**Plan de pruebas de software**

***Proyecto Plataforma de Gestión – Flota PepsiCo Chile.***

***Fecha: 19/11/2025***

**Historia de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| <2025-10-dd> | <1.1.0> | Documento inicial | <Nombre> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

**1** **Introducción 6**

1.1 Resumen ejecutivo 6

1.2 Alcance de las pruebas 6

*1.2.1* *Elementos de pruebas 6*

*1.2.2* *Pruebas funcionales 6*

*1.2.3* *Riesgos 6*

1.3 Definiciones, acrónimos y abreviaciones 8

1.4 Referencias 8

**2** **Requerimientos para pruebas 9**

2.1 Casos de uso 9

*2.1.1* *Vista global 9*

*2.1.2* *Caso de uso 1 9*

*2.1.3* *Caso de uso 2 9*

*2.1.4* *Caso de uso 3 9*

2.2 Requerimientos funcionales 9

*2.2.1* *Componentes comunes 9*

*2.2.2* *Componente 1 9*

*2.2.3* *Componente 2 9*

2.3 Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica) 9

*2.3.1* *Componente 1 (puede tener uno o más casos de uso) 9*

*2.3.2* *Componente 2 (puede tener uno o más casos de uso) 9*

**3** **Estrategia de Pruebas 10**

3.1 Tipos de pruebas 10

*3.1.1* *Pruebas funcionales 10*

*3.1.2* *Pruebas de rendimiento (Performance) 11*

*3.1.3* *Pruebas de seguridad y de acceso a datos 11*

*3.1.4* *Herramientas involucradas 11*

**4** **Recursos 12**

4.1 Profesionales 12

4.2 Ambiente de pruebas 13

*4.2.1* *Preparación del ambiente de pruebas 13*

*4.2.2* *Diseño del ambiente de pruebas 13*

*4.2.3* *Diseño ambiente de pruebas 15*

*4.2.4* *Integración del ambiente de pruebas y configuración 16*

*4.2.5* *Generación de datos 16*

**5** **Actividades e Hitos del Plan de Pruebas 16**

**6** **Entregables 19**

6.1 Plan de pruebas 19

*6.1.1* *Criterio de entrada para el “Plan de pruebas” 19*

*6.1.2* *Criterio de salida para el “Plan de pruebas” 19*

*6.1.3* *Criterio de suspensión y resumisión 19*

6.2 Resultados de las pruebas 19

6.3 Reporte de defectos 19

**7** **Anexos 20**

7.1 A: Tareas del proyecto 20

7.2 B: Pruebas de rendimiento (*performance*) 21

7.3 C: Pruebas de seguridad y de control de acceso 22

# Introducción

## Resumen ejecutivo

Este Plan de Pruebas define el enfoque, alcance y criterios de aceptación para validar el **PGF – Plataforma de Gestión de Flota Pepsico**. El plan es **detallado** (no maestro) y cubre el ciclo extremo a extremo de atención del vehículo representado por **UC-00** y los diez casos de uso atómicos (**UC-01…UC-10**). El objetivo es demostrar que el sistema cumple los requisitos funcionales y no funcionales definidos para operación en producción, con foco en: flujos E2E, control de acceso por rol/zona/taller (RBAC), manejo de evidencias (imágenes), trazabilidad/auditoría, rendimiento objetivo y estabilidad.

Las pruebas se ejecutarán en **staging** equivalente a producción (pgf-api, pgf-web, PostgreSQL, Redis), con datos semilla controlados y snapshots para restauración. La estrategia es **basada en riesgo y data-driven**: smoke inicial, pruebas funcionales por UC, integración/E2E, UAT por actor y regresión. Los **actores** involucrados en UAT son: Guardia, Recepcionista, Jefe de Taller, Mecánico, Asistente de Repuestos, Encargado de Llaves/Documental, Coordinador de Zona, Supervisor Zonal, Conductor/Ejecutivo de Ventas y Administrador del Sistema.

## Alcance de las pruebas

### Elementos de pruebas

**Áreas/módulos cubiertos (con mapeo a UC):**

**Ingresos y Portería**: UC-01 Registrar Ingreso; UC-07 Pre-Check y Derivación.

**Recepción y Orden de Trabajo**: UC-02 Recepción Técnica; UC-08 Asignar/Priorizar; UC-03 Estados y Pausas; UC-04 Cierre y Salida.

**Aprobaciones y Repuestos**: UC-09 Aprobar/Rechazar (incluye interacción con Asistente de Repuestos).

**Reportes y Exportación**: UC-05 Consultar Reportes/Exportar.

**Mantenimiento Preventivo**: UC-10 Agendar Citas.

**Administración**: UC-06 Usuarios y Catálogos.

**Orquestación E2E**: UC-00 Gestionar Ciclo de Atención (trazabilidad y métricas).

### Pruebas funcionales

# 1.2.2 Pruebas funcionales (texto para pegar)

Las pruebas funcionales serán **caja negra**, diseñadas con datos de entrada representativos y verificación de salidas/efectos en el sistema. Se aplicarán técnicas de **partición de equivalencia** y **valores límite**, **tablas de decisión** para reglas (RBAC, aprobación UC-09) y **transición de estados** para flujos con estados/pausas (UC-03).

**Cobertura de datos (muestras concretas por PGF):**

* **Patente**: formatos válidos/ inválidos (AAA A00 / AA A000 / con y sin guion), min/máx longitud, duplicadas (ingreso abierto).
* **Evidencias (fotos)**: 0, 1, **mínimo requerido (≥2)**, máximo permitido, tamaños **≤5 MB** y >5 MB, extensiones válidas/ inválidas, metadatos.
* **Roles y visibilidad (RBAC)**: Guardia, Recepción, Jefe, Mecánico, Asistente de Repuestos, Encargado de Llaves, Coordinador, Supervisor, Conductor, Admin (acceso a vistas, acciones y filtros por **zona/taller**).
* **OT y estados**: creación, asignación, ejecución, pausas con/ sin motivo, cierre con/ sin pendientes.
* **Aprobación (UC-09)**: montos bajo/igual/sobre **umbral CLP 400.000**, con/ sin stock crítico, con solicitud de ajuste.
* **Agenda (UC-10)**: disponibilidad de taller, conflictos de horario, vínculos a OT existente.
* **Administración (UC-06)**: alta/baja/cambio de rol, **bloqueo del “último admin”**, versionado de catálogos y rollback.
* **Offline (UC-07/UC-01)**: captura en patio sin red, cola de sincronización y reconciliación.

**Resultado esperado:** para cada caso se verificará el estado final (p. ej., “Ingreso en Portería”, “OT En espera”, “OT Cerrada”), la bitácora/auditoría, reglas de negocio aplicadas y mensajes de error por campo.

### Riesgos

Riesgos específicos del PGF a considerar durante el testing:

1. **RBAC mal configurado (zona/taller).** Exposición de datos o acciones fuera de rol.
2. **Conectividad inestable en Portería/Recepción.** Puede impedir registrar ingresos o evidencias.
3. **Evidencias fuera de política.** Imágenes pesadas o con formato no permitido bloquean flujos.
4. **Stock crítico sin ETA confiable.** Bloquea aprobación/ejecución y el cierre.
5. **Skills de mecánico no mapeadas.** No permite asignar correctamente en UC-08.
6. **Catálogos/roles desactualizados o inconsistentes.** Reglas de negocio aplicadas de forma errónea.

Los riesgos serán identificados de acuerdo a un concepto de Bajo, Medio o Alto, dependiendo de la importancia del caso de uso para el cual se está desarrollando el *testing*.

Así mismo, se han identificado una serie de riesgos (calificados entre 1 y 10 dependiendo de su gravedad) los que están detallados en el artefacto “Lista de riesgos” con sus alcances y acciones, en el presente plan, son enunciados para sugerir algunas acciones.

#### Matrices de riesgos

##### Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | RBAC mal configurado (zona/taller) | 9 (Alto) | Suite de permisos por rol; pruebas de **IDOR** (acceso directo por URL); revisión de perfiles antes de UAT; logging y alerta ante acceso denegado. |
| 2 | Conectividad inestable en Portería/Recepción | 8 (Alto) | Escenarios **offline** obligatorios (cola y reconciliación); pruebas en red real; procedimiento de contingencia documentado. |
| 3 | Evidencias fuera de política (tipo/tamaño/metadatos) | 7 (Medio-Alto) | Validaciones cliente/servidor; compresión/resize; límites claros (≤5 MB, formatos permitidos); casos negativos. |
| 4 | Stock crítico sin ETA | 7 (Medio-Alto) | Datos semilla con alternativos; casos de replanificación; reglas en UC-09/UC-03; simulación de proveedor alterno. |
| 5 | Duplicidad de Ingresos/OT por concurrencia | 8 (Alto) | Pruebas de concurrencia con misma patente; **llaves únicas y locks**; idempotencia en endpoints; manejo de reintentos; monitoreo de duplicados. |
| 6 | Catálogos/roles desactualizados | 6 (Medio) | Refuerzo de **UC-06** (bitácora y rollback); checklist de sincronización previo; snapshot/restauración de BD. |

**1) RBAC mal configurado (zona/taller)**

* **Qué es:** Permisos o filtros incorrectos permiten ver/hacer cosas fuera del rol o del ámbito (zona/taller).
* **Cuándo ocurre:** Perfiles mal asignados, endpoints sin filtro por zona, enlaces directos a IDs.
* **Por qué importa:** Exposición de datos y acciones no autorizadas.
* **Señales:** Respuestas **200** donde debería haber **403**; listados que incluyen otras zonas.
* **Prueba:** Matriz por rol (permitido/denegado), intentos de acceso directo a recursos de otra zona (IDOR), verificación de filtros en listados.
* **Control:** Filtros por zona/taller en todas las consultas; políticas por rol; pruebas automáticas de permisos; logging de denegaciones.

**2) Conectividad inestable en Portería/Recepción**

* **Qué es:** Cortes de red durante el registro de ingresos o evidencias.
* **Cuándo ocurre:** Wi-Fi débil en patio/recepción, caídas intermitentes.
* **Por qué importa:** Pérdida/duplicidad de registros, demoras.
* **Señales:** Formularios que no envían, fotos que fallan, registros que aparecen luego en lote.
* **Prueba:** Simular modo **offline** (navegador) al crear un ingreso con fotos; reconectar y verificar sincronización sin duplicados.
* **Control:** Cola local, reintentos, idempotencia, mensajes claros de “pendiente de sincronizar”.

**3) Evidencias fuera de política (tipo/tamaño/metadatos)**

* **Qué es:** Archivos no permitidos, demasiado grandes o con metadatos sensibles.
* **Cuándo ocurre:** Capturas desde móviles sin compresión, formatos no validados.
* **Por qué importa:** Bloquea flujos, lentitud, exposición de datos EXIF.
* **Señales:** Errores por tamaño/formatos; tiempos de carga anormales.
* **Prueba:** Casos límite de tamaño (≤/ > 5 MB), conteo mínimo de fotos (p. ej., ≥2), extensiones válidas/invalidas, verificación de limpieza de metadatos.
* **Control:** Validación en cliente y servidor, compresión/resize, eliminación de metadatos, límites configurables.

**4) Stock crítico sin ETA confiable**

* **Qué es:** Repuestos necesarios sin fecha estimada o sin alternativa.
* **Cuándo ocurre:** Inventario insuficiente o proveedor sin confirmación.
* **Por qué importa:** Bloquea aprobación (UC-09), detiene ejecución/cierre (UC-03/UC-04).
* **Señales:** OTs detenidas por faltantes; aprobaciones en espera indefinida.
* **Prueba:** Escenarios con stock=0 y ETA nula; replanificación obligatoria; alertas visibles en pantallas y reportes.
* **Control:** Proveedor alterno/sustitutos, reglas de reprogramación, visibilidad en reportes para priorización.

**5) Duplicidad de Ingresos/OT por concurrencia**

* **Qué es:** Dos usuarios crean el mismo Ingreso/OT casi al mismo tiempo.
* **Cuándo ocurre:** En horas pico o con reconexiones después de modo offline.
* **Por qué importa:** Inconsistencias, trabajo duplicado, métricas infladas.
* **Señales:** Múltiples ingresos “En portería” para la misma patente; OTs duplicadas asociadas a un ingreso.
* **Prueba:** Disparos concurrentes (hilos/usuarios) con la misma patente; esperar 1 éxito y rechazos controlados (400/409) en el resto.
* **Control:** Restricciones únicas (patente con ingreso abierto), **idempotency keys**, locks/transacciones, manejo correcto de reintentos.

**6) Catálogos/roles desactualizados o inconsistentes**

* **Qué es:** Catálogos (servicios, repuestos) y roles que no reflejan la operación actual.
* **Cuándo ocurre:** Cambios operativos sin actualización del sistema.
* **Por qué importa:** Reglas mal aplicadas, asignación/aprobación incorrecta, cierres fallidos.
* **Señales:** Servicios inexistentes referenciados; rol sin permisos clave; errores al cerrar OT por datos obsoletos.
* **Prueba:** Alta/baja/modificación con **versionado**; **rollback**; validación de referencias (no eliminar si está en uso).
* **Control:** Bitácora y versionado en UC-06, checklist previo a cada ciclo, snapshot/restore de base de datos.

##### 

## Definiciones, acrónimos y abreviaciones

**Glosario (proyecto y pruebas)**

* **PGF:** Plataforma de Gestión de Flota.
* **UC-00 / UC-01…UC-10:** Casos de uso del sistema; UC-00 es orquestador E2E.
* **E2E**: End-to-End, siendo el funcionamiento completo del sistema.
* **OT:** Orden de Trabajo.
* **RBAC:** Control de acceso basado en roles (scope por zona/taller).
* **E2E:** End-to-End; flujo extremo a extremo.
* **UAT:** User Acceptance Testing; pruebas de aceptación por usuarios finales.
* **SLA:** Service Level Agreement; objetivo de servicio (tiempos/umbrales).
* **TAT:** Turnaround Time; tiempo total del ciclo de atención.
* **p95:** Percentil 95 de tiempo de respuesta.
* **IDOR:** Insecure Direct Object Reference; acceso directo no autorizado a recursos.
* **DAST:** Dynamic Application Security Testing (ej. OWASP ZAP).
* **KPI:** Indicador clave de desempeño (pase, defect density, p95, tiempo E2E).
* **Severidad (S1–S4):** Impacto del defecto; S1 crítico / S4 cosmético.
* **Prioridad (P0–P3):** Urgencia de corrección; P0 inmediata / P3 baja.
* **Snapshot:** Copia puntual de la base de datos usada para restaurar el ambiente.
* **Semilla (seed):** Conjunto de datos iniciales de prueba.
* **FR:** Requerimientos funcionales

**Actores (abreviaciones usadas en el documento)**

* **G:** Guardia
* **R:** Recepcionista
* **JT:** Jefe de Taller
* **M:** Mecánico
* **REP:** Asistente de Repuestos
* **LLAV:** Encargado de Llaves y Control Documental
* **CZ:** Coordinador de Zona
* **SZ:** Supervisor Zonal
* **COND:** Conductor / Ejecutivo de Ventas
* **ADM:** Administrador del Sistema
* **SYS:** Sistema (PGF)

## Referencias

* **Especificación de Casos de Uso – PGF (UC-00 + UC-01…UC-10)**, vX.Y, [fecha], Autor: [equipo].
* **Diagrama de Casos de Uso (PlantUML/PNG) – “PGF – Plataforma de Gestión de Flota”**, vX.Y, [fecha].
* **Matriz de Trazabilidad (Req ↔ UC ↔ Casos de Prueba ↔ Defectos)**, vX.Y, [fecha].
* **Requisitos Funcionales y No Funcionales – PGF**, vX.Y, [fecha].
* **Plan de Implantación, Soporte y Mantenimiento – PGF**, vX.Y, [fecha].
* **Datos de Prueba (Semillas/Snapshots) – PGF**, vX.Y, [fecha].
* **Plantilla Plan de Pruebas (este documento)**, vX.Y, [fecha].

**Referencias externas**

* **ISO/IEC/IEEE 29119-3** – Test documentation.
* **OWASP Web Security Testing Guide (WSTG v5)** y **OWASP Top 10**.
* **Playwright Test** – Documentación de automatización UI.
* **pytest + requests** – Pruebas de API.
* **Locust** – Pruebas de carga/rendimiento.
* **OWASP ZAP** – Escaneo DAST.

# Requerimientos para pruebas

La siguiente lista identifica los ítems (casos de uso, requerimientos funcionales y requerimientos no funcionales) que se han identificado como requerimientos a ser probados.

## Casos de uso

### Vista global

El sistema PGF se prueba mediante **UC-00 (orquestador E2E)** y **UC-01…UC-10**.  
UC-00 incluye los casos del “core” (UC-01, UC-02, UC-03, UC-04, UC-06, UC-07, UC-08, UC-09) y es **extendido** por los opcionales **UC-05 (reportes)** y **UC-10 (preventivo/citas)**.  
Actores considerados: Guardia, Recepcionista, Jefe de Taller, Mecánico, Asistente de Repuestos, Encargado de Llaves y Control Documental, Coordinador de Zona, Supervisor Zonal, Conductor/Ejecutivo de Ventas, Administrador del Sistema y Sistema (PGF).

### Caso de uso 0: UC-00 – Gestionar Ciclo E2E de Atención de Vehículo

**Propósito:** Orquestar el ciclo completo desde pre‐check/ingreso hasta cierre de atención, incluyendo aprobaciones y asignaciones.  
**Actores principales:** Guardia, Recepción, Jefe, Mecánico, Supervisor, Asistente de Repuestos, Encargado de Llaves, Coordinador, Conductor, Admin, Sistema.  
**Criterios de aceptación (pruebas):**

* Ejecutar **tres E2E**: sin aprobación; con aprobación (UC-09); preventivo (UC-10).
* Estados y transiciones consistentes; trazabilidad y auditoría completas.
* Métricas de tiempo del ciclo registradas.

### Caso de uso 1: UC-01 – Registrar Ingreso en Portería

**Propósito:** Crear un **Ingreso** con patente, taller destino y evidencias.  
**Actores:** Guardia, Sistema.  
**Criterios:**

* Validación de patente y evidencias (mínimo, formato, tamaño).
* Rechazo de **duplicado** si ya existe ingreso abierto para la patente.
* Modo **offline** con sincronización posterior.

### Caso de uso 2: UC-02 – Recepción Técnica y Creación de OT

**Propósito:** Evaluar vehículo y crear **OT** asociada.  
**Actores:** Recepcionista, Jefe de Taller.  
**Criterios:**

* Creación de OT con campos obligatorios y vinculación al Ingreso.
* Bloqueos por vehículo no habilitado o datos incompletos.

### Caso de uso 3: UC-03 – Gestionar Estados y Pausas

**Propósito:** Controlar ejecución de OT (iniciar, pausar, reanudar, finalizar).  
**Actores:** Mecánico, Jefe de Taller.  
**Criterios:**

* Registro de **motivo** en pausa; temporización y SLA.
* Impedir cierre si existen pendientes no resueltos.

### Caso de uso 4: UC-04 – Cierre de Atención y Salida

**Propósito:** Validar trabajos/repuestos y **cerrar** la OT; autorización de salida.  
**Actores:** Jefe de Taller, Guardia, Encargado de Llaves.  
**Criterios:**

* Checklist completo, firmas/documentos, devolución de llaves.
* Estados finales consistentes y bitácora de cierre.

### Caso de uso 5: UC-05 – Consultar Reportes y Exportar

**Propósito:** Consultar indicadores y **exportar** datos.  
**Actores:** Jefe de Taller, Coordinador de Zona, Supervisor Zonal.  
**Criterios:**

* Filtros por **zona/taller/fechas**; **RBAC** correcto.
* Exportación a CSV/XLS con registro de auditoría.

### Caso de uso 6: UC-06 – Administrar Usuarios y Catálogos

**Propósito:** Gestión de usuarios/roles y catálogos (servicios, repuestos).  
**Actores:** Administrador del Sistema.  
**Criterios:**

* Alta/baja/modificación con **versionado** y bitácora.
* Protección del **último administrador** (no degradar/eliminar).
* Validación referencial (no eliminar ítems en uso).

### Caso de uso 7: UC-07 – Pre-Check de Seguridad y Derivación a Recepción

**Propósito:** Ejecutar checklist de seguridad y derivar a recepción.  
**Actores:** Guardia.  
**Criterios:**

* Bloqueo por ítems críticos; adjunto de fotos.
* Operación en **entorno con conectividad variable** (pruebas offline).

### Caso de uso 8: UC-08 – Asignar y Priorizar OT

**Propósito:** Priorizar y **asignar** OT a mecánicos según disponibilidad/skills.  
**Actores:** Jefe de Taller, Mecánico.  
**Criterios:**

* Sugerencias por disponibilidad; incompatibilidades detectadas.
* Replanificación ante conflictos (agenda o repuestos).

### Caso de Uso 9: UC-09 – Aprobar/Rechazar Estimación y Repuestos

**Propósito:** Aprobar o rechazar estimaciones y reserva de repuestos.  
**Actores:** Supervisor Zonal, Asistente de Repuestos.  
**Criterios:**

* Umbral de aprobación (≥ CLP 400.000); solicitud de ajuste.
* Manejo de **stock crítico** y ETA; reserva efectiva.

### Caso de uso 10: UC-10 – Agendar Mantenimiento Preventivo y Citas

**Propósito:** Crear y gestionar **citas** de mantenimiento.  
**Actores:** Coordinador de Zona, Recepcionista, Conductor.  
**Criterios:**

* Disponibilidad de taller y recordatorios; conflictos de agenda.
* Vinculación a OT existente o pre-OT.

## Requerimientos funcionales

### Componentes comunes

 **FR-CC-01 Autenticación y sesiones.** Inicio/cierre; expiración; un solo login activo por usuario.

 **FR-CC-02 RBAC por rol–zona–taller.** Toda consulta/acción respeta el ámbito asignado.

 **FR-CC-03 Auditoría.** Crear/editar/borrar, cambios de estado y exportaciones quedan en bitácora con usuario, fecha y origen.

 **FR-CC-04 Adjuntos.** Imágenes permitidas (formatos y tamaño), mínimo requerido por flujo y limpieza de metadatos.

 **FR-CC-05 Búsqueda y filtros.** Por patente, estado, taller, fechas; orden y paginación.

 **FR-CC-06 Estados estándar.** Definición y transición válidas por entidad (Ingreso, OT, Cita).

 **FR-CC-07 Exportación.** CSV/XLS según permisos, con registro de auditoría.

 **FR-CC-08 Notificaciones.** Recordatorios de citas, avisos de asignación/cambio de estado.

 **FR-CC-09 Integridad referencial.** No eliminar catálogos en uso; validaciones al actualizar.

 **FR-CC-10 Concurrencia e idempotencia.** Prevenir duplicidad (Ingresos/OT) ante envíos repetidos o simultáneos.

 **FR-CC-11 Modo offline (portería/recepción).** Cola local y sincronización automática.

 **FR-CC-12 Localización y zona horaria.** Tiempos y reportes consistentes (TZ corporativa).

### Componente 1 - Ingresos y Portería (UC-01, UC-07)

 **FR-IP-01 Crear Ingreso.** Patente, taller destino, hora automática, usuario; validación de formato.

 **FR-IP-02 Unicidad con abierto.** Rechazo si existe Ingreso abierto para la patente.

 **FR-IP-03 Evidencias.** Captura/carga múltiple con mínimo requerido; tamaño/formatos validados.

 **FR-IP-04 Pre-check de seguridad.** Checklist; bloqueo por ítem crítico; observaciones.

 **FR-IP-05 Derivación.** Envío del vehículo a Recepción, registrando hora y destino.

 **FR-IP-06 Operación offline.** Registrar Ingreso/Pre-check sin red y sincronizar al reconectar.

 **FR-IP-07 Consulta.** Listado de Ingresos con filtros (estado, taller, fecha, patente).

 **FR-IP-08 Edición controlada.** Solo antes de derivar; todo cambio queda auditado.

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

 **FR-RO-01 Crear OT.** Desde Ingreso; datos de falla y prioridad; responsable inicial.

 **FR-RO-02 Asignación.** A mecánico según disponibilidad/skills; posibilidad de replanificación.

 **FR-RO-03 Priorización.** Cambiar prioridad con motivo y trazabilidad.

 **FR-RO-04 Ejecución.** Iniciar, pausar (motivo obligatorio), reanudar y finalizar; cronometraje.

 **FR-RO-05 Consumo de repuestos/servicios.** Registrar cantidades y precios; control de stock.

 **FR-RO-06 Interacción con aprobación.** Enviar/recibir resultado de UC-09 y reflejarlo en la OT.

 **FR-RO-07 Restricciones de cierre.** Impedir cierre con pendientes o faltantes de evidencia.

 **FR-RO-08 Cierre y documentación.** Checklist final, firmas/documentos y estado “Cerrada”.

 **FR-RO-09 Autorización de salida.** Validación en portería y control de llaves (encargado).

 **FR-RO-10 Historial.** Línea de tiempo de estados, pausas y responsables.

 **FR-RO-11 Notificaciones.** A mecánico/asignados ante cambios relevantes.

 **FR-RO-12 Consulta de OTs.** Filtros por estado, taller, fecha, mecánico y prioridad.

### Componente 3 - Aprobaciones y Repuestos (UC-09)

 **FR-AR-01 Crear estimación.** Ítems de mano de obra y repuestos, descuentos/impuestos; cálculo automático del total.

 **FR-AR-02 Umbral de aprobación.** Umbral configurable; si el total **≤ umbral** aprueba Jefe; si **> umbral** eleva a Supervisor. Motivo obligatorio en rechazo.

 **FR-AR-03 Flujo de decisión.** Estados: “Pendiente”, “Aprobada”, “Rechazada”, “Ajuste solicitado”. Comentario obligatorio en rechazo/ajuste.

 **FR-AR-04 Reserva de stock.** Al aprobar, reservar existencias; si faltan, generar alerta y **ETA**; impedir iniciar trabajo sin reserva crítica.

 **FR-AR-05 Sustituciones.** Permitir repuesto alternativo/equivalente con justificación y trazabilidad.

 **FR-AR-06 Integridad de precios.** Tomar precios vigentes del catálogo con fecha de validez; no permitir edición directa sin rol.

 **FR-AR-07 Notificaciones.** Avisar a Jefe, Mecánico y Asistente de Repuestos ante decisiones y cambios.

 **FR-AR-08 Versión y reapertura.** Control de versiones; solo reabrir si la OT no ha iniciado ejecución.

 **FR-AR-09 Trazabilidad a consumo.** Ítems aprobados se reflejan como consumos en UC-03 y en el cierre UC-04.

 **FR-AR-10 Antifraude de fraccionamiento.** Bloquear dividir estimaciones para eludir el umbral; auditoría del intento.

 **FR-AR-11 Liberación de stock.** Si se rechaza o no se usa al cerrar, liberar reservas automáticamente con registro.

### Componente 4 - Reportes y Exportaciones (UC-05)

 **FR-RP-01 Tablero de KPIs.** OTs abiertas, TAT, cumplimiento SLA, ingresos por día, uso de repuestos, citas próximas.

 **FR-RP-02 Filtros.** Rango de fechas, zona, taller, estado, mecánico, tipo de servicio.

 **FR-RP-03 Alcance por RBAC.** Jefe ve su taller; Supervisor/Coordinador su zona; Admin global.

 **FR-RP-04 Drill-down.** Desde KPI a lista y a detalle de OT/Ingreso.

 **FR-RP-05 Exportación.** CSV/XLS de tablas respetando filtros y columnas visibles; registro en auditoría.

 **FR-RP-06 Rendimiento.** Tiempo de renderizado p95 < **3 s** con paginación.

 **FR-RP-07 Consistencia de zona horaria.** Mostrar/expotar fechas en TZ corporativa.

 **FR-RP-08 Programas y logs.** Registrar usuario, filtros y timestamp de cada exportación.

### Componente 5 - Mantenimiento Preventivo y Citas (UC-10)

 **FR-MP-01 Planes preventivos.** Catálogo por familia de vehículo (km/horas/tiempo) con vigencia.

 **FR-MP-02 Próxima intervención.** Cálculo automático según odómetro/fecha; recálculo al actualizar lecturas.

 **FR-MP-03 Crear cita.** Vehículo, taller, ventana de tiempo, responsable; validar **capacidad** del taller.

 **FR-MP-04 Conflictos.** Evitar solapamientos; advertir si el vehículo ya tiene cita activa.

 **FR-MP-05 Recordatorios.** Notificaciones a Conductor, Recepción y Taller (antes/después, configurable).

 **FR-MP-06 Reprogramación y cancelación.** Motivo obligatorio; historial de cambios.

 **FR-MP-07 Vinculación a OT.** Al ingreso, crear/pre-vincular OT a la cita; mantener trazabilidad.

 **FR-MP-08 Agenda del taller.** Vista calendario por box/mecánico; filtros por día/semana/mes.

 **FR-MP-09 Cumplimiento.** Reporte de adherencia al plan preventivo por vehículo/zona.

### Componente 6 - Administración y Catálogos (UC-06)

 **FR-AD-01 Usuarios.** Alta/baja/edición con asignación de **rol** y **ámbito** (zona/taller); restablecer contraseña.

 **FR-AD-02 Roles y permisos.** Gestión de plantillas por actor; verificación de dependencias antes de cambios.

 **FR-AD-03 Protección del último admin.** No permitir degradar/eliminar si es el único administrador activo.

 **FR-AD-04 Catálogos.** Servicios, repuestos, checklists, motivos de pausa, talleres, habilidades; estados **activo/inactivo** y **vigencias**.

 **FR-AD-05 Importación masiva.** CSV con validación previa y reporte de errores; opción de simulación (“dry-run”).

 **FR-AD-06 Dependencias.** Impedir eliminar elementos referenciados por OT/Ingreso/Estimación; mensajes claros.

 **FR-AD-07 Parámetros del sistema.** Umbral de aprobación, límites de adjuntos, tamaños máximos, TZ; todo con auditoría.

 **FR-AD-08 Bitácora.** Registro antes/después (quién, cuándo, desde dónde) de toda operación administrativa.

 **FR-AD-09 Seguridad reforzada.** Endpoints administrativos protegidos; opción de 2FA si está habilitada.

 **FR-AD-10 Reportes de cambios.** Listado de modificaciones a catálogos y usuarios con filtros y exportación.

## Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica)

### Componente 1 - Ingresos y Portería (UC-01, UC-07)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-01 Rendimiento subida | Cargar **3–5 fotos** de **1–2 MB** c/u y guardar ingreso | Respuesta UI **p95 < 2 s**; subida total **≤ 6 s**; ingreso queda “En portería”. |
| 2 | UC-01 Duplicidad por concurrencia. | **5 solicitudes** simultáneas para la **misma patente** | **1** éxito (200/201) y **resto 400/409**; sin ingresos duplicados; auditoría única. |
| 3 | UC-01/UC-07 Validación de adjuntos | Subir **1 archivo de 7 MB** y **extensión no permitida** | Rechazo del servidor con mensaje claro; **no** se guarda archivo ni registro parcial. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | Auditoría de creación | Crear ingreso exitoso | Registro en bitácora con **usuario, fecha/hora, IP, patente, taller, #fotos**. |

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-02 Rendimiento creación de OT | Crear OT desde un Ingreso con datos completos | Respuesta **p95 < 2 s**; OT queda ligada al Ingreso; auditoría registrada. |
| 2 | UC-08 Lista de asignación | Cargar vista con **200 OTs abiertas** y **50 mecánicos** | Render **p95 < 2.5 s** con paginación/filtros activos. |
| 3 | UC-03 Cronometraje de pausas | Iniciar → pausar (10 min) → reanudar | Medición con **desviación ≤ ±1 s / 10 min**; persistencia, aunque se recargue la página. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | UC-02 Concurrencia sobre un Ingreso | **2 recepcionistas** crean OT sobre el **mismo ingreso** | **1** OT creada; el segundo intento **409** “ya existe OT”; auditoría única. |

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-02 Rendimiento creación de OT | Crear OT desde un Ingreso con datos completos | Respuesta **p95 < 2 s**; OT queda ligada al Ingreso; auditoría registrada. |
| 2 | UC-08 Lista de asignación | Cargar vista con **200 OTs abiertas** y **50 mecánicos** | Render **p95 < 2.5 s** con paginación/filtros activos. |
| 3 | UC-03 Cronometraje de pausas | Iniciar → pausar (10 min) → reanudar | Medición con **desviación ≤ ±1 s / 10 min**; persistencia, aunque se recargue la página. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | UC-02 Concurrencia sobre un Ingreso | **2 recepcionistas** crean OT sobre el **mismo ingreso** | **1** OT creada; el segundo intento **409** “ya existe OT”; auditoría única. |

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-02 Rendimiento creación de OT | Crear OT desde un Ingreso con datos completos | Respuesta **p95 < 2 s**; OT queda ligada al Ingreso; auditoría registrada. |
| 2 | UC-08 Lista de asignación | Cargar vista con **200 OTs abiertas** y **50 mecánicos** | Render **p95 < 2.5 s** con paginación/filtros activos. |
| 3 | UC-03 Cronometraje de pausas | Iniciar → pausar (10 min) → reanudar | Medición con **desviación ≤ ±1 s / 10 min**; persistencia, aunque se recargue la página. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | UC-02 Concurrencia sobre un Ingreso | **2 recepcionistas** crean OT sobre el **mismo ingreso** | **1** OT creada; el segundo intento **409** “ya existe OT”; auditoría única. |

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-02 Rendimiento creación de OT | Crear OT desde un Ingreso con datos completos | Respuesta **p95 < 2 s**; OT queda ligada al Ingreso; auditoría registrada. |
| 2 | UC-08 Lista de asignación | Cargar vista con **200 OTs abiertas** y **50 mecánicos** | Render **p95 < 2.5 s** con paginación/filtros activos. |
| 3 | UC-03 Cronometraje de pausas | Iniciar → pausar (10 min) → reanudar | Medición con **desviación ≤ ±1 s / 10 min**; persistencia, aunque se recargue la página. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | UC-02 Concurrencia sobre un Ingreso | **2 recepcionistas** crean OT sobre el **mismo ingreso** | **1** OT creada; el segundo intento **409** “ya existe OT”; auditoría única. |

### Componente 2 - Recepción y OT (UC-02, UC-08, UC-03, UC-04)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | UC-02 Rendimiento creación de OT | Crear OT desde un Ingreso con datos completos | Respuesta **p95 < 2 s**; OT queda ligada al Ingreso; auditoría registrada. |
| 2 | UC-08 Lista de asignación | Cargar vista con **200 OTs abiertas** y **50 mecánicos** | Render **p95 < 2.5 s** con paginación/filtros activos. |
| 3 | UC-03 Cronometraje de pausas | Iniciar → pausar (10 min) → reanudar | Medición con **desviación ≤ ±1 s / 10 min**; persistencia, aunque se recargue la página. |
| 4 | UC-03/UC-04 **Transaccionalidad** de cierre | Cerrar OT con consumo de repuestos/servicios | Si algún paso falla, **rollback completo**; **sin estados zombies**; mensaje claro. |
| 5 | RBAC en OT | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. | RBAC en OT | Jefe de Taller intenta ver/editar OT de otra **zona/taller** | **403** en edición; listados filtrados a su taller; logs de denegación. |
| 6 | UC-02 Concurrencia sobre un Ingreso | **2 recepcionistas** crean OT sobre el **mismo ingreso** | **1** OT creada; el segundo intento **409** “ya existe OT”; auditoría única. |

# Estrategia de Pruebas

A continuación se describe la estrategia de testing a ser utilizada, en otras palabras, se indica cómo se realizará el testing de los requerimientos analizados en la sección precedente (Requerimientos para pruebas) que identifica qué se va a testear.

## Tipos de pruebas

### Pruebas funcionales

El *testing* funcional se realizará sobre los requerimientos funcionales antes descritos y sus casos de uso. Estas pruebas tienen por finalidad comprobar la funcionalidad de la aplicación a partir de datos válidamente seleccionados sobre las transacciones del sistema.

Este tipo de comprobación se basa en las técnicas de caja negra, que permiten probar la aplicación (y sus procesos interinos) vía GUI.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Validar que **UC-00** (E2E) y **UC-01…UC-10** funcionen conforme a las reglas del negocio: ingreso y pre-check, creación y asignación de OT, gestión de estados/pausas, cierre y autorización de salida, aprobaciones con umbral y reserva de stock, reportes, administración y mantenimiento preventivo.   Verificar **RBAC** (rol–zona–taller), **adjuntos** (tipo/tamaño/cantidad y limpieza de metadatos), **auditoría** de acciones y manejo **offline** en portería/recepción.   Detectar defectos funcionales y regresiones antes del pase a producción. |
| Técnica a utilizar: |  **Caja negra** con **partición de equivalencia** y **valores límite** para entradas (patente, tamaños de fotos, montos de estimación, etc.).   **Tablas de decisión** para aprobaciones por umbral, permisos RBAC y priorización.   **Máquinas de estados** para UC-03/UC-04 (iniciar/pausar/reanudar/finalizar/cerrar).   **Data-driven** con semillas controladas y **negativos** (duplicados, formatos inválidos).   **Concurrencia** dirigida (duplicidad de Ingresos/OT).   Evidencia con capturas/logs y trazabilidad a requisitos. |
| Criterio de validación: |  Cada caso de prueba debe cumplir:   1. **Resultado esperado** visible en UI/API y persistencia correcta de datos. 2. **Mensajes de error** adecuados ante datos inválidos o acciones no permitidas. 3. **Reglas de negocio** aplicadas (p. ej., rechazo de ingreso duplicado, motivo obligatorio en pausa, umbral de aprobación, bloqueo de cierre con pendientes). 4. **RBAC** respeta el ámbito (403/ocultamiento fuera de zona/taller). 5. **Auditoría** registra quién/cuándo/qué cambió.    **Criterio global de aceptación del ciclo:** ≥95% de casos *realizados* y 0 defectos **S1–S2** abiertos. |
| Consideraciones especiales: |  Se prioriza funcionalidad y reglas; no se verifican todas las propiedades visuales si no afectan el negocio.   Ejecutar en **staging** equivalente a producción, con **snapshot/restore** por iteración.   Usar **datos no sensibles**; semillas con vehículos/talleres/usuarios conocidos.   Integraciones externas no implementadas quedan fuera del alcance.   Probar **offline** en UC-07/UC-01 y **concurrencia** en UC-01/UC-02. |
| Observaciones: |  Las suites se organizan por **UC** y por **E2E** (UC-00).   Toda evidencia (screenshots, HAR, logs) debe referenciar **ID de caso, build y fecha**.   Los hallazgos se registran con **Severidad (S1–S4)** y **Prioridad (P0–P3)** y se revalidan en regresión. |

### Pruebas de rendimiento (Performance)

Un detalle de este tipo de *testing*, adjunta en el Anexo B de este documento.

**Objetivo**  
Validar que la plataforma cumpla tiempos de respuesta, estabilidad y uso eficiente de recursos bajo carga representativa, con foco en los flujos críticos (UC-01, UC-02, UC-03/UC-04, UC-05, UC-08, UC-09).

**Alcance**

* UI (medición con Playwright/cronómetros) y API (Locust).
* Carga concurrente moderada (operación normal) y micro-picos.
* No incluye pruebas de estrés 24×7.

**Métricas y umbrales (p95 salvo donde se indique)**

* Acciones de UI: **< 2 s**.
* UC-01 (crear ingreso con 3–5 fotos 1–2 MB c/u): **≤ 6 s** total de subida.
* UC-07 checklist + derivación: **≤ 90 s** operativo.
* UC-08 (abrir bandeja/guardar asignación): **< 2.5 s / < 2 s**.
* UC-04 cierre con validaciones: **≤ 90 s**.
* UC-05 render tablero: **< 3 s**; **export 50.000 filas ≤ 30 s**.
* Error rate HTTP (≥400): **≤ 1%**.
* Estabilidad: sin degradación progresiva ni fugas de memoria.

**Metodología**

1. **Baseline UI**: medir con Playwright tiempos de interacción (n=30 por caso).
2. **Carga API** (Locust):
   * **Escenario A – Operación normal:** 30 usuarios concurrentes, rampa 10 min, sostener 15 min; mezcla de tareas: UC-01 10%, UC-02 10%, UC-03 30%, UC-08 15%, UC-04 10%, UC-05 20%, UC-09 5%.
   * **Escenario B – Pico corto:** 60 concurrentes por 10 min (misma mezcla).
3. **Monitoreo**: CPU/memoria de pgf-api y pgf-db, tiempos SQL, colas, GC.
4. **Datos**: semillas controladas; 10 vehículos por zona, 3 talleres, 50 mecánicos, 30 repuestos.
5. **Repetibilidad**: ejecutar con la misma versión/tag y snapshot.

**Criterios de aceptación**

* Todos los umbrales anteriores cumplidos en **A** y **B**.
* Error rate ≤ 1%; sin timeouts; sin reinicios del contenedor.
* Sin cuellos críticos en consultas (planes/índices ajustados si aparecen).

**Entregables (Anexo B)**

* Scripts (locustfile.py, colección de tareas), reportes HTML/CSV, gráficos p50/p95/p99, consumo de recursos, hallazgos y recomendaciones.

### Pruebas de seguridad y de acceso a datos

Un detalle de este tipo de *testing*, adjunta en el Anexo C de este documento.

**Objetivo**  
Verificar autenticación, autorización por rol/zona/taller (RBAC), manejo seguro de archivos, validación de entradas y exposición mínima de datos.

**Alcance**

* **Autenticación** y gestión de sesión.
* **Autorización/RBAC** en UI y API (incluye **IDOR**).
* **Cargas de archivos** (extensión/tamaño/metadatos).
* **Validación de entrada** y salidas (XSS reflejado básico).
* **Config. de cabeceras** y cookies.

**Controles a verificar**

* **RBAC**: Jefe ve solo su taller; Supervisor/Coordinador su zona; Admin global.
* **IDOR**: acceso directo a /api/ot/{id} o recursos de otra zona debe devolver **403**.
* **Uploads**: limitar tipos (jpg/png), tamaño (≤ **5 MB**), conteo; limpiar EXIF.
* **Inputs/outputs**: escape de HTML en campos de texto; rechazo de payloads malformados.
* **Sesión/cookies**: HttpOnly, Secure, SameSite, expiración; logout invalida token.
* **Cabeceras**: Content-Security-Policy (si aplica), X-Content-Type-Options: nosniff, X-Frame-Options: DENY.

**Metodología**

1. **DAST** automatizado (ZAP baseline) contra BASE\_URL para hallazgos comunes.
2. **Pruebas manuales** guiadas por checklist RBAC e IDOR (por cada rol).
3. **Pruebas negativas** de subida de archivos (tipo/tamaño/duplicados).
4. **Fuzzing ligero** de parámetros críticos (IDs, filtros, montos).

**Casos mínimos por módulo**

* **UC-05 Reportes**: validar que filtros restringen datos por ámbito; exportaciones respetan RBAC; no exponen campos sensibles.
* **UC-01/UC-07**: uploads con tipos no permitidos, >5MB, y verificación de limpieza EXIF.
* **UC-02/UC-08/UC-03/UC-04**: IDOR en OT y cambios de estado entre zonas.
* **UC-06**: protección de endpoints administrativos; bloqueo “último admin”.

**Criterios de aceptación**

* 0 vulnerabilidades **Alta** abiertas (según ZAP/OWASP).
* 0 bypass de **RBAC/IDOR** reproducible.
* Uploads y validaciones cumplen política; cookies/cabeceras seguras presentes.

**Entregables (Anexo C)**

* Informe ZAP (HTML/XML), checklist RBAC ejecutado por rol con evidencias, lista de vulnerabilidades y plan de remediación (si aplica).

### Herramientas involucradas

| **Herramienta** | **Vendedor/Tienda** | **Versión** |
| --- | --- | --- |
| **Guía de Referencia (Testing)** | **ISTQB – International Software Testing Qualifications Board** | **2024** |
| **Administración de Testing** | **Jira Software + Xray Test Management (Atlassian)** | **Cloud – Última versión** |
| **Seguimiento de Defectos** | **Jira Software / GitHub Issues** | **Cloud – Última versión** |
| **Testing Funcional** | **Cypress.io Automation Framework** | **13.x** |
| **Testing E2E (End-to-End)** | **Microsoft Playwright** | **1.45.x** |
| **ASQ Tool (Performance Testing)** | **Apache JMeter (Apache Foundation)** | **5.6.x** |
| **Test Coverage Monitor o Profiler** | **SonarQube (SonarSource)** | **9.x LTS** |
| **Herramienta de DBMS** | **PostgreSQL (PostgreSQL Global Development Group)** | **16.x** |
| **Herramienta auxiliar (API Testing)** | **Postman (Postman Inc.)** | **11.x** |
| **Entorno de pruebas Backend** | **Django Framework** | **5.x** |
| **Entorno de pruebas Frontend** | **Navegadores Web: Chrome / Edge / Firefox** | **Última versión** |
| **Entorno de contenedores** | **Docker Desktop** | **Última versión** |

# Recursos

## Profesionales

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Rol** | **Recursos mínimos recomendados**  **(Número de personas full-time)** | **Responsabilidades específicas / Comentarios** |
| Diseñador de casos de prueba |  Levantar la **matriz de trazabilidad** (Req ↔ UC ↔ Casos).   Diseñar casos **funcionales** por UC y **E2E (UC-00)** con datos de prueba.   Definir criterios de aceptación y de salida/entrada por suite.   Priorizar por **riesgo** y mantener el repositorio de pruebas.   Coordinar revisiones con Jefe de Taller/Recepción/Admin para validar reglas. | Responsabilidades   * Identificar, priorizar e implementar los casos de prueba. * Evaluar de forma el esfuerzo de *testing*. |
| Testeador |  Ejecutar suites (Smoke, Funcional, E2E, Regresión, RBAC, Offline).   Registrar **evidencias** (capturas, logs) y **defectos** con S/P.   Reejecutar tras correcciones; mantener avance y KPIs diarios.   Apoyar **UAT** por actor (guardia, recepción, jefe, etc.).   Medir tiempos (p95) de pasos críticos y reportarlos. | Responsabilidades:   * Ejecutar los casos de prueba. * Guardar estado de los resultados. * Recuperación de errores. * Generar peticiones de cambios en la documentación. |
| Administrador de sistema del pruebas |  Preparar y **versionar** el ambiente de **staging** (contenedores, variables).   Gestionar **snapshots/restore** y datos semilla.   Mantener herramientas (Playwright, pytest, Locust, ZAP) y reportería.   Integrar con CI (si aplica) y conservar artefactos de ejecución.   Control de accesos/credenciales del entorno de pruebas. | Responsabilidades   * Administrar el sistema de control de pruebas. * Instalar / administrar el acceso al sistema de pruebas. |
| Administrador de la base de datos / Encargado de la base de datos |  Generar y actualizar **semillas** (vehículos, talleres, usuarios, catálogos).   Ejecutar **backup/restore** (pg\_dump/pg\_restore) por iteración.   Anonimización de datos y limpieza post-pruebas.   Soporte a **índices/consultas** durante performance.   Crear usuarios/roles de BD y políticas de acceso. | Responsabilidades:   * Administra los datos del prueba (Base de Datos) * Asegurar que el entorno de datos de prueba (base de datos) y los valores que contiene son controlados y mantenidos. |
| Diseñador |  Definir arquitectura de **automatización** (Page Objects, utilidades comunes).   Seleccionar librerías y convenciones; **code review** de scripts.   Diseñar datos/fixtures reutilizables y **mocks/stubs** si se requieren.   Asegurar estabilidad (reintentos, sincronización, tagging de suites). | Responsabilidades:   * Identificar y definir las operaciones, atributos y asociaciones de las clases de prueba. * Identificar y definir los paquetes de prueba. |
| Implementador |  Desarrollar tests **UI** (Playwright) y **API** (pytest+requests).   Implementar suites de **regresión** y E2E con orquestación UC-00.   Integrar reportería (Allure u otra) y evidencias automáticas.   Mantener scripts ante cambios de la app; hardening de flakiness. | Responsabilidades:   * Implementar las clases de prueba y los paquetes de prueba. |

## Ambiente de pruebas

Se identifican los requerimientos de hardware, software y de comunicación necesarios para crear y dar soporte permanente al Ambiente de pruebas. Las actividades de instalación y configuración para el conjunto de los componentes del Ambiente de pruebas, deberán ser planificadas y calendarizadas. Se requiere que este ambiente sea seguro, estable y dedicado exclusivamente para las pruebas del sistema.

### Preparación del ambiente de pruebas

**Objetivo:** disponer de un entorno estable, seguro y trazable para la ejecución de Smoke, Funcional, E2E, UAT, Regresión, Performance y Seguridad.

**Checklist previo a cada ciclo**

* **Versionado:** build/tag de backend (pgf-api) y frontend (pgf-web) registrados en el plan de ejecución.
* **Despliegue:** docker compose up -d de pgf-api, pgf-web, pgf-db (PostgreSQL) y pgf-redis.
* **Migraciones:** ejecución de migraciones y verificación de esquemas.
* **Semillas:** carga de datos (usuarios por rol, zonas/talleres, vehículos, catálogos) y políticas (límite de adjuntos, umbral de aprobación).
* **Snap-restore:** snapshot inicial de BD y carpeta de evidencias para poder **restaurar** el entorno por iteración.
* **Smoke test:** UC más críticos (login, UC-01, UC-02, UC-03/04) en verde antes de iniciar suites.
* **Accesos:** cuentas UAT creadas para todos los actores; caducidad y reseteo controlados.
* **Seguridad básica:** variables .env protegidas, cookies seguras, cabeceras mínimas activas.
* **Monitoreo:** métricas de API/DB y logs habilitados para capturar evidencia.

**Criterio de preparación OK:** checklist completo y Smoke en verde.

### Diseño del ambiente de pruebas

**Arquitectura (staging, equivalente a producción en menor escala):**

* **Frontend (pgf-web):** app web (React/Next o similar) detrás de Nginx.
* **API (pgf-api):** backend Django/REST con autenticación JWT.
* **BD (pgf-db):** PostgreSQL (15.x) con pg\_stat\_statements habilitado para performance.
* **Cache/colas (pgf-redis):** Redis para sesiones/colas (sin datos sensibles persistentes).
* **Almacenamiento de evidencias:** volumen dedicado para imágenes (o bucket S3/MinIO en pruebas).
* **Red/TLS:** red de Docker aislada; TLS autofirmado si se requiere HTTPS en UAT.
* **Observabilidad:** logs estructurados de app, métricas básicas (CPU/RAM/latencia).

**Paridad:** mismas versiones de librerías y configuración funcional; menor tamaño de datos y réplicas.

* **HARDWARE**
* **Estación de Pruebas (Cliente)**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Procesador** | Intel Core i5 / Ryzen 5 o superior | Equipo estándar de desarrollo |
| **Memoria RAM** | 8 GB mínimo | 16 GB recomendado |
| **Espacio en Disco** | 50 GB libres | Para repositorio, logs y herramientas |
| **Tipo Monitor y Resolución** | 1080p FullHD (1920x1080) | Resolución recomendada para UI |
| **Unidad de Disquete** | No aplica | Tecnología obsoleta |
| **Tarjeta de red** | Ethernet 1 Gbps / WiFi 5 o superior | Necesario para conexión a Docker/servidor |
| **Modem** | No aplica | — |
| **Mouse** | Cualquier mouse USB/inalámbrico | — |
| **Tipo Enlace** | Ethernet / WiFi | Preferencia: cable para estabilidad |

* **Servidores**
* **Base de Datos (Pruebas)**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Procesador** | 4 vCPU (virtual) | En máquina virtual o nube |
| **Memoria RAM** | 8 GB | Soporta PostgreSQL + API |
| **Espacio en Disco** | 100 GB SSD | Datos + backups |
| **Tipo Monitor y Resolución** | No aplica | Administrado remotamente |
| **Unidad de Disquete** | No aplica | — |
| **Tarjeta de red** | Red virtual 1 Gbps | Para conexión API/DB |
| **Modem** | No aplica | — |
| **Mouse** | No aplica | — |
| **Tipo Enlace** | Conexión a Internet con IP asignada | Requisito para pruebas remotas |

* **Aplicación (Pruebas)**

*(Servidor donde corre Django + Next.js en Docker)*

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Procesador** | 4 vCPU | — |
| **Memoria RAM** | 8 GB | Recomendado para contenedores |
| **Espacio en Disco** | 100 GB SSD | — |
| **Tipo Monitor y Resolución** | No aplica | Gestionado por CLI |
| **Unidad de Disquete** | No aplica | — |
| **Tarjeta de red** | 1 Gbps virtual | — |
| **Modem** | No aplica | — |
| **Mouse** | No aplica | — |
| **Tipo Enlace** | Acceso remoto SSH / Docker | — |

* **RED**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Topología** | Cliente – Servidor vía Docker | Estándar de desarrollo |
| **Medio** | Ethernet / WiFi | — |
| **Velocidad** | 100–1000 Mbps | Recomendado 1 Gbps |
| **Protocolo** | TCP/IP | — |
| **Módems** | No aplica | — |
| **Conexión Internet** | Banda ancha estable (≥ 40 Mbps) | Necesaria para Docker pull, GitHub |

* **Sistema de Respaldo / Restauración**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Unidad (Modelo y Marca)** | Backups automatizados PostgreSQL + GitHub | Ambos sistemas guardan progreso |
| **Capacidad** | Según tamaño del proyecto (5–20 GB) | Expandible |
| **Ubicación** | Repositorio GitHub + Backups en servidor | Seguro y replicado |

* **SOFTWARE**
* **Estación de Pruebas (Cliente)**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Sistema Operativo** | Windows 10 / Windows 11 | Actualizado |
| **Herramienta de Testing** | Postman, Pytest, Playwright | Testing API, unitario y E2E |
| **Herramienta de Modelamiento** | Draw.io / Figma | Diagramas de arquitectura/flujo |
| **BDMA** | No aplica | Acceso directo vía API |
| **Browser** | Chrome / Edge / Firefox | Última versión |
| **Software de Escritorio** | VS Code | Recomendado para desarrollo |

* **Repositorios**

| **Ítem** | **Descripción** | **Observaciones** |
| --- | --- | --- |
| **Servidor** | GitHub | Control de versiones |
| **Dominio/Cuenta** | https://github.com/Diego… | Acceso del equipo |

* **Seguridad**

| **Ítem** | **Descripción** |
| --- | --- |
| **Autenticación** | Tokens JWT – Sesiones seguras |
| **Control de Accesos (RBAC)** | Roles: Guardia, Mecánico, Jefe Taller, Supervisor, Admin |
| **Backups automáticos** | Base de datos PostgreSQL |
| **Repositorio privado** | GitHub con permisos restringidos |
| **Transmisión segura** | HTTPS (cuando se despliegue en producción) |

### Diseño ambiente de pruebas

El siguiente diagrama muestra la arquitectura del Ambiente de pruebas requerido para realizar las pruebas. La arquitectura del ambiente de pruebas debe ser, en la medida de lo posible, similar a la arquitectura definida para el sistema en producción.

Diagrama

El contenido generado por IA puede ser incorrecto.

### Integración del ambiente de pruebas y configuración

Para esta actividad se requerirá la participación de profesionales de ingeniería en cuanto a instalación, configuración y puesta en marcha del Ambiente de Pruebas. Principalmente se requiere del responsable de la Red y Administración de Bases de Datos, de tal forma de obtener un ambiente lo más consistente y similar al de producción, con las bases de datos creadas y el software configurado para asegurar que el sistema funciona de acuerdo con el diseño.

Las actividades generales para considerar son:

| **Actividad** | **Responsable** | **Fecha Estimada** | **Fecha Real** | **Observaciones** |
| --- | --- | --- | --- | --- |
| Instalación del servidor de pruebas (4 vCPU, 8 GB RAM, 100 GB SSD) | Administrador Infraestructura / TI | 3/11/2025 | 3/11/2025 | Configuración inicial del entorno remoto |
| Configuración de Docker y contenedores base (API, DB, Frontend) | Equipo de Desarrollo Backend | 3/11/2025 | 3/11/2025 | Asegurar correcta comunicación entre servicios |
| Creación y configuración de la Base de Datos PostgreSQL | Administrador BD | 3/11/2025 | 3/11/2025 | Carga de tablas iniciales para pruebas |
| Configuración del entorno virtual de pruebas (Python/Django) | Equipo de Desarrollo Backend | 3/11/2025 | 3/11/2025 | Instalación de dependencias y archivos .env |
| Instalación de herramientas de testing (Postman, Pytest, Playwright) | QA / Tester | 3/11/2025 | 3/11/2025 | Validar compatibilidad con Windows 10/11 |
| Configuración de red interna entre servidor y estaciones de trabajo | Responsable de Red | 3/11/2025 | 3/11/2025 | Asegurar conectividad y puertos abiertos |
| Configuración del repositorio GitHub y sincronización con ambiente de pruebas | DevOps / Backend | 3/11/2025 | 3/11/2025 | Accesos y llaves SSH configuradas |
| Validación del ambiente de pruebas (smoke test inicial) | QA / Tester | 3/11/2025 | 3/11/2025 | Verificación de endpoints, login y BD |
| Configuración de backups automáticos de BD de pruebas | Administrador BD | 3/11/2025 | 3/11/2025 | Backup diario vía script |
| Documentación del ambiente de pruebas | QA / Analista | 3/11/2025 | 3/11/2025 | Checklist completo |

### Generación de datos

La generación de los datos para las pruebas considera los siguientes aspectos, que se deben definir de acuerdo con los requerimientos y posibilidades de obtención. Los aspectos que se describen a continuación buscan que los datos sean los correctos y que cubran todos los riesgos y situaciones necesarias.

#### Muestra de producción

Para que la muestra de datos sea realmente representativa, se deberá elegir una fecha testigo adecuada y que posibilite la mayor cobertura de datos. En este sentido se ha elegido como fecha testigo el 19/11/2025

Para la obtención de datos por esta vía, se deberán definir las restricciones (por motivos de confidencialidad) y generar algún utilitario para filtrar los datos de tal forma de obtener la mayor variabilidad de datos posible.

Además se deben considerar los siguientes aspectos para asegurar que estos datos funcionen correctamente en el Ambiente de Pruebas, si corresponde:

* Archivos Maestros al Inicio del Día
* Tablas de Parámetros
* Interfaces de Entrada
* Archivos de Movimientos del día o del periodo

Todos estos aspectos se deben considerar en los distintos ambientes donde los datos van a ser utilizados en las transacciones o actualizaciones.

**Fecha/período testigo**  
Se utilizará como muestra el período **T-1** (último mes completo previo al inicio de ejecución del plan). Ejemplo: **01–31/10/20XX**. Este período refleja operación regular con volumen suficiente de ingresos, OTs, aprobaciones y cierres.

**Criterios y restricciones de confidencialidad**

* No se migran datos personales (nombres, RUT, teléfonos, correos).
* **Patentes**: se aplicará **enmascaramiento parcial** (p. ej., ABC•123 → A*C•1*3) o **hash irreversible** manteniendo consistencia entre tablas.
* **Usuarios**: solo IDs técnicos y roles; credenciales de prueba generadas ad hoc.
* **Evidencias (imágenes)**: no se copian desde producción; se reemplazan por **placeholders** con mismo conteo y metadatos básicos (tamaño, mime).
* Archivo de mapeo (si existiera) queda fuera del entorno de pruebas.

**Obtención y preparación (procedimiento ETL)**

1. **Extracción** (producción → staging) de tablas maestras y de movimiento del período testigo mediante scripts SQL versionados.
2. **Transformación**
   * Enmascarado/anonimización de campos sensibles.
   * Recorte del período a T-1 y **submuestreo estratificado** por zona/taller para conservar distribución.
   * Sustitución de rutas de evidencias por placeholders.
3. **Carga** en staging y **verificaciones**: conteos por tabla, checks de integridad referencial y muestras aleatorias.

**Archivos/Tablas maestros al inicio del día (snapshot base)**

* VEHICULOS (ID, patente\*, zona, taller, estado).
* TALLERES, ZONAS.
* USUARIOS, ROLES, USUARIO\_ROL\_AMBITO (RBAC).
* Catálogos: SERVICIOS, REPUESTOS (incluye existencias y algunos en **stock=0**), MOTIVOS\_PAUSA, CHECKLISTS\_SEGURIDAD.
* PLANES\_PREVENTIVOS por familia de vehículo.

**Tablas de parámetros**

* CONFIG\_SISTEMA (umbral\_aprobación, límite\_adjuntos\_MB, TZ, retención\_logs).
* SLA\_ESTADOS (tiempos objetivo por estado/pausa).
* RBAC\_PARAM (reglas de alcance por rol/zona/taller).

**Interfaces de entrada (si aplican en pruebas)**

* **Importación CSV** de catálogos/usuarios (UC-06): archivos de ejemplo con 5–10 filas válidas y 2–3 con error para validar reportes.
* **Carga de evidencias**: carpeta de placeholders con distribución realista de tamaños (0.5–2 MB).

**Archivos/Tablas de movimientos del período**

* INGRESOS (UC-01/UC-07).
* RECEPCIONES y OT (UC-02).
* OT\_ESTADOS y OT\_PAUSAS (UC-03).
* ESTIMACIONES, APROBACIONES, RESERVAS\_STOCK (UC-09).
* OT\_CONSUMOS (servicios/repuestos).
* OT\_CIERRES y controles de **llaves/salida** (UC-04).
* CITAS\_PREVENTIVO (UC-10).
* **Metas de volumen mínimo** para variabilidad: ≥ **150 Ingresos**, **120 OTs**, **60** con pausas, **40** con aprobación, **40** cierres, **30** citas.

**Validaciones posteriores a la carga**

* Conteos por zona/taller dentro de ±10% del período original.
* Integridad referencial 100% (sin huérfanos).
* RBAC: al menos 2 zonas × 2 talleres con datos en cada una.
* Ejecución de **smoke** sobre UC-01/02/03/04 en verde con la muestra cargada.

Resultado esperado: un dataset **representativo y seguro** que permite ejecutar pruebas funcionales, E2E, rendimiento y seguridad sin exponer información sensible.

# Actividades e Hitos del Plan de Pruebas

* **Plan de Pruebas**

| **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** | **Observaciones** |
| --- | --- | --- | --- | --- | --- |
| Identificar el Proyecto | Analista QA | 2 horas | 29/10/2025 | 29/10/2025 | Revisión del alcance PGF |
| Definir Estrategia | Analista QA / Líder Técnico | 2 horas | 29/10/2025 | 29/10/2025 | Se define estrategia: unitarias, API, E2E |
| Estimar Actividades | Analista QA | 2 horas | 29/10/2025 | 29/10/2025 | — |
| Identificar Recursos | Jefe de Proyecto | 1 horas | 30/10/2025 | 30/10/2025 | Hardware, software y personal |
| Documentar el “Plan de Pruebas” | Analista QA | 3 horas | 3/11/2025 | 3/11/2025 | Documento oficial |
| Agendar Actividades | Jefe de Proyecto | 2 horas | 4/11/2025 | 4/11/2025 | Cronograma pruebas Sprint |
| Revisar el “Plan de Pruebas” | QA Lead / Jefe Proyecto | 2 horas | 4/11/2025 | 4/11/2025 | Validación final del documento |

* **Diseño de las Pruebas**

| **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** | **Observaciones** |
| --- | --- | --- | --- | --- | --- |
| Analizar Requerimientos | Analista QA | 2 horas | 5/11/2025 | 5/11/2025 | Se analizan RF y RNF del PGF |
| Especificar Procedimientos de Prueba | QA Tester | 2 horas | 6/11/2025 | 6/11/2025 | Incluye flujos de OT, roles y validadores |
| Especificar Casos de Prueba | QA Tester | 3 horas | 7/11/2025 | 7/11/2025 | Casos: Ingreso OT, Backup, Roles, Reportes |
| Revisar Cobertura de los Requerimientos | Analista QA | 1 horas | 10/11/2025 | 10/11/2025 | Matriz: requisito → caso de prueba |

* **Implementación de las Pruebas**

| **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** | **Observaciones** |
| --- | --- | --- | --- | --- | --- |
| Establecer Ambiente de Implementación | DevOps | 3 horas | 11/11/2025 | 11/11/2025 | Docker + BD PostgreSQL |
| Desarrollar los Procedimientos de Prueba | QA Tester | 4 horas | 12/11/2025 | 12/11/2025 | Playwright, Postman, Pytest |
| Probar y Depurar los Procedimientos | QA Tester | 4 horas | 13/11/2025 | 13/11/2025 | — |
| Modificar los Procedimientos | QA Tester | 3 horas | 14/11/2025 | 14/11/2025 | Ajustes por cambios en PGF |

* **Ejecución de las Pruebas**

| **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** | **Observaciones** |
| --- | --- | --- | --- | --- | --- |
| Ejecutar Pruebas | QA Tester | 4 horas | 14/11/2025 | 15/11/2025 | Pruebas funcionales y API |
| Comprobar Resultados Esperados | QA Tester | 2 horas | 17/11/2025 | 16/11/2025 | Comparación vs criterios de aceptación |
| Investigar Resultados Inesperados | QA Tester / Dev | 2 horas | 17/11/2025 | 17/11/2025 | — |
| Registrar Defectos (log) | QA Tester | 1 horas | 18/11/2025 | 18/11/2025 | Defectos en Jira / GitHub |
| Re-ejecutar Pruebas | QA Tester | 4 horas | 18/11/2025 | 18/11/2025 | Regression Tests |

* **Evaluación de las Pruebas**

| **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** | **Observaciones** |
| --- | --- | --- | --- | --- | --- |
| Revisar el Log de Pruebas | QA Lead | 1 horas | 19/11/2025 | 19/11/2025 | — |
| Evaluar Cobertura de Casos | Analista QA | 2 horas | 19/11/2025 | 19/11/2025 | Matriz de cobertura |
| Evaluar Defectos | QA Lead / Dev | 3 horas | 19/11/2025 | 19/11/2025 | Clasificación: crítico, mayor, menor |
| Reportar Defectos | QA Lead | 3 horas | 19/11/2025 | 19/11/2025 | Informe final de defectos |

# Entregables

## Plan de pruebas

Esta sección identifica los reportes del **Plan de Pruebas** que serán creados y distribuidos.

### Criterio de entrada para el “Plan de pruebas”

La ejecución del Plan de Pruebas podrá comenzar cuando se cumplan **todos** los siguientes puntos:

* Los requerimientos funcionales y no funcionales del proyecto PGF estén documentados y aprobados.
* Se haya completado la configuración del **Ambiente de Pruebas** (Docker, API, BD, Frontend).
* Existan datos iniciales cargados en la base de datos de pruebas.
* Los accesos estén habilitados para roles: QA, Dev, Supervisor, Jefe de Taller.
* Las herramientas de testing estén instaladas y operativas:
  + Postman
  + Pytest
  + Playwright
  + JMeter (si aplica)
* El “Plan de Pruebas” esté revisado y aprobado.

### Criterio de salida para el “Plan de pruebas”

La ejecución del Plan de Pruebas se considera completada cuando:

* Se han ejecutado **todos los casos de prueba** definidos.
* Todos los errores críticos y mayores están solucionados o mitigados.
* Los errores menores están documentados con plan de corrección.
* Los reportes de defectos y logs de pruebas están actualizados en Jira/GitHub.
* Se han realizado las pruebas de regresión posteriores a las correcciones.
* Se ha generado el informe final de resultados y aprobado por el equipo.
* La cobertura de requisitos alcanza el porcentaje establecido (ideal ≥ 90%).

### Criterio de suspensión y resumisión

La ejecución del Plan de Pruebas puede ser **suspendida** si ocurre alguno de los siguientes casos:

* **Motivos de suspensión**
* Fallas del ambiente de pruebas:
  + caída de la base de datos
  + caída del servidor API/Frontend
  + fallos en Docker
* Código inestable o no ejecutable en el ambiente.
* Cambios no planificados en los requerimientos que afecten casos clave.
* Bloqueos en datos críticos (usuarios, vehículos, OT).
* Fallos graves que impidan continuar pruebas (por ejemplo, login no funciona).
* **Criterios de reanudación**

Las pruebas se reanudarán cuando:

* El ambiente de pruebas haya sido restaurado correctamente.
* Los errores que bloqueaban la ejecución hayan sido corregidos.
* El sistema pueda volver a ejecutarse sin fallas críticas.
* QA valide que los casos de prueba pueden retomarse sin inconsistencias.
* Se actualice la versión en pruebas del proyecto PGF con un build estable.

## 6.2 Resultados de las pruebas

*[Describe los métodos y herramientas usadas para registrar y reportar los resultados y el estado de la ejecución del “Plan de pruebas”.]*

## 6.3 Reporte de defectos

*[Define los métodos y herramienta usados para registrar, ajustar y reportar los incidentes en la ejecución del “Plan de pruebas”.]*

# Anexos

## A: Tareas del proyecto

La siguiente lista muestra las tareas relacionadas con el “Plan de pruebas”:

|  |  |
| --- | --- |
| *✔* | “Plan de pruebas” (preliminar, al inicio del proyecto) |
| *✔* | Identificar requerimientos para el testing |
| *✔* | Identificar los riesgos, cuantificar impacto |
| *✔* | Desarrollar la estrategia de pruebas |
| *✔* | Identificar los recursos para las pruebas |
| *✔* | Generar “Plan de Pruebas” detallado |
| *✔* | Diseño general de las pruebas |
| *✔* | Análisis de carga |
| *✔* | Identificar y describir los casos de prueba |
| *✔* | Identificar y estructurar los procedimientos de prueba |
| *✔* | Revisar y accesar la cobertura de las pruebas |
| *✔* | Implementar las pruebas |
| *✔* | Grabar o programar los *scripts* de las pruebas, si aplica |
| *✔* | Identificar las funcionalidades a probar, específicos en el modelo de diseño e implementación |
| *✔* | Establecer el conjunto de datos externos |
| *✔* | Ejecutar las pruebas |
| *✔* | Ejecutar los procedimientos de prueba |
| *✔* | Evaluar la ejecución de las pruebas |
| *✔* | Comprobar los resultados |
| *✔* | Investigar los resultados inesperados |
| *✔* | Registro de defectos, Informe de Resultados |
| *✔* | Evaluar las pruebas |
| *✔* | Evaluar la cobertura de los casos de prueba |
| *✔* | Evaluar la cobertura del código |
| *✔* | Analizar defectos |

## B: Pruebas de rendimiento (*performance*)

Para esto, se definen las transacciones de acuerdo a los casos de uso específicos que se espera que un actor del sistema realice usando un conjunto de datos para agregar o modificar transacciones.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Comprobar la conducta de rendimiento para las transacciones seleccionadas o funcionalidades bajo las siguientes condiciones:  - Una carga de trabajo normal.  - Una sobrecarga de trabajo. |
| Técnica a usar: | Usar los procedimientos de pruebas desarrollados para el testing funcional.  Modificar los archivos de datos para aumentar las transacciones o los *script* de robotización para incrementar el número de iteraciones de cada transacción.  Los *script* deberán correr en una máquina (la mejor referencia es un solo usuario y una única transacción) y repetirla con múltiples clientes (virtuales o reales). |
| Criterio de validación: | Una Transacción / Un Usuario: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro del tiempo esperado (por transacción en forma independiente).  Múltiples Transacciones / Múltiples Usuarios: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro tiempo estimado. |
| Consideraciones especiales: | La extensión del testing de rendimiento requiere tener en *background* la carga de trabajo en el servidor.  Existen varios métodos que se pueden usar para realizar esto como por ejemplo:  Gatillar transacciones directamente al servidor, normalmente en forma de llamadas de SQL.  Crear una carga de usuarios virtuales para simular (normalmente varios cientos) los clientes. Para esto se utilizan herramientas de emulación de terminales remotas para lograr esta carga. Esta técnica también puede usarse para someter a la red a un alto tráfico.  Usar múltiples clientes físicos, cada uno corriendo los Test scripts para agregar una carga al sistema.  El testing de rendimiento debería realizarse en una máquina dedicada o en un tiempo dedicado. Esto permite un control total y una exacta medición.  Las bases de datos utilizadas para realizar el testing de rendimiento deberán ser del tamaño equivalente a las de producción o a escala similar. |
| Observaciones: | - |

## C: Pruebas de seguridad y de control de acceso

Se recomienda que el Administrador de la Red y del Sistema planifiquen algunas pruebas en este sentido.

Este *testing* se enfoca en dos áreas claves de la seguridad:

* Seguridad a nivel de la Aplicación, incluyendo acceso a los datos o funciones de negocio, y
* Seguridad a nivel del Sistema, incluyendo la autenticación (*login*) y/o acceso remoto al sistema.
* La seguridad a nivel de la aplicación asegura que, sobre la base de la seguridad deseada, se restringen a los usuarios a ciertas funciones o casos de uso específicos o se les limita el acceso a datos disponible para ellos.
* La seguridad a nivel de sistema asegura que sólo los usuarios definidos en el sistema son capaces de acceder a la aplicación y sólo a través de entradas apropiadas.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Seguridad a Nivel de Aplicación: comprobar que un usuario puede acceder sólo a las funcionalidades y datos para las cuales ese tipo de usuario tiene permiso.  Seguridad a Nivel de Sistema: comprobar que sólo esos usuarios con acceso al sistema y aplicación tienen permitido el acceso. |
| Técnica a usar: | Nivel de Aplicación: Identifique y liste cada tipo de usuario y las funcionalidades y datos de cada tipo para las cuales tiene permiso.  Cree pruebas para cada tipo de usuario y verifique cada permiso creando transacciones específicas para cada usuario.  Modifique los tipos de usuarios y vuelva a ejecutar los casos de prueba para los mismos usuarios. En cada caso verifique si las funcionalidades y los datos están correctamente disponibles o denegados.  Acceso a Nivel de Sistema: vea las consideraciones especiales más abajo. |
| Criterio de validación: | Para cada tipo de usuario conocido, las funcionalidades y los datos correctos debieran estar disponibles y todas las transacciones ejecutadas debieran ejecutarse de acuerdo a lo esperado. |
| Consideraciones especiales: | El acceso al sistema debería ser comprobado con el administrador de la red o del sistema.  Este *testing* quizás pueda requerir de la participación del administrador de la red o del sistema. |
| Observaciones: | - |