**Planes de Pruebas**

**1.Autentificar Usuario al iniciar Sesión**

**Introducción**  
El plan de prueba está diseñado como una línea base para validar la funcionalidad de autenticación de usuarios al iniciar sesión en el sistema. Este documento identifica el alcance, los recursos, los riesgos y las suposiciones asociadas a esta prueba.

**Recursos**

| **Tester** | **% Participación** |
| --- | --- |
| Tester | 100% |
| Tester | 50% |

**Alcance**  
Las pruebas incluyen:

* Validar la autenticación de usuarios mediante RUT y clave alfanumérica.
* Verificar que el sistema redirija al usuario según su perfil (ej. administrador, usuario estándar).
* Comprobar mensajes de error para credenciales incorrectas o campos vacíos.
* Pruebas de regresión para asegurar que la funcionalidad no afecte otras áreas del sistema.

**Fuera del Alcance**

* Pruebas de recuperación de contraseña.
* Pruebas de autenticación con métodos alternativos (ej. redes sociales, huella digital).

**Pruebas de Rendimiento**

* Evaluar el tiempo de respuesta del sistema durante la autenticación bajo carga simulada.
* Verificar la escalabilidad del sistema cuando múltiples usuarios inician sesión simultáneamente.

**Pruebas de Aceptación (UAT)**

* Los usuarios finales validarán que el proceso de autenticación sea intuitivo y funcione según lo esperado.
* Se seleccionarán usuarios con distintos niveles de experiencia y perfiles para garantizar una evaluación integral.

**Infraestructura**

* El entorno de pruebas debe contar con bases de datos de usuarios realistas y servidores configurados para simular el entorno de producción.

**Suposiciones**

* Las credenciales de prueba (RUT y claves) han sido previamente generadas y son válidas.
* El sistema de autenticación está integrado con la base de datos de usuarios.

**Riesgos**

| **No.** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad (Prob\*Impacto)** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| 1 | Fallos en la validación de credenciales | 3 | 4 | 12 | Revisar y ajustar las reglas de validación. |
| 2 | Lentitud en el proceso de autenticación | 2 | 3 | 6 | Optimizar consultas a la base de datos y mejorar la infraestructura. |
| 3 | Usuarios no pueden acceder según su perfil | 2 | 5 | 10 | Verificar los permisos asociados a cada perfil y realizar pruebas exhaustivas. |

**2.** **Crear Cuentas de Usuario**

**Introducción**  
Este plan de pruebas está diseñado para validar la funcionalidad de creación de cuentas de usuario por parte del Administrador. El objetivo es asegurar que el sistema permita la creación correcta de cuentas para Recepcionistas y Clientes Premium, asignando los atributos y permisos adecuados según su perfil.

**Recursos**

| **Tester** | **% Participación** |
| --- | --- |
| Tester 1 | 100% |
| Tester 2 | 50% |

**Alcance**

Las pruebas incluyen:

* **Creación de cuentas**: Validar que el Administrador pueda crear cuentas para Recepcionistas y Clientes Premium.
* **Atributos de usuario**: Verificar que los campos obligatorios (ej. nombre, RUT, correo, contraseña, perfil) se guarden correctamente.
* **Permisos**: Confirmar que los Recepcionistas y Clientes Premium tengan acceso solo a las funcionalidades asignadas a su perfil.
* **Validación de datos**: Comprobar mensajes de error para campos inválidos (ej. RUT mal formateado, correo duplicado).
* **Notificaciones**: Asegurar que el usuario reciba una confirmación (ej. correo electrónico) al crear su cuenta.

**Fuera del Alcance**

* Restablecimiento de contraseñas.
* Edición o eliminación de cuentas (se cubrirá en otro plan de pruebas).

**Pruebas de Rendimiento**

* Evaluar el tiempo de respuesta al crear múltiples cuentas simultáneamente.
* Verificar que el sistema maneje correctamente altas cargas de solicitudes de creación de cuentas.

**Pruebas de Aceptación (UAT)**

* Los usuarios finales (Administradores) validarán que el proceso sea intuitivo y funcione según lo esperado.
* Se probará con distintos perfiles (Recepcionistas y Clientes Premium) para garantizar que los permisos se asignen correctamente.

**Infraestructura**

* Entorno de pruebas con base de datos configurada para almacenar nuevos usuarios.
* Servidor de correo para pruebas de notificaciones.

**Suposiciones**

* El Administrador tiene los permisos necesarios para crear cuentas.
* Los perfiles (Recepcionista/Cliente Premium) están predefinidos en el sistema.

**Riesgos**

| **No.** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad (Prob\*Impacto)** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| 1 | Campos obligatorios no validados | 3 | 4 | 12 | Revisar reglas de validación en frontend y backend. |
| 2 | Permisos incorrectos asignados | 2 | 5 | 10 | Verificar la lógica de asignación de roles con casos de prueba específicos. |
| 3 | Fallos en el envío de notificaciones | 2 | 3 | 6 | Configurar logs para monitorear el servicio de correo. |

**3. Verificar disponibilidad de habitaciones**

**Introducción**  
Este plan de pruebas está diseñado para validar la funcionalidad de consulta de habitaciones disponibles en el sistema. El objetivo es garantizar que los usuarios puedan realizar búsquedas precisas y obtener resultados correctos según los criterios de filtrado (fechas, tipo de habitación, capacidad, etc.).

**Recursos**

| **Tester** | **% Participación** |
| --- | --- |
| Tester 1 | 100% |
| Tester 2 | 50% |

**Alcance**

**Pruebas a realizar:**

1. **Búsqueda básica:**
   * Validar que el sistema muestre habitaciones disponibles al ingresar fechas de entrada y salida.
   * Verificar que se apliquen filtros predeterminados (ej. habitaciones no reservadas, no en mantenimiento).
2. **Filtros avanzados:**
   * Tipo de habitación (individual, doble, suite).
   * Capacidad (número de personas).
   * Precio (rango mínimo y máximo).
   * Amenidades (Wi-Fi, TV, aire acondicionado).
3. **Validación de datos:**
   * Comprobar que el sistema no muestre habitaciones reservadas o en mantenimiento.
   * Mensajes claros cuando no hay disponibilidad.
4. **Rendimiento:**
   * Tiempo de respuesta con distintos volúmenes de datos (ej. alta demanda en temporada).

**Fuera del Alcance:**

* Proceso de reserva (se cubrirá en otro plan).
* Modificación de disponibilidad (ej. mantenimiento programado).

**Pruebas de Rendimiento**

* Simular múltiples consultas simultáneas para evaluar escalabilidad.
* Medir tiempos de respuesta con base de datos cargada (ej. 1000+ habitaciones).

**Pruebas de Aceptación (UAT)**

* Usuarios finales (Recepcionistas/Clientes) validarán:
  + Facilidad de uso de la interfaz.
  + Precisión de los resultados.

**Infraestructura**

* Entorno de pruebas con base de datos realista (ej. habitaciones, reservas, calendarios).
* Servidores configurados para simular cargas altas.

**Suposiciones**

* La información de reservas y mantenimiento está actualizada.
* Los filtros están correctamente configurados en el sistema.

**Riesgos**

| **No.** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad (Prob\*Impacto)** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| 1 | Resultados incorrectos por fechas | 3 | 4 | 12 | Verificar sincronización con el calendario de reservas. |
| 2 | Filtros no aplicados correctamente | 2 | 4 | 8 | Revisar lógica de filtrado en backend. |
| 3 | Lentitud en respuesta con alta demanda | 2 | 3 | 6 | Optimizar consultas SQL y añadir índices |

**4. Sincronización de inventario**

**Introducción**

Este plan verifica la funcionalidad de actualización automática del inventario al realizar transacciones de venta, compra o devolución de productos. Garantiza que el sistema mantenga consistencia en tiempo real entre las operaciones y los registros de stock.

**Recursos**

| **Rol** |  | **% Participación** | | | |
| --- | --- | --- | --- | --- | --- |
| Líder QA | 100% | | | |
| Tester Senior | 80% | | |
| Soporte TI | 30% | |

**Alcance**

**Funcionalidades a probar:**

* **Actualización por ventas**:
  + Disminución automática de stock al confirmar ventas
  + Validación de stock mínimo antes de permitir ventas
* **Actualización por compras**:
  + Incremento automático al registrar nuevas compras
  + Ajuste correcto de lotes y fechas de caducidad
* **Procesos de devolución**:
  + Reintegro de productos al inventario
  + Actualización de estados (ej: "Dañado", "Reutilizable")
* **Escenarios complejos**:
  + Transacciones simultáneas sobre el mismo producto
  + Sincronización entre múltiples sucursales

**Exclusiones**:

* Generación de reportes históricos
* Integración con proveedores externos

**Tipos de Pruebas**

| **Tipo** | **Objetivo** | **Herramientas** |
| --- | --- | --- |
| Funcionales | Validar reglas de negocio | Postman, JIRA |
| Estrés | Evaluar concurrencia | JMeter |
| Regresión | Verificar integridad tras updates | Selenium |

**Entorno de Pruebas**

* **Base de datos**: SQL Server 2022 con dataset de 50,000 productos
* **Servidores**: 3 nodos en cluster AWS
* **Conectores**: SAP ECC para transacciones financieras

**Criterios de Aceptación**

* Latencia máxima de actualización: 3 segundos
* Precisión del 100% en cálculos de stock
* 0% de transacciones huérfanas

**Matriz de Riesgos**

| **ID** | **Riesgo** | **Prob.** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- | --- |
| INV-01 | Race conditions en transacciones | 4 | 5 | Implementar bloqueos distribuidos |
| INV-02 | Pérdida de conexión durante updates | 3 | 4 | Mecanismo de colas persistentes |
| INV-03 | Inconsistencias en inventario físico vs digital | 2 | 5 | Integrar sistema de auditoría RFID |

**Cronograma**

| **Fase** | **Duración** | **Entregables** |
| --- | --- | --- |
| Preparación | 3 días | Casos de prueba |
| Ejecución | 5 días | Reporte de bugs |
| Validación | 2 días | Certificación UAT |

**Métricas Clave**

* Tasa de error: < 0.1%
* Rendimiento: 150 transacciones/segundo
* Cobertura: 95% de SKUs críticos

**5.** **Alertas de stock**

**Introducción**

Verificar el sistema de notificaciones automáticas para reposición de inventario cuando los productos alcanzan niveles mínimos predefinidos, garantizando:

* Exactitud en los cálculos de stock
* Oportunidad de las alertas
* Canales de notificación efectivos

**Recursos**

| **Rol** |  |
| --- | --- |
| QA |  |  |
| Tester |  |  |
| Product Owner |  |  |

**Alcance de Pruebas**

**Flujo Principal:**

1. Configuración de umbrales por producto/categoría
2. Detección automática de stock crítico
3. Generación y envío de alertas

**Casos a Validar:**

* Alertas por producto individual (ej.: "Stock crítico: Jeans M - 3 unidades")
* Notificaciones agrupadas por categoría (ej.: "Ropa - 5 productos bajo mínimo")
* Multi-canal (email, SMS, dashboard, app móvil)
* Escalado de alertas (ej.: notificar a jefe de almacén si no se repone en 24h)

**Exclusiones:**

* Proceso físico de reposición
* Integración con proveedores externos

**Criterios de Éxito**

| **Métrica** | **Requisito** |
| --- | --- |
| Tiempo de alerta | < 5 min tras alcanzar umbral |
| Precisión | 100% coincidencia stock real vs notificado |
| Cobertura | 100% de productos con umbral definido |

**Tipos de Pruebas**

**Pruebas Funcionales**

* Verificar cálculos con:

**Pruebas de Rendimiento**

* Simular 1000+ productos alcanzando umbral simultáneamente

**Pruebas de Regresión**

* Impacto tras actualizaciones de módulo de inventario

**Entorno de Pruebas**

| **Componente** | **Especificación** |
| --- | --- |
| Base de Datos | MongoDB con 10K productos de prueba |
| Servidor | AWS EC2 t3.xlarge |
| Notificaciones | Mailtrap (QA), Twilio (Prod) |

**Cronograma**

| **Fase** | **Duración** |
| --- | --- |
| Diseño casos | 2 días |
| Ejecución | 3 días |
| Reporte | 1 día |

**Motor de recomendaciones**

**Introducción**  
Este plan valida el sistema de sugerencias automáticas basadas en el historial de compras de los usuarios. Garantiza que las recomendaciones sean relevantes, personalizadas y se actualicen dinámicamente según el comportamiento del cliente.

**Recursos**

| **Rol** | **% Participación** |
| --- | --- |
| Tester Principal | 100% |
| Analista de Datos | 75% |
| UX Tester | 50% |

**Alcance**

**Pruebas a realizar:**

* **Precisión de recomendaciones**:
  + Verificar que los productos sugeridos coincidan con el perfil de compra histórico
  + Validar ponderación correcta de categorías (ej.: si usuario compra 70% electrónica, priorizar ese rubro)
* **Lógica de actualización**:
  + Comprobar que nuevas compras modifiquen las sugerencias en <24h
  + Pruebas con usuarios sin historial (debe mostrar recomendaciones genéricas)
* **Interfaz**:
  + Ubicación y diseño de sección "Recomendados"
  + Personalización por dispositivo (web/móvil)

**Exclusiones**:

* Integración con sistemas de marketing externos
* Generación de reportes analíticos

**Pruebas de Rendimiento**

| **Escenario** | **Métrica Objetivo** |
| --- | --- |
| 1000 usuarios simultáneos | <2s de respuesta |
| Base con 1M+ registros históricos | Precisión >85% |

**Pruebas de Aceptación (UAT)**

**Participantes:**

* 10 clientes frecuentes
* 5 nuevos usuarios

**Criterios:**

* 80% de satisfacción en relevancia de sugerencias
* 0% de recomendaciones incoherentes (ej.: carnearía para usuario vegano)

**Infraestructura**

| **Componente** | **Especificación** |
| --- | --- |
| Entorno | AWS SageMaker (modelo ML) + Redis (caché) |
| Datos | Dataset anonimizado de 50K transacciones reales |

**Suposiciones**

* El historial de compras está correctamente etiquetado por categorías
* Los algoritmos de ML fueron entrenados previamente

**Riesgos**

| **ID** | **Riesgo** | **Prob.** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- | --- |
| MR-01 | Sesgo en recomendaciones | 3 | 4 | Auditoría mensual de algoritmos |
| MR-02 | Latencia en actualizaciones | 2 | 3 | Optimizar jobs nocturnos |
| MR-03 | Filtros ineficientes | 2 | 5 | Implementar sistema de feedb |

**6.** **Click & Collect**

**Introducción**  
Este plan verifica el sistema de reserva en tienda con confirmación inmediata, garantizando que los clientes puedan:

1. Reservar productos online
2. Recibir confirmación en tiempo real
3. Retirar físicamente en el local seleccionado

**Recursos**

| **Rol** | **Responsabilidad** | **% Participación** |
| --- | --- | --- |
| QA Lead | Coordinación general | 100% |
| Tester Comercio | Pruebas en POS | 80% |
| Tester Web | Pruebas frontend | 70% |

**Alcance**

**Flujo completo a validar:**

1. **Reserva Online**
   * Disponibilidad real-time por sucursal
   * Límite de stock reservable
   * Selección de franja horaria para retiro
2. **Confirmación Automática**
   * Notificación multicanal (email/SMS/app)
   * Código QR único por reserva
   * Actualización instantánea en inventario
3. **Retiro en Tienda**
   * Validación QR en POS
   * Tiempo máximo de espera <5 min
   * Flujo para reservas no retiradas

**Exclusiones:**

* Sistema de pago online (probar solo reservas)
* Logística de reposición entre tiendas

**Criterios de Éxito**

| **KPI** | **Objetivo** |
| --- | --- |
| Tiempo confirmación | <15 segundos |
| Precisión stock | 100% concordancia |
| Tasa errores | <0.5% transacciones |

**Matriz de Riesgos**

| **ID** | **Riesgo** | **Severidad** | **Mitigación** |
| --- | --- | --- | --- |
| CC-01 | Stock desactualizado | Crítico | Sincronización cada 30 segundos |
| CC-02 | QR no reconocido | Alto | Validación offline + código manual |
| CC-03 | Retiros fuera de horario | Medio | Notificación automática al staff |

**Tipos de Pruebas**

**1. Pruebas de Estrés**

* 100 reservas simultáneas para mismo producto
* Caída/reconexión de internet durante proceso

**2. Pruebas Regresión**

* Impacto en otros módulos (inventario, CRM)

**Entorno de Pruebas**

| **Componente** | **Configuración** |
| --- | --- |
| Backend | Node.js + WebSockets |
| Base de datos | MongoDB Atlas (geo-replicado) |
| Dispositivos | 3 POS físicos + 5 móviles |

**Cronograma**

| **Fase** | **Duración** |
| --- | --- |
| Preparación | 2 días |
| Ejecución | 5 días |
| Reporte | 1 día |

**7.** **Backup**

**Introducción**  
Este plan verifica el proceso de copias de seguridad diarias automatizadas hacia almacenamiento cloud, asegurando la integridad y disponibilidad de los datos críticos del sistema.

**1. Recursos del Proyecto**

| **Rol** | **Responsabilidades** | **% Participación** |
| --- | --- | --- |
| Administrador BD | Configuración jobs | 100% |
| Especialista Cloud | Validación AWS/GCP | 80% |
| Tester QA | Ejecución pruebas | 70% |

**Alcance de las Pruebas**

**Funcionalidades críticas a validar:**

* **Programación automática**
  + Ejecución puntual diaria (00:00 hs)
  + Trigger por eventos críticos (ej. caída de servidor)
* **Cobertura de datos**
  + Tablas completas de BD principal
  + Archivos de configuración
  + Logs de transacciones
* **Proceso de restauración**
  + Recuperación completa en <4 horas
  + Pruebas de integridad post-restauración

**Exclusiones:**

* Backup de dispositivos endpoint
* Almacenamiento físico externo

**Especificaciones Técnicas**

| **Parámetro** | **Requerimiento** |
| --- | --- |
| Frecuencia | Diaria + incremental cada 6h |
| Retención | 30 días + 1 copia mensual |
| Cifrado | AES-256 |
| Ubicación | Regiones múltiples (AWS us-east-1 + eu-central-1) |

**Tipos de Pruebas**

**Pruebas de Recuperación**

* Escenario 1: Pérdida del 50% de datos
* Escenario 2: Corrupción de tablas críticas

**Pruebas de Seguridad**

* Test de penetración a archivos .bak
* Validación certificados SSL

**5. Matriz de Riesgos**

| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- | --- |
| BK-01 | Backup incompleto | 2 | 5 | Checksum verification |
| BK-02 | Latencia en restore | 3 | 4 | Pre-provisionar recursos |
| BK-03 | Costos cloud excesivos | 2 | 3 | Alarmas por desviación |

**Entorno de Pruebas**

| **Componente** | **Especificación** |
| --- | --- |
| Servidor Origen | SQL Server 2019 (4vCPU, 32GB RAM) |
| Destino Cloud | AWS S3 Intelligent-Tiering |
| Monitorización | Datadog + CloudWatch |

**7. Criterios de Aceptación**

* **RTO (Recovery Time Objective):** <4 horas
* **RPO (Recovery Point Objective):** Máximo 15 min de pérdida
* **Success Rate:** 99.99% en últimos 90 días

**Cronograma**

| **Fase** | **Actividades** | **Duración** |
| --- | --- | --- |
| Fase 1 | Configuración inicial | 3 días |
| Fase 2 | Pruebas automatizadas | 5 días |
| Fase 3 | Auditoría seguridad | 2 días |

**8.** **Segmentación de clientes**

**Introducción**  
Este plan valida el módulo de categorización automática de clientes en el CRM, asegurando que el sistema identifique y clasifique correctamente los perfiles según comportamiento de compra, frecuencia e interacciones.

**Recursos**

| **Rol** | **Responsabilidades** | **% Participación** |
| --- | --- | --- |
| Analista CRM | Definición reglas | 100% |
| Data Engineer | Configuración ETL | 80% |
| QA Tester | Validación resultados | 70% |

**Alcance**

**Categorías a validar:**

* **VIP**: >10 compras/año + ticket promedio >$500
* **Frecuente**: 5-9 compras/año + fidelidad >2 años
* **Ocasional**: <5 compras/año
* **Potencial**: Alto engagement sin compras

**Datos de Prueba**

**Dataset de ejemplo:**

| **Cliente\_ID** | **Compras\_12m** | **Ticket\_Prom** | **Antigüedad** | **Segmento\_Esperado** |
| --- | --- | --- | --- | --- |
| 1001 | 12 | $650 | 3 años | VIP |
| 1002 | 7 | $300 | 4 años | Frecuente |
| 1003 | 2 | $150 | 6 meses | Ocasional |

**Tipos de Pruebas**

**4.1 Pruebas de Reglas**

* Validar bordes (ej.: 9 vs 10 compras para VIP)
* Campos nulos/excepciones

**4.2 Pruebas de Rendimiento**

* Procesamiento de 100K+ registros en <15 min

**4.3 Pruebas de Integración**

* Sincronización con módulo de marketing
* Actualización en dashboards

**Matriz de Riesgos**

| **ID** | **Riesgo** | **Prob.** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- | --- |
| SG-01 | Falsos positivos VIP | 3 | 4 | Auditoría semanal |
| SG-02 | Latencia en actualización | 2 | 3 | Optimizar jobs nocturnos |
| SG-03 | Sesgo en reglas | 2 | 5 | Análisis estadístico trimestral |

**Entorno**

| **Componente** | **Especificación** |
| --- | --- |
| CRM | Salesforce v2023.2 |
| Procesamiento | AWS Glue (ETL) |
| Almacenamiento | Snowflake DW |

**Criterios de Éxito**

* **Precisión:** 98% en clasificación automática
* **Actualización:** Máximo 24h de desfase
* **Cobertura:** 100% clientes activos

**Cronograma**

| **Fase** | **Duración** |
| --- | --- |
| Configuración | 3 días |
| Pruebas piloto | 5 días |
| Implementación | 2 días |