

Plan de Pruebas

Caso: Sistema de Junta de vecinos

Fecha: [16/10/2025]

Asignatura: Portafolio de Título

Integrantes:

Felipe Fuentes

—

Fecha de entrega: 17/10/2025

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

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	5
1.2.3.1 Matrices de riesgos	6
1.2.3.1.1 Pruebas	6
1.3 Tipos de pruebas	6
1.3.1 Pruebas Funcionales	6
1.3.2 Herramientas involucradas	7
1.3.3 Preparación del ambiente de pruebas	7
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	10
CU-002 – Validación de Identidad	10
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	11
CU-008 – Notificaciones Automáticas	11
CU-016 – Generación Automática de Certificados	12
3. Reporte de defectos	12

1. Introducción

1.1 Resumen ejecutivo

El presente documento define el Plan de Pruebas del sistema MiJuntaDigital, un proyecto orientado a la digitalización de los procesos de gestión vecinal, desarrollado con Django 5.2.6, MariaDB y un entorno web responsivo.

Su objetivo principal es verificar el correcto funcionamiento, estabilidad y usabilidad de los módulos implementados hasta la fecha: gestión de usuarios, emisión de certificados con QR, reserva de espacios comunales con integración Webpay, y módulo de actividades vecinales.

El plan de pruebas busca asegurar la calidad del software mediante pruebas funcionales y de integración aplicadas en un entorno controlado, garantizando que el sistema cumpla con los requerimientos definidos durante la fase de análisis.

1.2 Alcance de las pruebas

1.2.1 Elementos de pruebas

El alcance del plan se limita a los módulos ya implementados en la versión actual del sistema. Las pruebas se centran en verificar la correcta ejecución de las funcionalidades críticas, sin incluir módulos aún en desarrollo (por ejemplo: Chatbot con IA, OCR, tablero de métricas).

Módulos considerados en las pruebas:

1. Gestión de Usuarios: registro, validación, inicio de sesión y control de roles.
2. Certificados de Residencia: emisión automática con QR y firma digital.
3. Reservas de Espacios Comunales: flujo completo de reserva y pago Webpay.
4. Actividades Vecinales: creación, inscripción y finalización de actividades.
5. Integraciones externas (N8N): notificación automática tras confirmación de pago.

Los resultados esperados son la validación funcional y la detección de posibles defectos en cada módulo.

1.2.2 Pruebas Funcionales

Se aplicará la técnica de caja negra, verificando los resultados del sistema frente a entradas válidas e inválidas, sin considerar la lógica interna del código.

Objetivos de la prueba:

- Comprobar que cada flujo de usuario responde según lo definido en los requerimientos.
- Validar que los mensajes de error sean claros y coherentes.
- Garantizar que las operaciones críticas (como pagos o generación de documentos) finalicen correctamente.

Criterios de aceptación:

- Todos los flujos principales completados sin errores.
- Las transacciones exitosas reflejan cambios correctos en la base de datos.
- Las notificaciones automáticas se envían y registran correctamente.

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

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	MariaDB	11.8.3-GA
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	

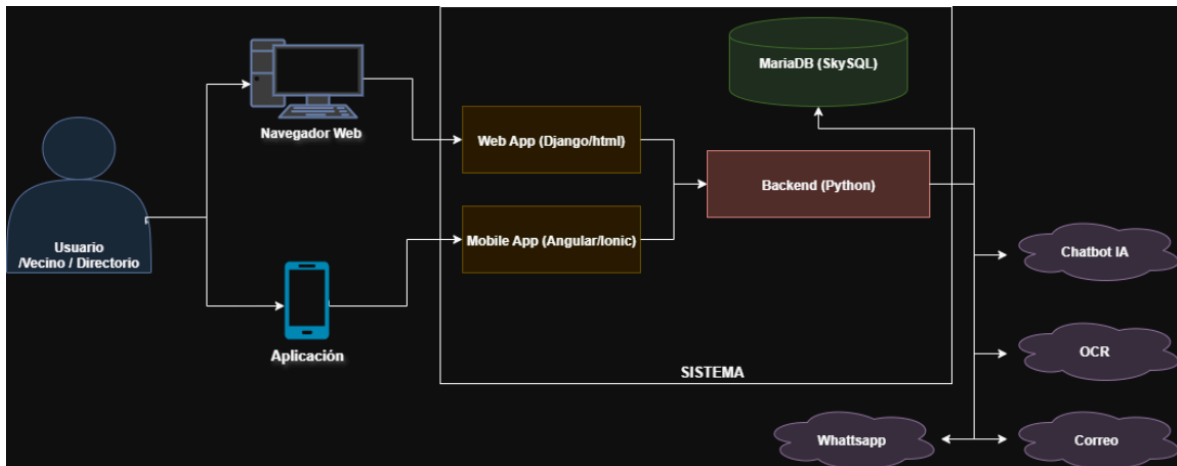
Este equipo cumple con los requerimientos necesarios para ejecutar el entorno de desarrollo y pruebas, incluyendo la base de datos local y las dependencias del proyecto Django.

ITEM	DESCRIPCIÓN	OBSERVACIONES
SOFTWARE		
Framework Backend	Django 5.2.6	
Lenguaje	Python 3.13	
Base de datos	MariaDB 10.11	
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 MariaDB (SkySQL) 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 o aplicación móvil (Angular/Ionic), 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 MariaDB, 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 y N8N) depende de la estabilidad de la red local.
- No se incluyeron en esta fase las integraciones futuras (Chatbot IA, OCR, mensajería), por lo que no fueron probadas.
- 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.

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 MariaDB.