos tecnológicos y materiales estén disponibles y sean utilizados de manera óptima durante la verificación y validación del sistema. Para llevar a cabo las actividades de prueba de **QuickContentMedia**, se asegura que se cuente con el personal adecuado, así como con el equipamiento y las herramientas tecnológicas necesarias para la ejecución.

6.1. Recursos humanos

Para la ejecución del plan de pruebas del sistema **QuickContentMedia**, se asignaron roles específicos a los miembros del equipo con el objetivo de asegurar la cobertura efectiva de todas las actividades planificadas. La tabla 6.1 detalla los integrantes del equipo, el rol asumido y una breve descripción de sus responsabilidades:

Tabla 6.1 – Asignación de roles según el plan de pruebas del sistema **QuickContentMedia**

Nombre	Rol en las pruebas	Responsabilidades principales
Alexander Murillo	Tester Senior / Coordinador de pruebas	Diseñar los casos de prueba, coordinar actividades del equipo, controlar el cronograma, validar entregables, ejecutar pruebas de integración, validación.
Alexis Espinoza	Tester Senior	Ejecutar y supervisar pruebas unitarias, pruebas de integración, y apoyar en pruebas de validación y sistema.
Camila Salazar	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados.
Jean Condori	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados.
Santiago Salas	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados, apoyar en diseño de casos de prueba.

6.2. Recursos físicos

- **Equipos informáticos:** Cinco computadoras de escritorio o portátiles con características suficientes para soportar la ejecución de pruebas funcionales, de integración, validación y sistema:
 - **Procesador:** Intel Core i5 o AMD Ryzen 5.
 - **Memoria RAM:** Mínimo 8 GB.
 - **Almacenamiento:** Disco SSD de al menos 256 GB para ejecución ágil de entornos de pruebas y herramientas de testing.
 - **Tarjeta Gráfica:** Integrada o dedicada, suficiente para pruebas de interfaz gráfica y ejecución de simulaciones ligeras.
 - **Sistema Operativo:** Windows 10/11, macOS o una distribución Linux compatible.
- **Infraestructura Tecnológica:**
 - **Plataformas de prueba:** *Heroku Dyno* para entornos de prueba E2E, *Selenium*

Grid (BrowserStack) para pruebas automatizadas cross-browser, y *JMeter Cloud* para pruebas de carga.

- **Control de versiones:** GitHub (versión Free), para el control y trazabilidad del avance en scripts y configuraciones de prueba.
- **Editor de texto/IDE:** Pycharm versión 2025.1 para revisar logs, configurar scripts de prueba automatizados y analizar resultados.
- **Herramientas complementarias:** Google Docs y Overleaf para registrar hallazgos e incidencias; Figma y VisualParadigm para validar consistencia visual y funcional en interfaces.
- **Base de Datos:** PostgreSQL (versión 17.4) para pruebas relacionadas con integridad de datos y validaciones de back-end.

6.3. Gestión de recursos

La gestión de recursos en el plan de pruebas del proyecto **QuickContentMedia** es esencial para asegurar que las actividades de validación, verificación y ejecución de pruebas cuenten con los recursos humanos y tecnológicos necesarios en el momento adecuado. Ante cualquier solicitud de modificación en los recursos, se activa un proceso estructurado que inicia con la evaluación del impacto del cambio sobre las asignaciones de pruebas, y culmina con la actualización de los documentos correspondientes. Este proceso busca garantizar que las decisiones estén alineadas con los objetivos de calidad del software y no afecten negativamente los tiempos del cronograma de pruebas.

Solicitud Formal de Cambio: Toda solicitud de modificación al plan de recursos de pruebas debe ser presentada de forma formal usando el formato de la tabla 6.2. Esta debe incluir una descripción del cambio propuesto, su justificación técnica o estratégica, y una estimación del impacto que tendría en el cronograma y los costos del proceso de pruebas.

Evaluación de la Solicitud de Cambio: Se analiza el impacto de la solicitud en la asignación de testers, la disponibilidad de plataformas, herramientas y su efecto sobre el cronograma de ejecución de pruebas. Se evalúan riesgos como retrasos, sobrecarga de tareas o conflictos de planificación. Para ello, se utiliza el cronograma de pruebas y el diagrama de Gantt como herramientas de análisis.

Aprobación y Actualización del Cambio: Si se justifica el cambio, este debe ser aprobado por el equipo completo. Posteriormente, se actualizan los documentos del plan de pruebas, asignaciones de roles y cronograma correspondiente. De ser necesario, también se redistribuyen responsabilidades o se modifican los entornos y herramientas requeridas.

Actualización del Cronograma y Registro de Cambios: Cualquier modificación aprobada se reflejará en el cronograma de pruebas, detallando los nuevos recursos asignados y fechas reprogramadas. Estos cambios serán comunicados a todo el equipo y se documentarán formalmente para su trazabilidad.

La Tabla 6.2 muestra un ejemplo completo de una solicitud de cambio aplicada durante el proceso de pruebas del sistema **QuickContentMedia**. Este tipo de formato permite documentar de manera clara y estructurada cualquier modificación que pueda impactar en los recursos y costos del proyecto.

Tabla 6.2 – Ejemplo de solicitud de cambio de recursos en el plan de pruebas

Campo	Descripción
Código	SCR-001
Fecha de solicitud	10/06/2025
Solicitado por	Alexis Espinoza
Descripción del cambio	Se solicita reasignar a Jean Condori (Tester Junior) desde la tarea AP-06 (Validación) hacia AP-05 (Pruebas de integración módulo cliente), debido a mayor carga de pruebas en esta última actividad.
Justificación técnica o estratégica	Durante la ejecución de AP-05 se identificaron más escenarios de integración de lo previsto, lo cual exige mayor esfuerzo en ejecución y documentación.
Impacto en cronograma	Se anticipa reducción de 1 día en la finalización de AP-05. No se afecta el inicio de AP-06.
Impacto en costos	No se genera costo adicional, se mantiene la misma distribución de horas hombre.
Estado del cambio	Aprobado
Fecha de resolución	11/06/2025

Capítulo 7

Diseño de casos de prueba

Esta sección presenta el diseño de los casos de prueba desarrollados para verificar y validar las funcionalidades del sistema **QuickContentMedia**. El diseño de pruebas tiene como objetivo asegurar que cada componente del sistema cumple con los requerimientos establecidos, anticipando posibles fallos y garantizando un comportamiento esperado bajo diferentes condiciones. Para ello, se han diseñado pruebas unitarias, de integración, de verificación y de sistema, cubriendo así distintos niveles de prueba.

7.1. Pruebas unitarias

Las pruebas unitarias validan el funcionamiento de componentes individuales del sistema. Los datos de prueba son genéricos y cumplen con las reglas de negocio definidas.

Tabla 7.1 – Diseño de caso de prueba unitario PU-001: Validación del método `agregarAlCarrito`

Código	PU-001
Nombre del caso	Validación de agregar contenido al carrito de compras
Técnica de prueba	Caja blanca · Prueba funcional
Componente	<code>gestorCarrito</code> (G-006)
Método bajo prueba	<code>agregarAlCarrito(idCliente: int, idContenido: int)</code>
Precondiciones	<ul style="list-style-type: none"> ■ <code>idCliente</code> existe en la tabla Cientes (BD-001) y es un entero positivo. ■ <code>idContenido</code> existe en la tabla Contenidos (BD-008) y es un entero positivo. ■ El contenido no está duplicado en el carrito del cliente.
Datos de prueba	<ul style="list-style-type: none"> ■ <code>idCliente</code>: Entero positivo registrado en BD-001. ■ <code>idContenido</code>: Entero positivo registrado en BD-008.
Ejecución	<ol style="list-style-type: none"> 1. Invocar el método con los parámetros válidos. 2. Verificar la inserción en Carrito_compras (BD-003).
Resultado esperado	<ul style="list-style-type: none"> ■ El método retorna <code>true</code>. ■ El registro se inserta en BD-003 con los campos: <ul style="list-style-type: none"> • <code>idCliente</code>: Valor proporcionado. • <code>idContenido</code>: Valor proporcionado. • <code>descuentoAplicado</code>: <code>false</code> (valor por defecto).
Criterio de éxito	<ul style="list-style-type: none"> ■ El método valida que los IDs existan y no permita duplicados. ■ Cumple con las reglas de negocio (RF-005).

Tabla 7.2 – Diseño de caso de prueba unitario PU-008: Validación del método actualizarSaldoCliente

Código	PU-002
Nombre del caso	Validación de actualización de saldo de cliente
Técnica de prueba	Caja blanca · Prueba funcional
Componente	gestorUsuario (G-001)
Método bajo prueba	actualizarSaldoCliente(idCliente: int, nuevoSaldo: float)
Precondiciones	<ul style="list-style-type: none"> ■ idCliente existe en la tabla Clientes (BD-001). ■ nuevoSaldo es un valor numérico válido.
Datos de prueba	<ul style="list-style-type: none"> ■ idCliente: Valor entero positivo registrado. ■ nuevoSaldo: Valor numérico dentro del rango permitido.
Ejecución	<ol style="list-style-type: none"> 1. Invocar el método con parámetros válidos. 2. Verificar la actualización en Clientes (BD-001).
Resultado esperado	<ul style="list-style-type: none"> ■ El método retorna true. ■ El campo saldo se actualiza en BD-001. ■ Solo se permiten valores numéricos positivos. ■ Se rechazan valores que excedan el límite máximo.
Criterio de éxito	<ul style="list-style-type: none"> ■ El método valida que el cliente exista. ■ Acepta solo valores numéricos dentro del rango permitido. ■ Cumple con las reglas de negocio (RF-008).

7.2. Pruebas de integración

Las pruebas de integración validan la interacción entre múltiples componentes del sistema. A continuación, se describe el caso de prueba para el flujo de compra con descuento.

Tabla 7.3 – Diseño de caso de prueba de integración PI-001: Flujo de compra con descuento aplicado

Código	PI-001
Nombre del caso	Integración entre gestión de carrito, promociones y pagos durante una compra con descuento
Técnica de prueba	Caja blanca · Prueba funcional
Actor	Cliente (ACT-001)
Componentes integrados	<ul style="list-style-type: none"> ■ gestorCarrito (G-006): Agrega contenido al carrito. ■ gestorPromocion (G-008): Aplica descuentos. ■ gestorPago (G-004): Procesa el pago. ■ UIInicioCliente (MK-003): Interfaz de inicio. ■ UIResumenPedido (MK-045): Interfaz de confirmación.
Precondiciones	<ul style="list-style-type: none"> ■ Cliente autenticado en UIInicioCliente (MK-003). ■ Contenido en promoción registrado en Promociones (BD-007). ■ Saldo suficiente en Cientes (BD-001).
Pasos de la prueba	<ol style="list-style-type: none"> 1. Cliente agrega contenido en promoción al carrito (método agregarAlCarrito de G-006). 2. Sistema verifica descuento aplicable (método validarPromociónActiva de G-008). 3. Cliente confirma compra en UIResumenPedido (MK-045). 4. Sistema actualiza saldo (método procesarPago de G-004) y registra la compra en Compras (BD-004).
Resultado esperado	<ul style="list-style-type: none"> ■ Descuento aplicado correctamente (precio final reducido). ■ Saldo actualizado en BD-001. ■ Compra registrada en BD-004 con estado exitosa. ■ Interfaz MK-045 muestra resumen con descuento aplicado.
Criterio de éxito	<ul style="list-style-type: none"> ■ Todos los componentes interactúan sin errores. ■ Los datos persisten consistentemente en las tablas BD-001, BD-003, BD-004 y BD-007. ■ Cumple con las reglas de negocio (RF-005 y RF-007).

Tabla 7.4 – Diseño de caso de prueba PI-002: Integración para ver el historial del cliente

Campo	Descripción
Código	PI-002
Nombre del caso	Integración entre la interfaz de administración de clientes y gestión de contenido al visualizar el historial de descargas de un cliente.
Técnica de prueba	Caja blanca · Prueba funcional
Componentes integrados	<ul style="list-style-type: none"> ■ <code>UIAdministrarCliente</code> (MK-038) ■ <code>gestorContenido</code> (G-005)
Precondiciones	<ul style="list-style-type: none"> ■ El administrador está autenticado y en la sección <i>Administrar Clientes</i>. ■ Existen registros de descargas en BD-005 para el <code>idCliente</code> seleccionado.
Pasos de la prueba	<ol style="list-style-type: none"> 1. El administrador pulsa <i>Ver historial</i> sobre el cliente. 2. <code>UIAdministrarCliente</code> invoca <code>obtenerContenidosDescargados(idCliente)</code> en <code>gestorContenido</code>. 3. <code>gestorContenido</code> llama a <code>obtenerDescargas(idCliente)</code> en BD-005 Descargas y recibe <code>listaDescargas</code>. 4. Con los identificadores devueltos, <code>gestorContenido</code> ejecuta <code>obtenerContenidos(listaDescargas)</code> en BD-008 Contenidos y recibe <code>listaContenidos</code>. 5. <code>gestorContenido</code> devuelve <code>listaContenidos</code> a <code>UIAdministrarCliente</code>. 6. <code>UIAdministrarCliente</code> muestra los resultados en orden cronológico inverso (<code>UIHistorialCliente</code>) mediante el método <code>verHistorialCliente()</code>.
Resultado esperado	<ul style="list-style-type: none"> ■ <code>UIAdministrarCliente</code> presenta cada contenido (nombre, autor, formato, categoría y fecha de descarga). ■ Las llamadas <code>obtenerDescargas</code> y <code>obtenerContenidos</code> se completan sin errores.
Criterio de éxito	<ul style="list-style-type: none"> ■ Todos los componentes interactúan sin errores. ■ Los datos mostrados coinciden con la unión lógica entre BD-005 y BD-008. ■ El administrador puede regresar a <code>UIAdministrarCliente</code> sin fallos ni pérdida de información.

7.3. Pruebas de validación

Las pruebas de validación tienen como propósito asegurar que los componentes del sistema **QuickContentMedia** satisfacen los requisitos funcionales establecidos, evaluándolos desde la perspectiva del usuario final. Estas pruebas permiten confirmar que el sistema responde correctamente ante diversas entradas y escenarios de uso. A continuación, se presentan los diseños de los casos de prueba de validación definidos para el sistema **QuickContentMedia**.

Tabla 7.5 – Diseño de caso de prueba PV-001: Validación del ingreso de saldo a un cliente

Campo	Descripción
Código	PV-001
Nombre del caso	Validación del ingreso de saldo a un cliente desde el panel del administrador.
Técnica de prueba	Caja negra
Actor	Administrador del sistema
Precondiciones	<ul style="list-style-type: none"> ■ El administrador está autenticado en el sistema. ■ Se encuentra visualizando la interfaz UIAdministrar-Cliente (MK-038). ■ La tabla de clientes está cargada correctamente.
Pasos de la prueba	<ol style="list-style-type: none"> 1. Clic en el botón <i>Gestionar saldo</i> de un cliente. 2. En la ventana emergente UIGestionarSaldo (MK-039), ingresa un valor numérico entero no negativo. 3. Hace clic en el botón <i>Confirmar</i>.
Resultado esperado	<ul style="list-style-type: none"> ■ El sistema acepta el valor ingresado si es válido. ■ Se actualiza el campo de saldo correspondiente en la tabla de clientes.
Criterio de éxito	El sistema solo permite continuar si el valor ingresado cumple con el formato requerido (entero no negativo), y se actualiza visualmente.

Tabla 7.6 – Diseño de caso de prueba PV-002: Validación del destinatario en el flujo de envío de regalo

Campo	Descripción
Código	PV-002
Nombre del caso	Validación del ingreso y verificación del usuario destinatario antes del envío de regalo.
Técnica de prueba	Caja negra
Actor	Cliente
Precondiciones	<ul style="list-style-type: none"> ■ El cliente está autenticado en el sistema. ■ Tiene al menos un contenido en su carrito. ■ Se encuentra visualizando la interfaz UICarrito (MK-015).
Pasos de la prueba	<ol style="list-style-type: none"> 1. El cliente hace clic en el botón <i>Regalar</i>. 2. En la ventana emergente UIDestinatario (MK-016), ingresa un valor alfanumérico correspondiente al username del destinatario. 3. Hace clic en <i>Continuar</i>.
Resultado esperado	<ul style="list-style-type: none"> ■ Si el username ingresado está asociado a una cuenta activa, el sistema permite avanzar a la interfaz UIResumenPedido (MK-045). ■ Si el username ingresado no existe o la cuenta está inactiva, se muestra la ventana UIErrorDestinatario (MK-046) con un mensaje informativo y opción de volver.
Criterio de éxito	El sistema distingue correctamente entre usernames válidos e inválidos y responde adecuadamente con la interfaz correspondiente.

7.4. Pruebas de sistema

Las pruebas de sistema tienen como propósito evaluar el comportamiento del sistema **QuickContentMedia** como una unidad integrada, verificando que sus componentes colaboran correctamente bajo diferentes condiciones y cumplen los requisitos especificados. Estas pruebas abarcan funcionalidades completas, así como características no funcionales como seguridad, compatibilidad y rendimiento. A continuación, se presentan los diseños de casos de prueba de sistema definidos.

Tabla 7.7 – Diseño de caso de prueba PS-001: Verificación de compatibilidad en navegadores

Campo	Descripción
Código	PS-001
Nombre del caso	Verificación de compatibilidad de interfaz y funcionalidad en múltiples navegadores.
Técnica de prueba	Caja negra · Prueba de compatibilidad cruzada (Cross-browser testing).
Actor	Cliente (usuario final)
Precondiciones	<ul style="list-style-type: none"> ■ El sistema está desplegado en un entorno de pruebas. ■ Se dispone de navegadores actualizados: Chrome, Firefox, Edge, Safari y Opera. ■ El usuario cuenta con una cuenta activa.
Pasos de la prueba	<ol style="list-style-type: none"> 1. Abrir el sistema en cada navegador. 2. Navegar por las secciones: inicio, contenidos, carrito y formularios. 3. Realizar acciones como enviar formularios y hacer clic en botones y enlaces.
Resultado esperado	<ul style="list-style-type: none"> ■ La interfaz se renderiza correctamente y sin errores visuales. ■ Las acciones funcionales se comportan igual en todos los navegadores. ■ No se producen errores de ejecución ni comportamientos inesperados.
Criterio de éxito	El sistema debe conservar la coherencia visual y funcional en todos los navegadores especificados.

Tabla 7.8 – Diseño de caso de prueba PS-002: Verificación de control de acceso basado en roles

Campo	Descripción
Código	PS-002
Nombre del caso	Verificación de restricciones de acceso a funciones administrativas desde cuentas no autorizadas.
Técnica de prueba	Caja negra · Basada en roles (RBAC).
Actor	Cliente (usuario con rol restringido)
Precondiciones	<ul style="list-style-type: none"> ■ El cliente ha iniciado sesión en el sistema. ■ Existen rutas protegidas implementadas: <code>/admin</code>, <code>/usuarios</code>, <code>/promociones</code>. ■ Se cuenta con roles diferenciados (cliente y administrador).
Pasos de la prueba	<ol style="list-style-type: none"> 1. Ingresar manualmente a la URL de un módulo restringido (ej. <code>/admin</code>). 2. Intentar ejecutar funciones restringidas como ver usuarios o crear promociones. 3. Observar la respuesta del sistema frente al intento.
Resultado esperado	<ul style="list-style-type: none"> ■ El sistema bloquea el acceso no autorizado. ■ Se muestra un mensaje claro o se redirige automáticamente. ■ No se filtra información ni se permite ejecución de funciones restringidas.
Criterio de éxito	El sistema protege efectivamente las rutas y funciones administrativas, impidiendo accesos no autorizados.

Referencias

- [1] «OMG Unified Modeling Language (OMG UML), Version 2.5.1,» Object Management Group, inf. téc., 2017, Disponible en línea. dirección: <https://www.omg.org/spec/UML/>.
- [2] Overleaf. «Overleaf, Online LaTeX Editor.» Accedido el 10 de abril de 2025. (2024), dirección: <https://www.overleaf.com>.
- [3] Visual Paradigm International Ltd. «Visual Paradigm – Visual Modeling and Agile Development Tools.» Accedido el 10 de abril de 2025. (2024), dirección: <https://www.visual-paradigm.com>.
- [4] Figma Inc. «Figma – Collaborative Interface Design Tool.» Accedido el 13 de abril de 2025. (2024), dirección: <https://www.figma.com>.