

Plan de Pruebas

Kairos

NexTech

Centurión Valeria, Escalante Guillermo, Maldonado Agustina, Mendez Florencia, Ulloa Gonzalo.

**

**



**Tabla de contenido**

[**Introducción. 5**](#_35crghc6bu5t)

[**Definición de objetivos. 5**](#_mvtgvc34wune)

[**Consideraciones sobre el plan de pruebas. 5**](#_gkklfrz72o2y)

[**Antecedentes y Propósito 6**](#_qfwr2lscvcj1)

[*Antecedentes 6*](#_hoc0zgn2uu5c)

[*Propósito de la Evaluación 6*](#_p52acz9rpa7a)

[*Motivadores de la prueba 6*](#_96pykia8cgk1)

[**Objetos a ser Evaluados 7**](#_td6qssl5xa9q)

[**Ámbito de las Pruebas 7**](#_21fbwjnmkty5)

[*Dentro del Ámbito 7*](#_120fhs6c1i4k)

[*Fuera del Ámbito 7*](#_rk98gl9lcgg)

[**Lista de Ideas de las Pruebas 7**](#_se70fl6p4tbh)

[**Enfoque de las Pruebas 8**](#_b8ryygazbd9h)

[**Herramientas para las Pruebas 10**](#_er8k5tvtbnyf)

[*Software 10*](#_lrl21ynat4w4)

[*Herramientas de Soporte y Productividad 10*](#_dkfdegggku4x)

[*Secuencias de Comandos Personalizadas (Script de Pruebas) 10*](#_nfzxlw68pdpq)

[*Hardware 11*](#_t7m449xinfys)

[*Configuraciones de Pruebas de ambiente 11*](#_q191v4wfaaal)

[**Casos de Prueba 12**](#_5s672wo0liia)

[**Prioridades 13**](#_w9bymhz2el3e)

[*Casos de Prueba por Características de Prioridad 13*](#_vwk10hf43zsx)

[*Esenciales 13*](#_s8ybzy5jfe5m)

[*Esperadas 13*](#_tn9ksjewl64j)

[*Deseadas 13*](#_jjvbgadfoc)

[*Casos de Pruebas por Prioridad de Caso de Uso 13*](#_hvjw78s50aba)

[*Esenciales 13*](#_i6b8k0wfh998)

[*Esperados 13*](#_yqk22in0j1dk)

[*Deseados 13*](#_nylqk93u3a9z)

[*Flujos de Trabajo de Pruebas 13*](#_50cmaak13y1t)

[**Entregables 14**](#_tjslu9su6xjo)

[*Lista de Entregables de Pruebas 14*](#_nou0y3tyu2e8)

[*Ficha: Escenarios por Caso de Uso 14*](#_q5b933k4kwzz)

[*Ficha: Resumen de Ciclos de Prueba 15*](#_rf3i6j6iyj33)

[*Ficha: Matriz de Trazabilidad 15*](#_tf8gnx511jw5)

[**Criterio para el Inicio y Fin del Plan de Pruebas 16**](#_pwgonk4pzkri)

[*Criterios de Inicio 16*](#_69urxdhxyr4b)

[*Criterios de Fin 16*](#_5jrgj3its6pl)

[*Criterios de Suspensión y Retomo de Actividades 16*](#_rmjsemt0uwsv)

[**Criterios para el Lanzamiento 17**](#_v4c8a3sfhwgq)

[*Criterios de Evaluación 17*](#_kg4rul8yb8kb)

[*Clasificación de los errores 17*](#_4vfv4mr703yt)

[*Resultados de la prueba 17*](#_legjc4newyy6)

[*Reportes del problema, escalada y resolución 17*](#_h8gnls1cq5xp)

[**Riesgos 18**](#_7mnggzqp28zt)

[**Reportes de Problemas y Resolución 18**](#_vm21uenlr6do)

[**Responsabilidades, Personal y Necesidades de Capacitación 18**](#_5ec8xypwi552)

[*Personal y Roles Necesarios 18*](#_jjk0tz6x50m3)

[*Personal y Necesidades de Capacitación 19*](#_tfwg0iop6mcr)

Plan de Pruebas

# Introducción.

El presente Plan de Pruebas tiene como finalidad establecer las actividades necesarias para garantizar la calidad del sistema Kairos, una aplicación orientada a brindar a los desarrolladores una herramienta de apoyo en la planificación de tareas, registro de tiempos y control del avance de los proyectos.

Durante el proceso de pruebas se validará el correcto funcionamiento de los módulos principales de la aplicación, entre los que se incluyen: gestión de usuarios y autenticación, planificación y asignación de tareas, registro de horas trabajadas, seguimiento del progreso de proyectos y generación de reportes. Estas pruebas permitirán identificar posibles errores, inconsistencias o desviaciones respecto a los requerimientos definidos, asegurando que el sistema cumpla con las expectativas de los usuarios finales.

Para el desarrollo de las actividades de prueba se emplearán recursos tanto humanos como técnicos. Dentro de los recursos humanos se contará con la participación de analistas de calidad, desarrolladores y usuarios de prueba, quienes colaborarán en la preparación, ejecución y validación de los resultados obtenidos. En cuanto a los recursos técnicos, se dispondrá de entornos de prueba, herramientas de gestión de pruebas, sistemas de seguimiento de incidencias y navegadores web que faciliten la ejecución de los casos de prueba en diferentes escenarios.

Este documento establece la planificación estructurada del proceso de verificación del sistema Kairos, asegurando que cada módulo sea evaluado de manera adecuada, con la participación de los actores correspondientes y el empleo de los recursos necesarios para garantizar la entrega de un producto confiable y de calidad.

# Definición de objetivos.

### Objetivo General

Asegurar la calidad del sistema Kairos, mediante la ejecución de pruebas planificadas que permitan verificar el correcto funcionamiento de sus módulos, validar el cumplimiento de los requerimientos establecidos y garantizar que la aplicación sea confiable, usable y adecuada para los usuarios finales.

### Objetivos Específicos

* **Verificar la funcionalidad de los módulos principales** (gestión de usuarios, planificación de tareas, registro de tiempo, control de avances y generación de reportes), asegurando que respondan conforme a lo especificado.
* **Identificar y documentar defectos o inconsistencias** que puedan presentarse durante la ejecución de los casos de prueba, para su posterior corrección.
* **Validar la integración entre los diferentes módulos** de la aplicación, comprobando que los datos fluyan de manera correcta y coherente.
* **Evaluar la compatibilidad del sistema** en distintos navegadores y entornos de prueba.
* **Comprobar la experiencia de usuario (usabilidad)** para garantizar que la interfaz sea intuitiva y facilite las tareas a los desarrolladores.
* **Asegurar la trazabilidad de los requerimientos** a través de la ejecución de pruebas que demuestren su cumplimiento.
* **Proporcionar información objetiva y medible** sobre la calidad del producto, que sirva de apoyo en la toma de decisiones respecto a su liberación.

# Propósito

## Propósito de la Evaluación

La calidad de un sistema es el grado en el cual los atributos del sistema operacional permiten que realice las funciones específicas para satisfacer los requerimientos del cliente. El propósito del plan de pruebas es asegurar que los atributos del sistema Kairos cumplan con las expectativas del cliente.

### Motivadores de la prueba

La presente iteración de pruebas sobre el sistema **Kairos** se fundamenta en la necesidad de garantizar que la aplicación cumpla con los objetivos de calidad, funcionalidad y rendimiento establecidos. Los principales elementos que motivan la realización de estas pruebas son:

* **Requerimientos Funcionales**
  + Verificar que los módulos de planificación de tareas, registro de tiempo y control de avance cumplan con las funcionalidades esperadas por los usuarios.
  + Asegurar que la gestión de usuarios, asignación de roles y permisos funcionen correctamente.
  + Validar la correcta generación de reportes y visualización de datos.
* **Requerimientos No Funcionales**
  + Garantizar compatibilidad entre los navegadores soportados.
  + Verificar la usabilidad e interfaz amigable para los desarrolladores.
  + Evaluar el rendimiento del sistema frente a tiempos de respuesta aceptables.
* **Cambios de Requerimientos**
  + Incorporar cualquier modificación reciente en las funcionalidades o mejoras sugeridas durante la etapa de diseño que puedan afectar el comportamiento de la aplicación.
* **Otros**
  + Cumplir con estándares de calidad académicos y buenas prácticas de pruebas de software.
  + Prevenir errores potenciales en la integración de módulos antes de la entrega final del proyecto.

# Objetos a ser Evaluados

Los objetos de prueba son los siguientes módulos y componentes de Kairos:

* Módulo de Gestión de Usuarios: registro, autenticación y asignación de roles.
* Módulo de Proyectos: creación, edición y asociación de usuarios a proyectos.
* Módulo de Etapas e Iteraciones: definición de ciclos de trabajo por proyecto.
* Módulo de Tareas: creación, modificación, aprobación y categorización.
* Módulo de Registro de Tiempo: registro manual y por cronómetro de horas trabajadas.
* Módulo de Reportes: generación y exportación de reportes por usuario o proyecto.

# Ámbito de las Pruebas

El objetivo de esta iteración es verificar que la solución desarrollada cumple con los requerimientos funcionales y no funcionales definidos, asegurando que cada componente del sistema se comporta según lo esperado en su entorno de ejecución. Para ello, se contemplan tareas como:

* Validación de la correcta carga y visualización de datos en la interfaz.
* Verificación del funcionamiento de formularios, botones y enlaces.
* Comprobación de la persistencia de la información en la base de datos.
* Evaluación del comportamiento del sistema frente a entradas válidas e inválidas.
* Pruebas de compatibilidad básica en los navegadores soportados.

## Dentro del Ámbito

Para esta iteración, se estructura el conjunto de pruebas de la siguiente manera:

* **Pruebas funcionales:** Validación de la ejecución correcta de cada función o módulo implementado.
* **Pruebas de interfaz:** Comprobación de la consistencia de la interfaz y la usabilidad del sistema.
* **Pruebas de integración:** Verificación de la correcta interacción entre módulos del sistema.
* **Pruebas de compatibilidad básica:** Evaluación en los navegadores definidos como soportados (por ejemplo, Chrome y Firefox).
* **Pruebas de mantenimiento:** Evaluación de la facilidad para actualizar, modificar y mantener el sistema, incluyendo la correcta gestión de errores y registros de actividad.
* **Pruebas de seguridad básicas:** Comprobación de acceso autorizado, protección de datos sensibles y correcta gestión de sesiones.

## Fuera del Ámbito

Las siguientes pruebas quedan excluidas de esta iteración, por considerarse de alcance avanzado o por requerir entornos especiales no disponibles en este ciclo:

* **Pruebas de rendimiento extremo y estrés**: No se incluyen debido a que se requiere un entorno con simulación de múltiples usuarios simultáneos, lo cual excede los recursos disponibles.
* **Pruebas avanzadas de compatibilidad con navegadores móviles**: Se excluyen en esta iteración, considerando que el enfoque inicial es la versión web de escritorio.

# Lista de Ideas de las Pruebas

Las pruebas se basan en los Casos de Uso (CU) definidos en la especificación del sistema. Se probarán los flujos normales y alternativos de cada uno:

* CU01: Iniciar sesión
* CU02: Exportar información
* CU05: Crear proyecto
* CU07: Crear etapa
* CU08: Crear iteración
* CU09: Crear tarea
* CU10: Sugerir tarea
* CU11: Aprobar tarea
* CU14: Modificar estado de tarea
* CU15: Ver tareas
* CU17: Categorizar tiempo
* CU20: Registrar tiempo por cronómetro
* CU21: Registrar tiempo manual
* CU22: Ver reporte
* CU23: Registrar usuario al proyecto

Para cada caso de uso se definirán entradas, precondiciones y resultados esperados.

# Enfoque de las Pruebas

| **Tipo de prueba** | **Objetivo** | **Descripción** | **Técnicas** | **Fases** | **Entorno** | **Criterios de Éxito** |
| --- | --- | --- | --- | --- | --- | --- |
| **Pruebas unitarias** | Validar funciones individuales | Se probarán los métodos del backend con JUnit  y Mockito | Caja blanca | Desarrollo | Local con Docker | Todos los métodos devuelven salida esperada |
| **Pruebas de integración** | Verificar la interacción entre módulos | Se probará la conexión entre backend y frontend | Caja negra | Post-  desarrollo | Local | Módulos integrados sin error |
| **Pruebas funcionales** | Validar requerimientos del usuario | Se probarán casos de uso principales | Caja negra | Validación | Chrome  Firefox | Casos ejecutados correctamente |
| **Pruebas de interfaz** | Evaluar usabilidad  y coherencia visual | Revisión de formularios, botones y navegación | Inspección manual | Validación | Web | Interfaz clara y funcional |
| **Pruebas de seguridad básica** | Validar accesos  y roles | Intentos de acceso no autorizado | Exploración manual | Validación | Web | Sin acceso sin login válido |

# Herramientas para las Pruebas

## Software

Las herramientas que se utilizarán para las pruebas son:

| **Herramienta** | **Versión** | **Descripción** |
| --- | --- | --- |
| **JUnit** | 5.x | Framework para pruebas unitarias en Java. |
| **Mockito** | 5.x | Simulación de dependencias en backend. |
| **Postman** | 11.x | Pruebas de endpoints REST. |
| **Selenium** | 4.x | Automatización de pruebas funcionales de la UI. |
| **Visual Studio Code** | 1.94 | Entorno de desarrollo y depuración. |
| **GitHub** | Última | Control de versiones. |
| **Google Sheets** | — | Registro de resultados de pruebas. |

## Herramientas de Soporte y Productividad

| Nombre | Descripción |
| --- | --- |
| Plantilla de Excel | Gestión de tareas de prueba |
| Whatsapp | Comunicación interna del equipo |
| Canva | Documentación visual |
| Draw.io | Documentación visual |

## Hardware

| Recurso | Cantidad | Descripción |
| --- | --- | --- |
| PC de escritorio/notebook | 5 | Equipos de los integrantes del grupo con procesadores I3/i5/i7 y conexión estable. |
| Servidor local (Docker) | 1 | Contenedor con backend Kairos y base de datos MySQL. |

## 

# Casos de Prueba

| **ID/Nombre/Sistema/Proyecto: Kairos** | | | **Nivel de Prueba: 1** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **ID Caso de Uso:** CU01 | | | **Tipo(s) de Pruebas(s):Funcional / Interfaz / Seguridad básica** | | | |
| **ID Requerimiento:** REQ01 – Autenticación con Google | | | **Ambiente de Prueba:** Localhost / Navegador Chrome | | | |
| **ID/Nombre Escenario:** Escenario de login correcto con cuenta válida de administrador | | | **Autor del Caso de Prueba: Centurión Valeria** | | | |
| **ID/Nombre Caso de Prueba:** CP1\_1 – Login exitoso con cuenta administrador | | | **Nombre del Probador: Ulloa gonzalo** | | | |
| **Versión del Caso de Prueba:** 1.0 | | | **Fecha de Creación:**  09/10/2025 | | **Fecha de Ejecución:** | |
| **Condición(es) para que se ejecute el Caso de Prueba:** | | | | | | |
| El usuario debe tener vinculada previamente su cuenta de Google al sistema. | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
| **Para la Ejecución del Caso de Prueba:** | | | | | | |
| **Nro. Paso Flujo** | **Condición** | **Valor(es)** | | **Resultado Esperado** | | **Resultado Obtenido** |
| 1 | El usuario administrador ingresa al sistema | correo: | | El sistema valida la cuenta y permite el acceso | | Resultado esperado |
| 2 | El sistema muestra la pantalla principal del administrador | - | | El sistema concede el acceso de administrador | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
| **Criterios de Aprobación del Caso de Prueba:** Si el sistema permite ingresar a la cuenta y reconoce el rol del usuario administrador, se aprueba. | | | | | | |
| **Decisión de Aprobación del Caso de Prueba:** Aprobó: \_\_\_ Fallo: \_\_\_ (marque con x el resultado) | | | | | | |
| **Fecha de Aprobación del Caso de Prueba:**  \_\_\_\_\_\_\_\_\_\_\_ | | | | | | |

# Prioridades

## Casos de Prueba por Características de Prioridad

### Esenciales

Características críticas sin las cuales el sistema no puede funcionar

| **Característica** | **Descripción** | **Justificación de Prioridad** |
| --- | --- | --- |
| Autenticación y Control de Acceso | Sistema de login y gestión de roles/perfiles | Sin esto no hay seguridad ni personalización de experiencia |
| Gestión Básica de Proyectos | Crear, visualizar y administrar proyectos | Funcionalidad core del sistema - razón de ser de la aplicación |
| Gestión de Tareas | Creación, asignación y seguimiento de tareas | Proceso fundamental del workflow de desarrollo |
| Registro de Tiempos | Captura manual y por cronómetro del tiempo trabajado | Valor principal diferenciador del sistema |
| Persistencia de Datos | Almacenamiento confiable y recuperación de información | Garantía de que no se pierde información crítica |

### Esperadas

Características importantes que los usuarios esperan encontrar

| **Característica** | **Descripción** | **Justificación de Prioridad** |
| --- | --- | --- |
| Generación de Reportes | Visualización de métricas y progreso de proyectos | Información para toma de decisiones - valor agregado clave |
| Exportación de Datos | Descarga de información en formatos estándar | Compatibilidad con otras herramientas y procesos |
| Gestión de Usuarios en Proyectos | Asignación y administración de equipos por proyecto | Trabajo colaborativo - necesidad de trabajo en equipo |
| Flujos de Aprobación | Validación y aceptación de trabajo completado | Control de calidad y formalización de avances |
| Categorización de Tiempos | Clasificación del tiempo por tipos de actividad | Análisis granular de cómo se usa el tiempo |

### Deseadas

Características que mejoran la experiencia pero no son críticas

| **Característica** | **Descripción** | **Justificación de Prioridad** |
| --- | --- | --- |
| Sugerencia Automática de Tareas | Recomendaciones basadas en patrones | Mejora de productividad - valor agregado avanzado |
| Personalización  de Vistas | Adaptación de interfaces según preferencias | Mejora experiencia de usuario  - no afecta funcionalidad core |
| Notificaciones  y Recordatorios | Alertas automáticas de vencimientos | Conveniencia pero no esencial para operación básica |
| Temas  y Customización Visual | Cambio de colores y estilos de interfaz | Aspecto estético - no impacta funcionalidad |

## Casos de Pruebas por Prioridad de Caso de Uso

### Esenciales

Casos de uso sin los cuales el sistema no cumple su propósito principal

| **Caso de Uso** | **ID** | **Descripción** | **Justificación** |
| --- | --- | --- | --- |
| **Iniciar Sesión** | **CU01** | Permite que el usuario acceda al sistema mediante credenciales válidas. | Es el punto de entrada obligatorio. Sin autenticación no se puede acceder a ninguna funcionalidad. |
| **Crear Proyecto** | **CU05** | Permite iniciar un nuevo proyecto de desarrollo | Sin esto no puede comenzarse a usar el sistema |
| **Crear Tarea** | **CU09** | Generar actividades específicas dentro de un proyecto | Unidad básica de trabajo - esencial para tracking |
| **Registrar Tiempo por Cronómetro** | **CU20** | Captura automática de tiempo trabajado | Funcionalidad diferenciadora principal del sistema |
| **Registrar Tiempo Manual** | **CU21** | Ingreso manual de horas trabajadas | Alternativa esencial cuando el cronómetro no aplica |
| **Ver Tareas** | **CU15** | Visualización del listado y estado de tareas | Necesidad básica de seguimiento y organización |
| **Modificar Estado de Tarea** | **CU14** | Cambiar el progreso de una tarea (pendiente, en curso, etc.) | Fundamental para reflejar avances reales |

### Esperados

Casos de uso importantes para una experiencia completa

| **Caso de Uso** | **ID** | **Descripción** | **Justificación** |
| --- | --- | --- | --- |
| Ver Reporte | CU22 | Generación de vistas consolidadas de progreso | Para análisis y toma de decisiones - valor agregado |
| Exportar Información | CU02 | Descarga de datos en formatos exportables | Necesidad de compartir información fuera del sistema |
| Registrar Usuario al Proyecto | CU23 | Asignación de miembros a proyectos específicos | Trabajo colaborativo en equipos |
| Aprobar Tarea | CU11 | Validación formal de tareas completadas | Control de calidad y formalización de procesos |
| Categorizar Tiempo | CU17 | Clasificación del tiempo por tipo de actividad | Análisis más detallado de productividad |
| Crear Etapa | CU07 | Definición de fases dentro de un proyecto | Organización de proyectos complejos |

### Deseados

Casos de uso que mejoran la experiencia pero no son críticos

| Caso de Uso | ID | Descripción | Justificación |
| --- | --- | --- | --- |
| Sugerir Tarea | CU10 | Recomendaciones automáticas de tareas | Mejora productividad  pero no esencial |
| Crear Iteración | CU08 | Definición de sprints o ciclos de trabajo | Para metodologías ágiles /avanzado |

# Entregables

## Lista de Entregables de Pruebas

| Entregables | Descripción |
| --- | --- |
| Documento de caso de  prueba | Por cada caso de prueba, se entregará un documento detallando el responsable de la ejecución del caso de prueba, los pasos realizados por cada caso de prueba, la fecha en la que se realizó, los resultados esperados y los resultados obtenidos. |

## Ficha: Escenarios por Caso de Uso

| <ID/Nombre de Caso de Uso> | | | | |
| --- | --- | --- | --- | --- |
| ID Escenario | Flujo Básico | Flujo Alterno 1 | Flujo Alterno 2 | Flujo Alterno (n) |
|  |  |  |  |  |

## Ficha: Resumen de Ciclos de Prueba

Se irá completando esta sección al iniciar la etapa de pruebas del sistema.

ID del Proyecto/ Nombre:

ID del Ciclo de Prueba:

Fechas para el Ciclo de Prueba: Desde: Hasta:

| ID Caso de Uso | ID Caso de Pruebas | Resultados Esperados | Resultados Obtenidos | Observación |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Resultados/Observaciones para el Ciclo de Prueba:

Aprobado Ciclo de Prueba por:

CLIENTE ORGANIZACIÓN PROBADOR

## Ficha: Matriz de Trazabilidad

Se irá completando esta sección al iniciar la etapa de pruebas del sistema.

| ID Caso de Uso | ID de Escenario | ID de Caso de Prueba | ID Tipo de Prueba | ID Ciclo de Prueba |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Criterio para el Inicio y Fin del Plan de Pruebas

## Criterios de Inicio

Para comenzar con el plan de pruebas se necesitará:

* Entorno de desarrollo estable con todas las herramientas configuradas (Spring Boot, Angular, MySQL)
* Funcionalidades esenciales implementadas y compiladas sin errores críticos
* Casos de uso documentados y especificaciones técnicas aprobadas
* Equipo completo disponible y capacitado en herramientas de prueba

## Criterios de Fin

El plan de prueba se dará por terminado si se ejecutaron todos los casos de prueba planificados, se corrigieron todos los errores críticos encontrados durante las pruebas realizadas y validaron las funcionalidades principales del sistema.

## Criterios de Suspensión y Retomo de Actividades

La ejecución de las pruebas será suspendida en los siguientes casos:

* Se presentan errores críticos que impidan la continuidad de la ejecución de los casos de prueba.
* El entorno de pruebas experimenta fallas que afecten el desarrollo normal de las actividades planificadas.
* El tester responsable no se encuentra disponible para la realización de las pruebas.

Las actividades serán reanudadas una vez que:

* Los errores críticos han sido corregidos.
* El entorno de pruebas se encuentra nuevamente estable.
* Se confirme la disponibilidad o reasignación del personal correspondiente.

# Criterios para el Lanzamiento

## Criterios de Evaluación

El producto de software será considerado listo para su **puesta en producción** una vez que se hayan ejecutado exitosamente todos los casos de prueba correspondientes a las funcionalidades críticas y no se registren errores de alta prioridad pendientes de resolución.

## Clasificación de los errores

| **Calificación** | **Definición de gravedad** | **Definición de prioridad** |
| --- | --- | --- |
| 1 | Error crítico que provoca bloqueo del sistema, pérdida de datos o imposibilidad de continuar con tareas esenciales. | Debe corregirse de inmediato, ya que impide el avance de la ejecución de pruebas y afecta funcionalidades críticas. |
| 2 | Error grave que afecta la funcionalidad principal o produce bloqueos en escenarios poco frecuentes. | Debe corregirse antes del lanzamiento del producto para garantizar la operatividad y continuidad de las pruebas. |
| 3 | Error de funcionalidad menor o inconsistencia que no impide el avance, pero genera resultados incorrectos o confusión en el usuario. | La corrección es recomendable antes del lanzamiento, aunque no bloquea las pruebas ni el uso del sistema. |
| 4 | Error de usabilidad o detalle estético que no afecta la funcionalidad (mensajes confusos, interfaz poco clara, elementos gráficos). | La corrección puede posponerse a futuras versiones, ya que no impacta la ejecución de pruebas ni el uso crítico del sistema. |

Con el fin de mantener la trazabilidad con el Plan de Aseguramiento de la Calidad (SQA) se incorporan las siguientes métricas de control:

| **Métrica** | **Descripción** | **Fórmula / Método de Cálculo** | **Objetivo** |
| --- | --- | --- | --- |
| **Porcentaje de casos de prueba aprobados** | Mide el nivel de éxito de las pruebas ejecutadas. | (Casos aprobados / Total de casos ejecutados) × 100 | Evaluar el cumplimiento de los requerimientos probados. |
| **Defectos detectados por iteración** | Cantidad total de defectos encontrados en cada ciclo de prueba. | Número de defectos registrados en una iteración. | Identificar la estabilidad del sistema a lo largo de las iteraciones. |
| **Tasa de resolución de defectos** | Mide la eficiencia del equipo en resolver errores. | (Defectos corregidos / Defectos reportados) × 100 | Evaluar la capacidad de respuesta ante incidencias. |
| **Severidad promedio de defectos** | Evalúa el impacto promedio de los defectos encontrados. | (Σ Nivel de severidad de defectos / Total de defectos) | Controlar si los errores afectan funcionalidades críticas. |

Las métricas incluidas en este plan se basan en las definidas en el Plan de SQA del proyecto Kairos, adaptadas al proceso de pruebas.

# Riesgos

| **Riesgo** | **Estrategias**  **de Mitigación** | **Plan**  **de Contingencia** | **Impacto** |
| --- | --- | --- | --- |
| **RK07: Dependencia herramientas gratuitas** | Backup diario en Google Drive, Identificar herramientas alternativas, Monitoreo de límites de capacidad | Migración a alternativas pre-configuradas, Uso de versiones locales | Coordinación del equipo, Acceso a documentación, Continuidad de pruebas |
| **RK10: Falta conocimientos en herramientas** | Sesiones de capacitación en JUnit/Postman, Pair programming en pruebas complejas, Tutoriales específicos | Reasignación de tareas según competencias, Apoyo intensivo del líder técnico | Calidad de pruebas automatizadas, Velocidad de ejecución, Cobertura de pruebas |
| **RK11: Arquitectura mal planificada** | Revisiones técnicas tempranas, Prototipos de integración, Validación con líder técnico | Refactorización de módulos problemáticos, Pruebas de integración extendidas | Integración entre módulos, Estabilidad del sistema, Eficiencia de pruebas |
| **RK14: Datos de prueba inadecuados** | Creación de dataset representativo, Validación de datos con equipo, Scripts de reset de BD | Generación manual de datos críticos, Pruebas con datos mínimos | Calidad de validación, Detección de errores, Confiabilidad de resultados |
| **RK05: Falta definición requerimientos** | Reuniones de clarificación semanales, Documentación en tiempo real, Validación incremental | Pruebas exploratorias adicionales, Ajuste de casos de prueba, Priorización de funcionalidades | Cobertura de pruebas, Eficiencia del equipo, Cumplimiento de objetivos |
| **RK16: Tiempo limitado cuatrimestre** | Priorización estricta por valor, Automatización de pruebas repetitivas, Paralelización de actividades | Enfoque en pruebas esenciales, Reducción de alcance acordada, Entrega por fases | Cobertura completa, Calidad final, Cumplimiento de fechas |
| **RK03: Ausencia de integrantes** | Distribución balanceada de conocimientos, Documentación de procedimientos, Reuniones diarias | Reasignación de responsabilidades, Enfoque en actividades críticas, Apoyo entre miembros | Velocidad de ejecución, Cumplimiento de cronograma, Calidad de trabajo |

# Responsabilidades, Personal y Necesidades de Capacitación

## Personal y Roles Necesarios

| Nombrede Rol | Cantidad Necesaria | Estado | Responsabilidades Específicas o Comentarios |
| --- | --- | --- | --- |
| Tester / QA | 3 | Asignado | Ejecutar los casos de prueba.Registrar y reportar incidencias.Verificar correcciones. |
| Analista | 2 | Asignado | Supervisar la ejecución del plan de pruebas.Coordinar la comunicación con el equipo de desarrollo. |
| Desarrollador | 2 | Asignado | Corregir errores reportados.Apoyar en la validación técnica de los defectos.Documentar las soluciones implementadas. |
| Líder de Proyecto | 1 | Asignado | Monitorear el avance del proceso de pruebas.Validar cierres de incidencias críticas. Gestionar riesgos y priorización. |

## 