**Índice**

[1 Identificación del Documento 5](#_Toc200477035)

[1.1 Nombre del Proyecto 5](#_Toc200477036)

[1.2 Nombre del Documento 5](#_Toc200477037)

[1.3 Versión 5](#_Toc200477038)

[1.4 Fecha 5](#_Toc200477039)

[1.5 Responsables 5](#_Toc200477040)

[1.6 Aprobación 5](#_Toc200477041)

[2 Introducción 6](#_Toc200477042)

[3 Alcance 7](#_Toc200477043)

[3.1 Componentes que Serán Probados 7](#_Toc200477044)

[3.2 Componentes Fuera del Alcance 8](#_Toc200477045)

[3.3 Supuestos y Dependencias 9](#_Toc200477046)

[3.4 Restricciones 10](#_Toc200477047)

[4 Objetivos de las Pruebas 12](#_Toc200477048)

[4.1 Validaciones Clave 12](#_Toc200477049)

[4.2 Estabilidad del Sistema 13](#_Toc200477050)

[4.2.1 Pruebas de resistencia ante múltiples usuarios 13](#_Toc200477051)

[4.2.2 Manejo de errores y aislamiento de fallos 13](#_Toc200477052)

[4.2.3 Protección ante entradas inválidas y condiciones inesperadas 14](#_Toc200477053)

[4.2.4 Pruebas de usabilidad tras interrupciones 14](#_Toc200477054)

[4.2.5 Verificación de integridad ante fallos temporales de red 14](#_Toc200477055)

[4.3 Rendimiento Bajo Condiciones Esperadas 15](#_Toc200477056)

[4.3.1 Tiempo de respuesta del sistema 15](#_Toc200477057)

[4.3.2 Rendimiento bajo carga moderada 15](#_Toc200477058)

[4.3.3 Consumo de recursos del backend 15](#_Toc200477059)

[4.3.4 Optimización en frontend 16](#_Toc200477060)

[4.3.5 Eficiencia en operaciones en segundo plano 16](#_Toc200477061)

[5 Criterios de Aceptación 17](#_Toc200477062)

[5.1 Criterios de Entrada 17](#_Toc200477063)

[5.2 Criterios de Salida 18](#_Toc200477064)

[6 Tipos de Pruebas 19](#_Toc200477065)

[6.1 Pruebas Unitarias 19](#_Toc200477066)

[6.2 Pruebas de Integración 20](#_Toc200477067)

[6.3 Pruebas del Sistema 21](#_Toc200477068)

[6.4 Pruebas de Aceptación 21](#_Toc200477069)

[6.5 Pruebas de Rendimiento 22](#_Toc200477070)

[7 Cronograma 23](#_Toc200477071)

[7.1 Diseño de Pruebas 23](#_Toc200477072)

[7.2 Ejecución de Pruebas 24](#_Toc200477073)

[7.3 Elaboración del Informe Final 25](#_Toc200477074)

[8 Roles y Responsabilidades 27](#_Toc200477075)

[8.1 QA Lead 27](#_Toc200477076)

[8.2 Tester 27](#_Toc200477077)

[8.3 Desarrolladores 27](#_Toc200477078)

[8.4 Product Owner 27](#_Toc200477079)

[9 Riesgos y Mitigaciones 28](#_Toc200477080)

[10 Aprobaciones 29](#_Toc200477081)

[10.1 QA Leader 29](#_Toc200477082)

[10.2 Jefe del Proyecto 29](#_Toc200477083)

[10.3 Product Owner 29](#_Toc200477084)

# Identificación del Documento

## Nombre del Proyecto

## Nombre del Documento

## Versión

## Fecha

## Responsables

## Aprobación

# Introducción

El presente Plan de Pruebas tiene como propósito definir la estrategia, alcance y lineamientos para validar la calidad del sistema de vinculación académica y empresarial desarrollado por estudiantes del Instituto Tecnológico de Costa Rica. Este documento establece los criterios que guiarán la ejecución de pruebas funcionales, no funcionales y de aceptación, asegurando que el sistema cumpla con los requerimientos especificados y entregue una experiencia confiable a todos los usuarios finales.

El sistema en cuestión es una plataforma web diseñada para facilitar la conexión entre estudiantes, egresados y empresas mediante la publicación y gestión de oportunidades de Trabajo Final de Graduación (TFG), pasantías y ofertas laborales. Incluye funcionalidades clave como el registro y autenticación de usuarios, postulación a vacantes, generación de informes estadísticos y coordinación de citas entre empresas y administradores académicos.

El sistema ha sido construido sobre el *stack* tecnológico MERN (MongoDB, Express.js, React, Node.js), con enfoque modular y principios de escalabilidad, seguridad y mantenibilidad. Las pruebas abarcarán todos los módulos del sistema (Autenticación, Gestión Empresarial, Estudiantes y Egresados, Administración TFG, Vinculación con Empresas, y Gestión de Archivos), y se enfocarán en verificar la integridad de los flujos críticos del sistema, la consistencia de los datos, la correcta asignación de roles y permisos, y el cumplimiento de los requisitos legales y técnicos establecidos.

Este plan servirá como guía operativa para el equipo de pruebas, definiendo los procedimientos, herramientas, métricas y criterios de aceptación necesarios para validar que el sistema está listo para su despliegue en entornos reales.

# Alcance

## Componentes que Serán Probados

El plan de pruebas abarca todos los módulos funcionales descritos en los documentos de requisitos y arquitectura, asegurando que cada componente cumpla con su propósito y opere de forma adecuada en el sistema de vinculación académica y empresarial. Las pruebas incluirán tanto las funcionalidades visibles para el usuario como los procesos internos del sistema.

A continuación, se detallan los componentes que serán objeto de prueba:

* **Módulo de Autenticación (AU):** Validación de los flujos de registro, inicio de sesión, recuperación de contraseña y control de acceso basado en roles (estudiante, egresado, empresa, administrador).
* **Módulo Empresarial (EM):** Publicación de oportunidades, gestión de estados de flyers, visualización de postulantes y estadísticas, así como funcionalidades de interacción empresarial.
* **Módulo de Estudiantes y Egresados (EE):** Exploración de oportunidades, postulación, marcación de interés y gestión de datos personales.
* **Módulo de Administración TFG (AT):** Aprobación de publicaciones de TFG, actualización de estado de usuarios (de estudiante a egresado), generación de informes estadísticos y gestión de usuarios.
* **Módulo de Vinculación con Empresas (AV):** Gestión de registros empresariales, generación de reportes de interacción y supervisión de la plataforma desde el rol de vinculación.
* **Módulo de Gestión de Archivos (AG):** Generación y descarga de documentos (PDFs y flyers), validación de formatos y control de acceso a archivos.
* **Componentes transversales a varios módulos:**
  + Autenticación y autorización basada en JWT.
  + Generación dinámica de documentos PDF.
  + Visualización de estadísticas mediante gráficos interactivos (Chart.js).
  + Internacionalización.

Se realizarán pruebas específicas según el tipo de usuario definido en el sistema (estudiante, egresado, empresa, administrador TFG, administrador de vinculación), para asegurar que cada uno tenga acceso a las funcionalidades adecuadas conforme a su perfil.

Nota: Los flujos de sincronización con calendarios externos (como Google Calendar u Outlook) no serán cubiertos en esta etapa de pruebas debido a limitaciones de tiempo en el cronograma actual. Esta funcionalidad se considera como futura línea de validación para pruebas complementarias o en entornos posteriores.

## Componentes Fuera del Alcance

Aunque el presente plan busca cubrir la mayor parte del sistema, existen ciertas funcionalidades o características que quedarán excluidas del proceso formal de pruebas en esta etapa, debido a limitaciones de tiempo, herramientas o dependencias externas no disponibles. Estos componentes podrán ser abordados en fases posteriores o pruebas especializadas.

Los elementos fuera del alcance son los siguientes:

* **Sincronización con calendarios externos (Google Calendar, Outlook):** Debido a restricciones de tiempo y complejidad en la integración con servicios de terceros, no se realizará la validación de sincronización bidireccional de agendas. Esta funcionalidad se considera para futuras pruebas específicas.
* **Soporte para accesibilidad conforme a WCAG 2.1:** Si bien el sistema está diseñado con una interfaz limpia y minimalista, no se ejecutarán pruebas formales para validar el cumplimiento específico de las directrices de accesibilidad web.
* **Personalización visual o cambio de temas:** Las opciones de personalización estética, si llegaran a incorporarse, no serán consideradas en esta fase de pruebas por no ser parte del núcleo funcional del sistema.
* **Soporte multibrowser o pruebas en dispositivos móviles:** Las pruebas se realizarán en un entorno de escritorio estándar (última versión de Chrome o Firefox). No se evaluará el comportamiento en otros navegadores (Edge, Safari) ni en dispositivos móviles, a pesar de que la interfaz sea responsiva.
* **Integraciones técnicas aún no disponibles**: Se excluirán aquellas funcionalidades que dependan de servicios de terceros o APIs externas aún en fase de integración o pruebas aisladas. Esto incluye, entre otros, servicios de verificación externa y plataformas complementarias.

Estas exclusiones permitirán focalizar los esfuerzos del equipo en la validación completa de los flujos funcionales clave para la entrega del producto.

## Supuestos y Dependencias

La ejecución del plan de pruebas se fundamenta en una serie de supuestos y dependencias técnicas y organizacionales que deben cumplirse para garantizar la validez y continuidad del proceso. A continuación, se detallan los principales:

* **Ambientes de desarrollo y producción disponibles**: Se asume que el sistema estará correctamente desplegado en los entornos definidos (desarrollo y producción en Render), según lo descrito en el documento de arquitectura (SAD). Esto incluye servicios backend (Node.js + Express), frontend (React), y base de datos (MongoDB Atlas).
* **Infraestructura del stack MERN funcional:** Las pruebas se realizarán bajo la suposición de que todos los componentes del stack tecnológico (MongoDB, Express.js, React, Node.js) se encuentran correctamente instalados, configurados y operativos en los entornos correspondientes.
* **Acceso completo a roles de usuario:** Se dará por sentado que los testers tendrán acceso a cuentas activas con cada uno de los perfiles definidos: estudiante, egresado, empresa, administrador TFG y administrador de vinculación.
* **Funcionalidades listas para prueba:** Todas las funcionalidades deben estar desarrolladas e integradas de forma estable antes del inicio de las pruebas. Se asume que no existen errores críticos abiertos que impidan el uso de los módulos principales.
* **Disponibilidad del equipo técnico:** Los miembros responsables de pruebas, desarrollo y revisión estarán disponibles durante las fechas estipuladas para ejecutar las actividades y corregir errores oportunamente.
* **Sin cambios mayores durante el proceso:** Se asume que no se introducirán modificaciones significativas en los requisitos funcionales o en la arquitectura del sistema durante el ciclo de pruebas, salvo situaciones críticas previamente aprobadas.

Estas condiciones constituyen la base para el desarrollo de las pruebas y su cumplimiento es esencial para alcanzar resultados confiables y eficientes.

## Restricciones

Durante la planificación y ejecución de las pruebas del sistema, se identifican una serie de restricciones que limitan el alcance, la duración y los recursos disponibles para esta fase del proyecto. Estas restricciones deben ser consideradas para definir expectativas realistas respecto al proceso de validación.

* **Límite de tiempo definido por calendario académico:** Todas las pruebas deberán estar finalizadas antes del 3 de julio de 2025, fecha máxima de entrega de actas para el semestre en curso. Esto impone una ventana temporal estricta para ejecutar pruebas completas, gestionar errores y generar informes.
* **Limitación al uso de herramientas gratuitas:** Por políticas del proyecto, se evitará el uso de herramientas de pago durante el desarrollo y ejecución de las pruebas. Esto restringe el acceso a plataformas especializadas que podrían facilitar pruebas automatizadas, seguimiento de errores o simulaciones de carga.
* **Recursos de infraestructura limitados:** Las pruebas se realizarán en entornos desplegados sobre la plataforma Render, los cuales cuentan con recursos acotados en cuanto a capacidad de procesamiento, almacenamiento y concurrencia. Esto puede afectar especialmente las pruebas de rendimiento o escalabilidad.
* **Automatización sujeta a herramientas disponibles:** Aunque se contempla la posibilidad de automatizar ciertos escenarios, la viabilidad de esta automatización dependerá del tipo de pruebas y de las herramientas compatibles con el stack tecnológico del proyecto. En caso de no contar con soporte adecuado, algunas pruebas se ejecutarán de forma manual.

Estas restricciones no impiden la validación integral del sistema, pero condicionan la forma, alcance y profundidad con la que se abordarán ciertos tipos de prueba.

# Objetivos de las Pruebas

## Validaciones Clave

El sistema de vinculación académica y empresarial incorpora múltiples funcionalidades orientadas a facilitar la interacción entre estudiantes, egresados, empresas y administradores académicos. Para asegurar que los objetivos principales del sistema se cumplan, se validarán prioritariamente las siguientes funcionalidades críticas:

* **Autenticación y control de acceso:** Validación del sistema de inicio de sesión con credenciales, control de acceso mediante roles, y recuperación de contraseña. Esta funcionalidad es fundamental para garantizar la seguridad y la personalización de las vistas y acciones por perfil de usuario.
* **Postulación y registro de interés (Estudiantes/Egresados):** Se validará la navegación de oportunidades, postulación a vacantes disponibles, marcación de interés en flyers y actualización del estado de participación. Estos flujos son esenciales para garantizar la funcionalidad principal desde el punto de vista del usuario final.
* **Publicación y gestión de oportunidades (Empresas):** Se probarán los procesos de creación de oportunidades, carga de flyers, edición y eliminación de vacantes. Además, se verificará la visualización del historial de postulaciones y la comunicación con los administradores mediante solicitudes de cita.
* **Aprobación y gestión académica (Administradores TFG):** Validación de flujos como la revisión y aprobación de oportunidades, gestión de usuarios (transición de estado entre estudiante y egresado), y generación de reportes académicos relevantes para la institución.
* **Supervisión y control institucional (Administrador de Vinculación):** Se validarán tareas como la aprobación de cuentas empresariales, visualización de datos agregados, administración de agenda y supervisión general del sistema para fines de trazabilidad y control de calidad institucional.
* **Procesos transversales críticos:**
  + Generación de documentos en formato PDF para informes y flyers.
  + Visualización de métricas y estadísticas usando gráficos interactivos.
  + Flujo de notificaciones automáticas.
  + Control de cambios visibles para auditoría de acciones administrativas.

## Estabilidad del Sistema

Esta sección valida que el sistema se mantenga confiable, consistente y resistente ante condiciones adversas, sin comprometer su funcionamiento general.

### Pruebas de resistencia ante múltiples usuarios

Aunque no se cuenta con métricas precisas (como la cantidad de estudiantes practicantes en el TEC) para dimensionar un esquema de carga, se simularán escenarios críticos (por ejemplo, 50–100 usuarios concurrentes) para comprobar que el sistema no presenta caídas ni errores graves bajo uso intensivo.

### Manejo de errores y aislamiento de fallos

* Se probará que, al presentarse errores en componentes individuales (falla en la carga de un gráfico, error al generar un PDF, desconexión breve), estos sean capturados localmente y no provoquen fallos globales.
* En React, se utilizarán Error Boundaries para recubrimiento en componentes críticos (por ejemplo, vistas de postulaciones o estadísticas) que permitan renderizar interfaces alternativas en caso de excepción.
* En Express, se validará el uso de un middleware general de captura de errores, que impida que excepciones no controladas colapsen el servidor y que los errores se manejen de forma segura.

### Protección ante entradas inválidas y condiciones inesperadas

Se evaluará la estabilidad del sistema al recibir entradas corruptas (e.g., formularios llenados parcialmente, cambios de estado incorrectos), asegurando que el servidor responda con mensajes HTTP adecuados y no se produzcan pérdidas de estado ni bloqueos de la aplicación.

### Pruebas de usabilidad tras interrupciones

Se simularán recargas forzadas o cierres de sesión repentino durante actividades clave (como generación de informes o carga de flyers) para verificar que el sistema pueda recuperar la sesión o guarde el progreso sin corrupción de datos.

### Verificación de integridad ante fallos temporales de red

Se comprobará que ante conexiones intermitentes (por ejemplo, fetch de datos JSON que falla) el frontend mantenga la experiencia del usuario y que el backend no se vea afectado globalmente; los errores deben registrarse y mostrarse de manera controlada.

## Rendimiento Bajo Condiciones Esperadas

Uno de los objetivos fundamentales del plan de pruebas es asegurar que el sistema responda adecuadamente bajo condiciones normales de uso, sin afectar la experiencia del usuario ni comprometer la eficiencia operativa. Para ello, se evaluarán distintos aspectos del rendimiento en escenarios representativos:

### Tiempo de respuesta del sistema

Se medirá el tiempo que tarda la plataforma en responder a acciones clave, como autenticación, postulación, carga de publicaciones y visualización de dashboards. El objetivo es que:

* Las respuestas del servidor se mantengan por debajo de 600 ms para operaciones estándar.
* Las vistas en React carguen en menos de 2 segundos para usuarios promedio en conexión de banda ancha.

### Rendimiento bajo carga moderada

Simulando entre 30 y 50 usuarios concurrentes, se evaluará que:

* No haya caídas de la API.
* El sistema mantenga su capacidad de generar documentos (PDF, Excel) sin cuellos de botella.
* El acceso concurrente a módulos críticos (postulación, visualización de interesados, generación de informes) no degrade la experiencia general más allá de umbrales aceptables.

### Consumo de recursos del backend

Se observará el uso de CPU y memoria en el entorno Render durante operaciones como:

* Generación de PDFs (con pdfkit).
* Consultas filtradas en MongoDB (uso de índices).
* Renderizado de estadísticas con gran volumen de datos.

Se espera que:

* El backend no supere el 70% de uso sostenido de CPU en condiciones normales.
* El consumo de memoria se mantenga debajo de 512 MB para operaciones habituales.

### Optimización en frontend

Se revisará que:

* El uso de React Router permita transiciones eficientes sin recarga de página.
* Las gráficas (Chart.js) se carguen asincrónicamente y no bloqueen la interfaz.
* No existan fugas de memoria por renderizado ineficiente o hooks mal gestionados.

### Eficiencia en operaciones en segundo plano

Se comprobará que tareas como:

* Envío de correos automáticos (con Nodemailer),
* Notificaciones de estado,
* y validaciones de sesiones,

se ejecuten de manera asincrónica sin afectar el hilo principal del servidor.

# Criterios de Aceptación

## Criterios de Entrada

Antes de iniciar la fase formal de pruebas, deben cumplirse una serie de condiciones mínimas que garantizan la viabilidad, utilidad y estabilidad del proceso. Estas condiciones de entrada aseguran que el sistema se encuentre en un estado funcional y documentado, listo para ser evaluado de manera efectiva:

* **Funcionalidades completas:** Todas las funcionalidades planificadas para esta fase deben estar completamente implementadas, sin características críticas pendientes por desarrollar.
* **Entorno de despliegue disponible:** El sistema debe estar correctamente desplegado en un entorno funcional (desarrollo o producción en Render), según la naturaleza de la prueba a ejecutar (unitaria, integración, aceptación, etc.).
* **Estabilidad básica verificada (Smoke Testing):** El equipo de desarrollo debe haber ejecutado pruebas preliminares que confirmen el funcionamiento esencial de los módulos principales, evitando pruebas sobre un sistema inestable.
* **Roles de usuario disponibles:** Deben existir cuentas activas para todos los perfiles de usuario (estudiante, egresado, empresa, administrador TFG y administrador de vinculación), para validar funcionalidades específicas por rol.
* **Documentación base aprobada:** La Especificación de Requisitos del Software (ERS) y el Documento de Arquitectura (SAD) deben estar finalizados y aprobados, sirviendo como base formal para la elaboración de los casos de prueba.
* **Datos de prueba disponibles cuando aplique:** Para pruebas que lo requieran, deben estar previamente cargados los datos necesarios (publicaciones simuladas, flyers, registros ficticios de usuarios, etc.), preparados conforme a los escenarios definidos.

Estas condiciones establecen el punto de partida mínimo aceptable para ejecutar pruebas controladas y representativas del uso real del sistema.

## Criterios de Salida

Los criterios de salida definen las condiciones que deben cumplirse para considerar finalizada la fase de pruebas y validar que el sistema se encuentra en un estado aceptable para su entrega o despliegue. Para este proyecto, se han establecido los siguientes criterios:

* **Cobertura mínima de pruebas alcanzada:** Al menos el 90% de los casos de prueba definidos deben haber sido ejecutados exitosamente, sin fallos. El porcentaje restante podrá incluir casos descartados o pospuestos por justificación técnica o funcional.
* **Errores críticos controlados o documentados:** No es estrictamente necesario que todos los errores críticos estén corregidos al cierre de la fase, siempre y cuando se documenten y no afecten funcionalidades fundamentales del sistema. Su resolución puede ser incluida en futuras iteraciones.
* **Permisión de errores menores:** Se permite que queden errores de baja prioridad abiertos, siempre que no comprometan la integridad de los datos, la seguridad del sistema ni la experiencia general del usuario.
* **Aprobación formal del equipo de QA o patrocinadora:** La finalización de pruebas deberá contar con la validación del líder de calidad o, en su defecto, la aprobación de la patrocinadora del proyecto, quien confirmará que se han cumplido los objetivos establecidos en el plan.
* **Elaboración de informe final:** Se deberá generar un informe que documente los resultados de las pruebas, incluyendo métricas clave, casos exitosos, errores encontrados, su prioridad y estado de resolución al momento del cierre.
* **Fecha de corte académico:** Independientemente del avance en la ejecución de pruebas, todo el proceso deberá concluir como máximo el día 3 de julio de 2025, fecha límite establecida por el calendario académico para la entrega oficial del proyecto.

# Tipos de Pruebas

## Pruebas Unitarias

Las pruebas unitarias tienen como objetivo validar el comportamiento de funciones, métodos y componentes individuales del sistema de forma aislada, garantizando que cada unidad de código cumpla su propósito sin errores internos.

En este proyecto, las pruebas unitarias se aplicarán tanto en el frontend (React) como en el backend (Node.js + Express), y se enfocarán principalmente en los controladores y servicios, donde se concentra la mayor parte de la lógica de negocio y procesamiento de datos.

**Herramientas utilizadas**

Se emplearán herramientas sencillas y ampliamente utilizadas en el ecosistema JavaScript/TypeScript, como:

* Jest: Framework de pruebas todo-en-uno para JavaScript, útil tanto en frontend como backend.
* Supertest: Para pruebas HTTP unitarias de endpoints Express.
* React Testing Library: Para probar componentes de React desde la perspectiva del usuario.

**Alcance y estrategia**

* Las pruebas serán una combinación de automáticas (mediante scripts) y manuales (para componentes de menor prioridad o complejidad baja).
* El equipo de QA es responsable de implementar, mantener y ejecutar estas pruebas.
* No se requiere pasar todas las pruebas unitarias como prerrequisito para las pruebas de integración, pero sí es recomendable ejecutar al menos un conjunto base antes de cada ciclo de prueba mayor.

**Cobertura esperada**

Se espera alcanzar un mínimo del 90% de cobertura en los módulos definidos como críticos (controladores y servicios), priorizando aquellos que forman parte de flujos esenciales como autenticación, postulación, generación de informes y gestión de usuarios.

## Pruebas de Integración

Las pruebas de integración tienen como objetivo verificar que los diferentes módulos y componentes del sistema interactúan correctamente entre sí. En esta fase se evalúan los puntos de conexión entre unidades funcionales, como la comunicación entre controladores, servicios, base de datos y la interfaz de usuario.

En el contexto del sistema de vinculación académica y empresarial, las pruebas de integración se centrarán en validar escenarios completos que combinan varias capas del sistema. Algunos ejemplos incluyen:

* Proceso completo de autenticación: desde la entrada de credenciales hasta la generación y validación del token JWT.
* Postulación de estudiantes y notificación automática a empresas.
* Aprobación de publicaciones por parte de administradores y visibilidad para usuarios finales.
* Carga de flyers y vinculación con publicaciones activas.
* Generación de informes desde el backend y su recuperación desde el frontend.

Estas pruebas permiten detectar errores en el intercambio de datos, inconsistencias en el flujo de control y fallos de comunicación entre módulos interdependientes. Para su ejecución se utilizarán entornos de prueba configurados en Render (tanto desarrollo como producción), donde los servicios pueden interactuar en condiciones reales.

Las pruebas de integración constituyen un puente fundamental entre las pruebas unitarias y las pruebas del sistema, y son esenciales para validar que el sistema opere como un todo coherente.

## Pruebas del Sistema

Las pruebas del sistema validan el comportamiento de la aplicación completa, desde el punto de vista del usuario final, evaluando flujos funcionales completos y requisitos definidos en el ERS. Estas pruebas aseguran que el sistema cumpla con las especificaciones funcionales y no funcionales bajo condiciones operativas normales.

Algunos escenarios que serán evaluados en esta etapa incluyen:

* Registro, autenticación y acceso a funcionalidades específicas según el rol.
* Publicación, revisión, edición y eliminación de oportunidades por parte de empresas.
* Exploración de vacantes, postulación y marcación de interés por parte de estudiantes.
* Gestión de usuarios, generación de informes y control de estados por parte de administradores.
* Visualización de estadísticas, interacción con componentes gráficos, y generación de documentos.

Estas pruebas se ejecutarán en el entorno de desarrollo o producción según el caso, e incluirán flujos completos con interacciones reales entre frontend, backend y base de datos.

## Pruebas de Aceptación

Las pruebas de aceptación tienen como finalidad confirmar que el sistema cumple con los requisitos establecidos por el cliente y los usuarios clave. Estas pruebas se realizan desde la perspectiva del usuario final, evaluando si el sistema satisface sus necesidades funcionales y de usabilidad.

Serán ejecutadas con acompañamiento o revisión de la patrocinadora del proyecto, y se enfocarán en:

* Flujo completo de postulación y aprobación de oportunidades.
* Generación de informes académicos y empresariales.
* Gestión de usuarios y control de agenda por parte de administradores.
* Visualización de información clave y respuestas adecuadas a errores comunes.

Las pruebas de aceptación representan el punto final de validación antes de la entrega formal del producto, y su éxito es determinante para la aprobación del sistema.

## Pruebas de Rendimiento

Las pruebas de rendimiento tienen como objetivo evaluar la capacidad del sistema para responder con eficiencia bajo condiciones de uso normales y exigentes. Estas pruebas buscan medir tiempos de respuesta, uso de recursos y comportamiento ante múltiples usuarios concurrentes.

Durante esta etapa se analizarán:

* Tiempos de carga de páginas clave (dashboard, publicaciones, reportes).
* Tiempos de respuesta de la API ante solicitudes simultáneas (registro, login, postulación, generación de documentos).
* Comportamiento del sistema con 30 a 50 usuarios concurrentes simulados.
* Consumo de CPU y memoria del backend en operaciones intensivas como generación de PDFs y filtrado de estadísticas.

Estas pruebas permitirán identificar cuellos de botella, validar la eficiencia de las consultas a base de datos y ajustar la configuración de los servicios en Render si fuese necesario.

# Cronograma

## Diseño de Pruebas

El diseño de pruebas constituye la primera etapa operativa del ciclo de pruebas y tiene como objetivo definir los casos que serán utilizados para validar las funcionalidades del sistema. Esta actividad contempla la estructuración de escenarios clave, condiciones de entrada, datos de prueba y resultados esperados, según lo establecido en los requisitos funcionales y la arquitectura del sistema.

**Planificación**

* **Inicio estimado:** Semana 15 del proyecto.
* **Duración:** 1 semana continua.
* **Responsable:** QA Leader.

**Estrategia**

El diseño de pruebas se desarrollará de forma incremental, es decir, los casos se irán generando en paralelo con el avance de los módulos desarrollados y conforme se completen funcionalidades validadas para prueba.

No se empleará una plantilla formal predeterminada; en su lugar, se utilizará una estructura flexible de casos registrada en hojas de cálculo (Excel o Google Sheets), adaptada a las necesidades de cada módulo o tipo de usuario.

**Detalle del diseño**

El nivel de detalle será moderado, enfocándose en flujos clave sin llegar a descripciones paso a paso para todos los casos. Esto permitirá una mayor agilidad en la generación y adaptación de pruebas durante las fases siguientes.

No se han identificado dependencias que impidan comenzar esta etapa. Con los documentos del ERS y el SAD ya aprobados, el diseño de pruebas puede comenzar con base sólida desde la planificación funcional establecida.

## Ejecución de Pruebas

La etapa de ejecución de pruebas consiste en aplicar los casos previamente diseñados sobre el sistema desplegado, con el fin de verificar el cumplimiento de los requisitos funcionales y detectar posibles errores. Esta fase constituye la validación principal del sistema desde una perspectiva funcional y operativa.

**Planificación**

* **Inicio estimado:** Semana 16 del proyecto.
* **Duración:** 2 semanas continuas.
* **Responsable:** Tester asignado.

**Estrategia**

La ejecución de pruebas se realizará en una sola pasada continua, es decir, se recorrerán todos los casos de prueba definidos para cada módulo sin dividir la ejecución en ciclos. Esta decisión se toma con base en el tiempo disponible y el tamaño del sistema.

Las pruebas se llevarán a cabo en el entorno de desarrollo desplegado en Render, garantizando condiciones representativas del entorno de producción sin afectar datos reales ni usuarios finales.

**Alcance**

Durante esta etapa se validarán todas las funcionalidades definidas en el ERS y disponibles en el sistema al momento de la prueba. Se probarán todos los roles de usuario (estudiante, egresado, empresa, administrador TFG y administrador de vinculación) en sus respectivos flujos funcionales.

No se contempla espacio formal para revalidación de errores corregidos durante esta misma fase; cualquier corrección deberá ser tratada como mejora posterior o, si el tiempo lo permite, revisada informalmente.

## Elaboración del Informe Final

La elaboración del informe final constituye la etapa de cierre del proceso de pruebas. Su objetivo es documentar de manera formal y estructurada los resultados obtenidos durante la ejecución, incluyendo métricas clave, hallazgos relevantes, cobertura alcanzada y una evaluación general del sistema.

**Planificación**

* **Inicio estimado:** Semana 16 del proyecto (en paralelo al cierre de ejecución de pruebas).
* **Duración:** 1 semana continua.
* **Responsables:** Tester, en colaboración con el equipo de QA.

**Contenido del informe**

El documento incluirá una compilación completa de los resultados, con énfasis en los siguientes aspectos:

* Resumen general del proceso de pruebas.
* Porcentaje de cobertura de casos ejecutados vs. planificados.
* Número y tipo de errores detectados, resueltos y pendientes.
* Módulos evaluados por tipo de usuario.
* Observaciones sobre funcionalidad, rendimiento y estabilidad.
* Capturas de pantalla como evidencia de ejecución.
* Recomendaciones para mejoras posteriores o refactorizaciones.

Además del contenido técnico, se incluirán gráficos visuales y tablas que ayuden a interpretar el estado del sistema en términos de calidad, cobertura y fiabilidad.

**Entrega**

El informe se generará como un documento PDF independiente, que será parte del entregable formal del proyecto, pero no como anexo al cierre general. Su finalidad es dejar un registro claro del estado del sistema validado, que pueda ser utilizado como referencia en futuras etapas de desarrollo, mantenimiento o auditoría.

No se requiere una aprobación formal para darlo por finalizado, aunque puede ser revisado por la patrocinadora o el equipo docente en el proceso de evaluación del proyecto.

# Roles y Responsabilidades

## QA Lead

## Tester

## Desarrolladores

## Product Owner

# Riesgos y Mitigaciones

Identificación de Riesgo

Evaluación de Probabilidad e Impacto

Estrategias de Mitigación

# Aprobaciones

## QA Leader

## Jefe del Proyecto

## Product Owner