

Plan de Pruebas

Caso: Sistema de Junta de vecinos

Fecha: [18/11/2025]

Asignatura: Portafolio de Título

Integrantes:

Felipe Fuentes

—

Fecha de entrega: 21/11/2025

Profesor: Arturo Vargas

Historial de revisiones

Fecha	Versión	Descripción	Autor
15/10/2025	<1.0>	Se desarrolla el documento incluyendo todos los puntos referentes a: Introducción, Requerimiento para pruebas, Estrategia de pruebas, Recursos, Actividades, Hitos y Entregables	Felipe Fuentes
Fecha	Versión	Descripción	Autor
15/10/2025	<1.1>	Se desarrollan los últimos detalles del avance del proyecto.	Felipe Fuentes
21/11/2025	<2.0>	Se desarrollan los últimos detalles de la finalización de la fase 2.	Felipe Fuentes

TABLA DE CONTENIDO

1. Introducción	4
1.1 Resumen ejecutivo	4
1.2 Alcance de las pruebas	4
1.2.1 Elementos de pruebas	4
1.2.2 Pruebas Funcionales	5
1.2.3 Riesgos	6
1.2.3.1 Matrices de riesgos	6
1.2.3.1.1 Pruebas	6
1.3 Tipos de pruebas	7
1.3.1 Pruebas Funcionales	7
1.3.2 Herramientas involucradas	7
1.3.3 Preparación del ambiente de pruebas	8
1.3.4 Diseño del ambiente de pruebas	8
1.3.5 Diseño ambiente de pruebas	9
2. Resultados de las pruebas	10
CU-001 – Registro y Validación de Vecinos	11
CU-002 – Validación de Identidad	11
CU-004 y CU-005 – Solicitud y Emisión de Certificado de Residencia	11
CU-010 – Reserva de Espacios Comunes	11
CU-011 – Gestión de Actividades Vecinales	12
CU-008 – Notificaciones Automáticas	12
CU-016 – Generación Automática de Certificados	12
CU-006 – Postulación de Proyectos Vecinales	12
CU-007 – Flujo de Aprobación de Proyectos	13
CU-009 – Publicación de Noticias	13
CU-012 – Tablero de Solicitudes Ciudadanas	13
CU-013 – Bitácora y Auditoría	13
CU-014 – Tablero de Métricas Básicas	14
CU-015 – Chatbot Vecinal IA	14
CU-017 – OCR para Documentos de Respaldo	14
CU-018 – Resumen Automático para Directorio	14
3. Reporte de defectos	15

1. Introducción

1.1 Resumen ejecutivo

El presente documento define el Plan de Pruebas del sistema MiJuntaDigital, un proyecto orientado a la digitalización integral de los procesos de gestión vecinal, desarrollado con Django 5.2.6, PostgreSQL, Angular/Ionic y servicios externos como Webpay, N8N y Gemini.

Su objetivo principal es validar el correcto funcionamiento, estabilidad, integridad de datos y coherencia funcional de todos los módulos implementados en la versión final del sistema. Esto incluye los módulos de registro y autenticación de vecinos, certificados con QR, reservas de espacios comunales con integración Webpay, actividades vecinales, postulación y aprobación de proyectos, publicación de noticias, tablero de solicitudes, tablero de métricas, chatbot inteligente con IA, OCR de documentos y resumen automático para directorio.

El plan de pruebas busca asegurar que cada funcionalidad cumpla con los requerimientos establecidos, considerando pruebas funcionales, de integración, validación IA, flujos automáticos y manejo de errores. Con ello, se garantiza que el sistema se encuentre apto para uso real por parte del vecindario y del directorio.

1.2 Alcance de las pruebas

1.2.1 Elementos de pruebas

El alcance del plan de pruebas abarca la totalidad de los módulos implementados en la versión final de MiJuntaDigital.

Se consideran todos los flujos críticos, administrativos, automatizados e inteligentes, incluyendo interacciones con bases de datos, integraciones externas, procesos internos y respuestas generadas por IA.

Módulos incluidos en el alcance final:

- Gestión de usuarios y control de roles
- Certificados de residencia con QR y firma digital
- Reservas de espacios comunales con integración Webpay
- Actividades vecinales e inscripciones
- Notificaciones automáticas mediante N8N
- Postulación de proyectos vecinales
- Flujo de aprobación de proyectos
- Publicación de noticias
- Tablero de solicitudes ciudadanas
- Bitácora y auditoría del sistema
- Tablero de métricas básicas

- Chatbot vecinal con IA
- OCR de documentos de respaldo
- Resumen automático para el directorio

La totalidad de estos módulos fue sometida a pruebas funcionales e integrales, asegurando una validación completa del sistema.

1.2.2 Pruebas Funcionales

Las pruebas funcionales aplicadas verificaron cada módulo del sistema desde la perspectiva del usuario final, analizando su comportamiento ante entradas válidas, inválidas y escenarios excepcionales.

En esta fase final se incluyeron también pruebas sobre módulos inteligentes y automatizados, tales como:

- Respuestas generadas por el chatbot IA siguiendo reglas específicas
- Extracción OCR sobre documentos de respaldo
- Generación automática del resumen semanal para directorio
- Validación de métricas dinámicas y filtros
- Registro autónomo de eventos en la bitácora

Objetivos ampliados de la prueba:

- Validar flujos completos en todos los casos de uso CU-001 a CU-018
- Confirmar que la IA responde dentro de los límites permitidos y sin inventar funciones
- Verificar consistencia, integridad y trazabilidad en los datos generados
- Comprobar que las automatizaciones se ejecutan correctamente
- Garantizar que el sistema reaccione adecuadamente ante errores controlados

Criterios de aceptación finales:

- Todos los casos de uso deben cumplir exactamente su comportamiento esperado}
- Los módulos IA y OCR deben operar bajo condiciones normales sin fallas de consistencia
- La auditoría debe registrar todos los eventos críticos
- Las métricas deben reflejar valores reales basados en la base de datos
- Webpay sandbox debe completar transacciones sin errores
- No deben existir inconsistencias en los estados de solicitudes, proyectos o reservas

1.2.3 Riesgos

Para poder identificar los riesgos que se pueden hallar dentro del sistema a través del respectivo testing, serán identificados a través del concepto de gravedad (bajo, medio y alto), estos mostraran la importancia de cada caso de uso con el cual se hará las pruebas de testing.

1.2.3.1 Matrices de riesgos

1.2.3.1.1 Pruebas

N°	Riesgo	Descripción	Severidad	Acción Correctiva
1	Fallo en la pasarela Webpay	Interrupción de flujo de reserva y certificados	Alta	Probar con distintos tokens respuesta tx.commit()
2	Fallo en coneccion MariaDB	Pérdida temporal de conexión o datos	Alta	Implementar backups automáticos y reconexión
3	Error en generación de PDF	Certificado sin QR o sin firma	Media	Verificar rutas absolutas y permisos MEDIA_ROOT
4	Fallo en N8N	No se notifica al vecino tras evento	Media	Revisar logs de webhook
5	Interrupción de sesión	Usuario redirigido a login en mitad de flujo	Baja	Revisar middleware de sesión
6	Comportamiento inesperado del chatbot IA	Gemini podría entregar respuestas incorrectas o inventadas	Alta	Aplicar reglas de restricción y validaciones previas y fallback a KB
9	Eventos no registrados en bitácora	Acciones críticas sin trazabilidad	Alta	Asegurar llamadas obligatorias al registrar_evento()
10	Estados de solicitudes/proyectos no actualizados	Cambios no persistidos correctamente	Media	Forzar commits y validar resultados tras guardar
11	Error en resumen automático	Falta de datos o ejecución fuera de horario	Media	Manejar errores y registrar logs en auditoría

1.3 Tipos de pruebas

1.3.1 Pruebas Funcionales

Dentro del proceso de validación del sistema MiJuntaDigital, se aplicaron distintos tipos de pruebas orientadas a verificar la correcta funcionalidad, la estabilidad y la seguridad de los módulos desarrollados.

En primer lugar, se realizaron pruebas funcionales, cuyo objetivo fue comprobar que cada requerimiento definido por el usuario final se cumple correctamente. Estas pruebas se enfocaron principalmente en los módulos de registro y autenticación de vecinos, gestión de reservas, pagos a través de Webpay y generación de certificados con código QR.

Posteriormente, se ejecutaron pruebas de integración, destinadas a validar la conexión entre los distintos módulos internos del sistema y los servicios externos involucrados, como la pasarela de pago Webpay y la automatización de notificaciones mediante N8N. Con esto se buscó garantizar la correcta comunicación entre componentes y la estabilidad de los flujos transversales.

1.3.2 Herramientas involucradas

A continuación, se presentarán las diferentes herramientas que se utilizaron para probar las distintas funcionalidades que tiene el programa.

Propósito	Herramienta	Versión
Framework backend	Django	5.2.6
Lenguaje principal	Python	3.13
Motor de base de datos relacional	PostgreSQL	17.6
API bd cloud	Supabase	Última versión
Entorno de desarrollo	Visual Studio Code	1.93
Navegación y pruebas UI	Google Chrome	—
Control de versiones	Git	2x
Registro de casos de prueba y defectos	Excel	—

1.3.3 Preparación del ambiente de pruebas

El ambiente de pruebas de MiJuntaDigital se ejecuta en un entorno local de desarrollo, configurado en el mismo equipo donde se implementa y valida el sistema.

Este entorno permite realizar pruebas funcionales, de integración y de rendimiento directamente sobre la aplicación web, garantizando una revisión controlada antes del despliegue final.

Las pruebas se realizaron utilizando los módulos implementados en Django, ejecutados a través del servidor local (**runserver**), con conexión directa a la base de datos MariaDB.

Desde este entorno se validan los principales flujos del sistema: registro de vecinos, emisión de certificados con QR, reservas de espacios comunales y pagos simulados con Webpay en entorno de prueba.

1.3.4 Diseño del ambiente de pruebas

ITEM	DESCRIPCIÓN	OBSERVACIONES
HARDWARE		
Equipo principal	Notebook/ PC de escritorio	
Procesador	Intel Core i5-10400	
Memoria RAM	16 GB DDR4	
Almacenamiento	SSD de 512 GB	
Sistema Operativo	Windows 11	
Conectividad	Red doméstica con fibra óptica	
Monitor	Full HD 1080p	
Tarjeta de red	Integrada Gigabit Ethernet	

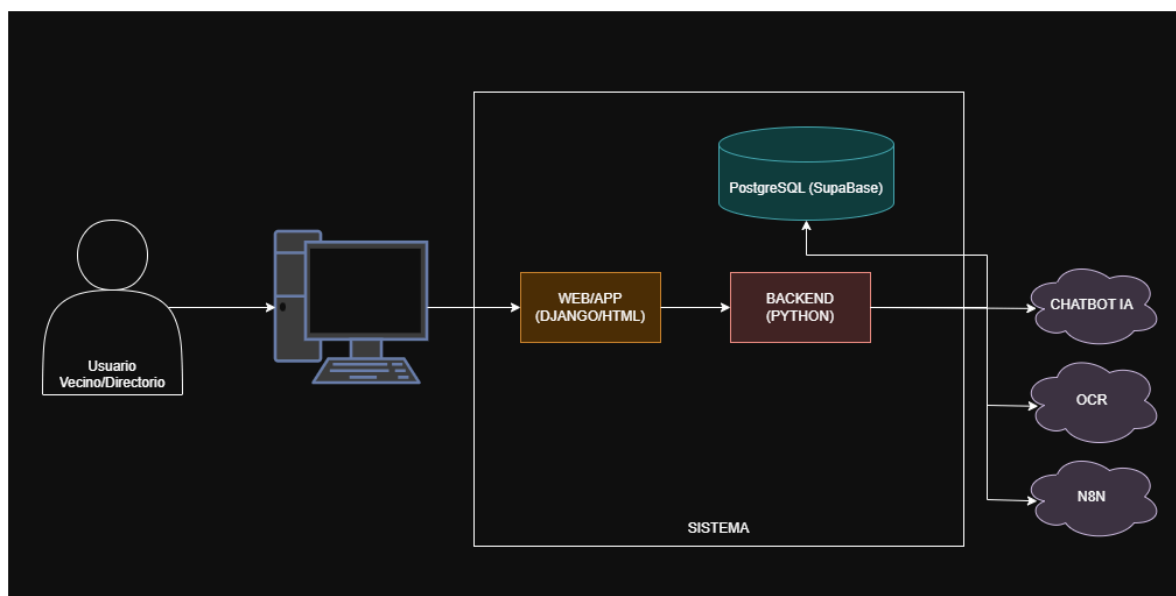
Este equipo cumple con los requerimientos necesarios para ejecutar el entorno de desarrollo y pruebas, incluyendo la base de datos local y las dependencias del proyecto Django.

ITEM	DESCRIPCIÓN	OBSERVACIONES
SOFTWARE		
Framework Backend	Django 5.2.6	
Lenguaje	Python 3.13	
Base de datos	PostgreSQL	
Entorno de desarrollo	Visual Studio Code	
Navegador web	Chrome	
Sistema de control de versiones	Git	
DEPENDENCIAS RELEVANTES		
Transbank	Webpay Test	
ReportLab	generación de certificados PDF	
N8N WebHook	automatización de notificaciones	

1.3.5 Diseño ambiente de pruebas

Para el siguiente diagrama se muestra el ambiente de pruebas requerido para la validación del sistema MiJuntaDigital, el cual representa la estructura real de comunicación entre los distintos componentes del proyecto.

La arquitectura del ambiente contempla tanto el entorno web como el backend y la base de datos, junto con las integraciones externas utilizadas durante las pruebas funcionales.



A nivel de arquitectura, a grandes rasgos, se cuenta con un servidor principal de aplicación que ejecuta el backend en Django (Python), el cual se comunica directamente con la base de datos PostgreSQL (Supabase) para almacenar y recuperar la información de vecinos, certificados, reservas y actividades.

El usuario —ya sea vecino o miembro del directorio— accede al sistema mediante un navegador web, los cuales interactúan con la capa lógica del sistema a través de peticiones HTTP.

Durante las pruebas, también se incluyeron los flujos de integración con Webpay (modo sandbox) y N8N (automatización de notificaciones), verificando su correcto funcionamiento dentro del entorno local.

Ventajas:

- Permite realizar las pruebas de forma controlada y reproducible, sin afectar datos reales.
- Facilita la depuración y monitoreo de los flujos críticos como el registro, pago y emisión de certificados.
- Mantiene comunicación directa entre el backend Django y la base de datos PostgreSQL, optimizando la velocidad de respuesta.
- La estructura modular del sistema permite aislar y validar cada componente por separado (usuarios, reservas, certificados, actividades).

Desventajas:

- El entorno local no simula la concurrencia real de múltiples usuarios accediendo simultáneamente.
- La conexión con los servicios externos (Webpay, N8N, OCR y IA) depende de la estabilidad de la red local.
- Ante actualizaciones del framework o dependencias, es necesario reconfigurar el entorno antes de ejecutar nuevamente las pruebas.

2. Resultados de las pruebas

Durante la ejecución de las pruebas se aplicó el método de caja negra, validando el cumplimiento funcional de cada caso de uso desarrollado.

Cada caso de prueba fue ejecutado manualmente desde la interfaz del sistema, verificando entradas, salidas, validaciones y comportamientos esperados, de acuerdo con los requerimientos levantados en las fases previas del proyecto.

A continuación se resumen los principales resultados obtenidos:

CU-001 – Registro y Validación de Vecinos

Paso: Completar el formulario de registro con datos válidos.

Propósito: Comprobar que un vecino puede registrarse exitosamente.

Condiciones: Campos requeridos llenos, RUN válido, contraseña segura.

Resultado esperado: El sistema guarda el registro y lo deja en estado “Pendiente”.

Resultado obtenido: Se registró correctamente, mostrando mensaje de confirmación.

Observaciones: El validador de RUN evita duplicados y muestra mensajes personalizados según estado (Pendiente, Activo, Desactivado).

CU-002 – Validación de Identidad

Paso: Aprobación o rechazo de vecinos por parte del directorio.

Propósito: Confirmar que el presidente puede activar usuarios pendientes.

Condiciones: Usuario con sesión iniciada y rol “Presidente”.

Resultado esperado: Cambio de estado de “Pendiente” a “Activo”.

Resultado obtenido: Correcto, el sistema actualizó el estado y notificó vía N8N.

Observaciones: Se validó el flujo completo de aprobación y rechazo con mensajes de confirmación.

CU-004 y CU-005 – Solicitud y Emisión de Certificado de Residencia

Paso: Generar un certificado desde el módulo de certificados.

Propósito: Validar la creación de un certificado PDF con QR y firma digital.

Condiciones: Vecino activo y solicitud aprobada.

Resultado esperado: Generación de PDF con datos correctos y folio único.

Resultado obtenido: Correcto, el documento se creó y descargó correctamente.

Observaciones: La función de expiración de 30 días y el QR redirigen correctamente al folio emitido.

CU-010 – Reserva de Espacios Comunes

Paso: Seleccionar espacio, fecha y horario, y proceder al pago.

Propósito: Verificar el flujo completo de reserva con integración Webpay. Condiciones: Vecino activo, horario disponible, conexión estable con sandbox.

Resultado esperado: Estado del pago “AUTHORIZED” y reserva creada como “Activa”.

Resultado obtenido: Correcto, los datos se guardaron en la base MariaDB y se notificó al N8N.

Observaciones: Se probó también el caso de pago fallido, redirigiendo correctamente al home con mensaje de error.

CU-011 – Gestión de Actividades Vecinales

Paso: Crear una nueva actividad con título, fecha y cupos.

Propósito: Validar que un vecino puede crear solo una actividad activa a la vez.

Condiciones: Vecino autenticado sin actividad activa previa.

Resultado esperado: Registro exitoso o mensaje de bloqueo si ya tiene una activa.

Resultado obtenido: Correcto, se bloquearon duplicados y se registró la actividad.

Observaciones: Los cupos e inscripciones se actualizaron correctamente en la base de datos.

CU-008 – Notificaciones Automáticas

Paso: Confirmar una transacción exitosa para enviar aviso.

Propósito: Validar comunicación con N8N al finalizar una reserva o emisión.

Condiciones: Webhook configurado y en escucha.

Resultado esperado: Mensaje recibido por N8N con datos del evento.

Resultado obtenido: Correcto, las notificaciones se enviaron sin errores.

Observaciones: El log de N8N registró correctamente los envíos y respuestas HTTP 200.

CU-016 – Generación Automática de Certificados

Paso: Emitir certificado para un vecino con solicitud aprobada.

Propósito: Validar la creación automática y su estado de vigencia.

Condiciones: Certificado existente con más de 30 días o nuevo.

Resultado esperado: Certificado nuevo en estado “Emitido” o marcado como “Vencido”.

Resultado obtenido: Correcto, el sistema actualizó los estados de forma automática.

Observaciones: Las fechas y estados se reflejan correctamente en el módulo de certificados.

CU-006 – Postulación de Proyectos Vecinales

Paso: Envío del formulario de postulación con datos, presupuesto y documentos adjuntos.

Propósito: Verificar el ingreso correcto de postulaciones y su almacenamiento en estado “En revisión”.

Condiciones: El vecino debe estar autenticado, contar con todos los campos obligatorios completos y adjuntar documentos válidos en formatos permitidos.

Resultado esperado: Registro válido, documentos asociados y notificación al directorio.

Resultado obtenido: Se generó correctamente la postulación y quedó disponible para revisión. Las validaciones de campos obligatorios funcionaron según lo esperado.

Observaciones: No se permitió enviar el formulario con archivos o campos inválidos.

CU-007 – Flujo de Aprobación de Proyectos

Paso: El directorio revisa, aprueba o rechaza una postulación con comentarios.

Propósito: Validar la actualización del estado del proyecto y la comunicación hacia el vecino.

Resultado esperado: Estado actualizado a “Aprobado” o “Rechazado”, notificación enviada al usuario.

Condiciones: El miembro del directorio debe tener sesión activa con rol autorizado (presidente, secretario o tesorero) y existir una postulación en estado “En revisión”.

Resultado obtenido: El flujo se completó correctamente. Los cambios se reflejaron en el panel y se registró el evento en la bitácora.

Observaciones: La interfaz mostró adecuadamente los proyectos aprobados/rechazados.

CU-009 – Publicación de Noticias

Paso: Creación y publicación de una noticia con imagen y texto.

Propósito: Comprobar que los comunicados puedan publicarse correctamente.

Condiciones: El usuario debe ser parte del directorio con permisos de publicación y las imágenes deben cumplir con los formatos permitidos.

Resultado esperado: La noticia aparece en la sección pública.

Resultado obtenido: La publicación se mostró de forma inmediata, con la imagen y formato correcto.

Observaciones: Las validaciones de formato de archivo se ejecutaron correctamente.

CU-012 – Tablero de Solicitudes Ciudadanas

Paso: Envío de solicitud de luminarias/aseo y actualización de estado por el directorio.

Propósito: Confirmar que el flujo completo desde creación a resolución funciona sin errores.

Condiciones: El vecino debe estar autenticado para crear solicitudes y el directorio debe contar con rol válido para actualizar estados.

Resultado esperado: Solicitud registrada, visible en el tablero, con estados actualizables.

Resultado obtenido: Las solicitudes se visualizan correctamente y los cambios de estado se reflejaron sin inconsistencias.

Observaciones: Las notificaciones al vecino funcionaron oportunamente.

CU-013 – Bitácora y Auditoría

Paso: Ejecución de acciones críticas (registro, aprobación, pagos) y posterior revisión de bitácora.

Propósito: Asegurar la trazabilidad completa del sistema.

Condiciones: Debe ejecutarse una acción crítica válida (aprobación, reserva, certificado, actualización de estado) y la base de datos debe estar operativa para registrar el evento.

Resultado esperado: Registro adecuado del usuario, acción, fecha y resultado.

Resultado obtenido: Todos los eventos críticos quedaron almacenados y filtrables.

Observaciones: No se detectaron saltos o eventos faltantes.

CU-014 – Tablero de Métricas Básicas

Paso: Carga de métricas (certificados, reservas, socios activos) y filtrado por fecha.

Propósito: Verificar cálculos dinámicos y correcta visualización.

Condiciones: El usuario debe tener rol del directorio y deben existir datos registrados para calcular métricas o mostrar advertencias en caso contrario.

Resultado esperado: Valores consistentes con la base de datos.

Resultado obtenido: Las métricas cargaron correctamente y respondieron a los filtros en tiempo real.

Observaciones: Se validó ausencia de errores cuando no había datos disponibles.

CU-015 – Chatbot Vecinal IA

Paso: Consultas reales sobre actividades, reservas, solicitudes y comandos especiales (“ver actividades”, “mis reservas”, “reservar ...”).

Propósito: Validar respuestas IA dentro de los límites funcionales definidos.

Condiciones: El chatbot debe estar operativo, las reglas de prompts activas y el usuario autenticado según corresponda para acceder a información personalizada.

Resultado esperado: Respuestas claras, sin inventar funciones y respetando los roles del usuario.

Resultado obtenido: El chatbot respondió correctamente y ejecutó acciones cuando se usaron los triggers pertinentes.

Observaciones: El fallback a KB se activó correctamente cuando la IA no pudo procesar la respuesta.

CU-017 – OCR para Documentos de Respaldo

Paso: Carga de PDF/imágenes, ejecución del OCR y validación de datos extraídos.

Propósito: Confirmar que el sistema pueda leer documentos escaneados.

Condiciones: Debe subirse un documento válido (Imagen) y el servicio de OCR debe estar disponible para procesar el archivo.

Resultado esperado: Texto extraído de forma correcta o mensaje de error para documentos ilegibles.

Resultado obtenido: La extracción funcionó correctamente en documentos nítidos y detectó fallas en archivos de baja calidad.

Observaciones: El sistema manejó adecuadamente los formatos no permitidos.

CU-018 – Resumen Automático para Directorio

Paso probado: Generación del informe semanal y envío al directorio.

Propósito: Validar la consolidación de certificados, solicitudes y reservas.

Condiciones: Deben existir datos recientes (certificados, solicitudes y reservas) y el proceso automático debe ejecutarse con conectividad hacia todos los módulos involucrados.

Resultado esperado: Informe claro con KPIs y métricas correctas.

Resultado obtenido: El resumen se generó sin errores y se envió correctamente al correo/tablero del directorio.

Observaciones: El proceso quedó registrado en la bitácora.

En general, todas las pruebas funcionales, de integración e interfaz fueron aprobadas, cumpliendo con los objetivos definidos en el plan de trabajo.

Las validaciones de formularios, autenticación, control de roles y transacciones fueron verificadas exitosamente, garantizando la estabilidad de los módulos implementados.

3. Reporte de defectos

Durante la ejecución de los casos de prueba definidos en el plan, no se detectaron defectos críticos ni incidencias funcionales en los módulos desarrollados.

El sistema *MiJuntaDigital* respondió conforme a los resultados esperados en cada caso de uso, manteniendo estabilidad en el flujo de operaciones, las integraciones externas y la persistencia de datos en la base de datos PostgreSQL.