**Plan de Pruebas**

***[NUTRIMOVE]***

***Fecha: [05/11/2024]***

Tabla de contenido

[**Historial de Versiones 2**](#_heading=h.dhhnh1hpm49x)

[**Información del Proyecto 3**](#_heading=h.30j0zll)

[**Aprobaciones 3**](#_heading=h.7edpgc6wz6hj)

[**Alto Nivel: Áreas Funcionales, Módulos y Procesos 5**](#_heading=h.bk9tpse044dh)

[**Nivel Detallado: Programas, Unidades o Módulos 6**](#_heading=h.h3x3eqswewtm)

[**Pruebas de Regresión (Desde el Punto de Vista del Usuario) 12**](#_heading=h.tlc9xijcduhz)

[**Tipos de Pruebas a Realizar 14**](#_heading=h.m956v8yijhs)

[**Configuraciones a Probar 16**](#_heading=h.v84nctkp70oq)

[**Nivel de Pruebas de Regresión 16**](#_heading=h.ijzgijl8m328)

[**Requerimientos de Entornos – Hardware 23**](#_heading=h.44sinio)

[**Requerimientos de Entornos – Software 24**](#_heading=h.wld5ajjiidy)

[**Procedimientos para las Pruebas 29**](#_heading=h.2xcytpi)

# Historial de Versiones

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 06/11/24 | 1.0 | NutriMove | DuocUC | Versión inicial |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Información del Proyecto

| Empresa / Organización | Equipo de DuocUC |
| --- | --- |
| Proyecto | NUTRIMOVE |
| Fecha de preparación |  |
| Cliente | \* |
| Patrocinador principal | \* |
| Gerente / Líder de Proyecto | Patricio Catejo |
| Gerente / Líder de Pruebas de Software | Alejandro Vásquez |

# Aprobaciones

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Resumen Ejecutivo

El plan de pruebas para el proyecto informático “**NutriMove”** tiene como propósito garantizar que la aplicación funcione correctamente, cumpla con los requisitos de calidad y satisfaga las expectativas de los usuarios.

El objetivo del plan de pruebas es:

* Validar la precisión y efectividad de las recomendaciones de ejercicio proporcionadas por la IA.
* Asegurar que la aplicación tenga una interfaz intuitiva y amigable.
* Verificar que el seguimiento de avances y métricas del usuario se realice correctamente.
* Comprobar que la aplicación sea robusta, escalable y que opere sin fallos en diferentes dispositivos y plataformas.

Este plan de pruebas se considera un **plan detallado**, ya que aborda las actividades específicas de prueba para cada componente de la aplicación Nutrimove, así como los recursos, responsabilidades, y tiempos asignados a cada fase de las pruebas. Aunque puede haber un **plan maestro** que cubra pruebas a nivel global para todo el proyecto, este es un plan específico enfocado en el software.

Alcance de las Pruebas

Elementos de Pruebas

# Alto Nivel: Áreas Funcionales, Módulos y Procesos

**Área de Autenticación**

* Módulo de autenticación: Incluye los procesos de inicio de sesión, registro de usuario y recuperación de contraseñas.

**Área de Gestión del Perfil de Usuario**

* Módulo de Perfil de Usuario: Configuración y visualización de la información del usuario, preferencias personales, y seguimiento de progreso.

**Área de Nutrición**

* Módulo de Nutrición: Generación de recomendaciones personalizadas, seguimiento de calorías, macronutrientes y comunicación con la API de Gemini para mejorar sugerencias.

**Área de Ejercicio**

* Módulo de Ejercicio: Creación de rutinas personalizadas, registro de ejercicios y métricas de rendimiento, con integración a la API de Gemini para ajuste de rutinas.

**Área de Almacenamiento de Datos**

* Base de Datos (Firebase): Procesos de almacenamiento en tiempo real, sincronización de datos de usuario y gestión de permisos de acceso.

**Área de Interfaz de Usuario (UI)**

* Interfaz de Usuario (UI): Procesos de navegación, usabilidad, y consistencia en diseño visual.

**Área de Configuración de la Aplicación**

* Módulo de Configuración: Personalización de notificaciones, preferencias y restablecimiento de ajustes.

**Área de Notificaciones y Recordatorios**

* Módulo de Notificaciones y Recordatorios: Generación de recordatorios automáticos y configurables para nutrición y ejercicio, con gestión de frecuencia y puntualidad en alertas.

# Nivel Detallado: Programas, Unidades o Módulos

**Módulo de Autenticación**

* Inicio de sesión de usuario (Unidad): Permite al usuario acceder a la aplicación con sus credenciales.
* Registro de usuario (Unidad): Proceso para la creación de una nueva cuenta de usuario.
* Recuperación de contraseña (Unidad): Opción para recuperar el acceso en caso de olvidar la contraseña.

**Módulo de Perfil de Usuario**

* Actualización de datos personales (Programa): Permite al usuario modificar su información (nombre, edad, estatura, peso, etc.).
* Gestión de preferencias de ejercicio y nutrición (Programa): Configuración de preferencias que afectarán las recomendaciones.
* Visualización de historial de progreso y estadísticas (Programa): Muestra al usuario un resumen de su progreso.

**Módulo de Nutrición**

* Generación y visualización de recomendaciones nutricionales (Programa): Sugiere planes de alimentación personalizados.
* Seguimiento de calorías y macronutrientes (Unidad): Permite al usuario llevar el control de su ingesta.
* Interacción con la API de Gemini (Unidad): Obtiene sugerencias personalizadas para mejorar la alimentación.

**Módulo de Ejercicio**

* Generación de rutinas de ejercicio personalizadas (Programa): Crea rutinas basadas en los datos y objetivos del usuario.
* Registro de ejercicios realizados y métricas de rendimiento (Unidad): Almacena el historial de ejercicios y el progreso.
* Integración con la API de Gemini (Unidad): Ajusta las rutinas según las sugerencias de la API.

**Base de Datos (Firebase) (Módulo)**

* Almacenamiento de datos de usuario en tiempo real (Unidad): Guarda los datos de usuario en la base de datos de Firebase.
* Sincronización y gestión de datos entre la aplicación y Firebase (Programa): Sincroniza la información de usuario con Firebase.
* Control de seguridad y permisos de acceso a los datos (Unidad): Administra permisos de acceso y protege los datos almacenados.

**Interfaz de Usuario (UI) (Módulo)**

* Pruebas de navegación entre pantallas (Unidad): Verifica que la navegación en la aplicación sea funcional y sin errores.
* Respuesta y funcionalidad de botones e interacciones (Unidad): Prueba la respuesta de todos los elementos interactivos.
* Comprobación de accesibilidad y consistencia de diseño (Unidad): Asegura que el diseño sea accesible y visualmente consistente.

**Módulo de Configuración**

* Personalización de configuraciones de la aplicación (Programa): Permite ajustar notificaciones y preferencias.
* Funcionalidad para restablecer ajustes y datos (Unidad): Permite reiniciar los ajustes a su configuración inicial.
* Manejo de conexión y desconexión de la cuenta (Unidad): Gestiona el acceso del usuario y la opción de cierre de sesión.

**Notificaciones y Recordatorios (Módulo)**

* Generación y gestión de recordatorios de nutrición y ejercicio (Unidad): Envía recordatorios basados en el plan del usuario.
* Personalización de la frecuencia y contenido de las notificaciones (Unidad): Ajuste de la frecuencia y contenido de recordatorios.
* Verificación de consistencia y puntualidad en las alertas (Programa): Asegura que las notificaciones se envíen correctamente.

**Tipos de pruebas a realizar**:

* **Pruebas funcionales**: Asegurar que todas las funciones de la aplicación, como la generación de rutinas personalizadas, el seguimiento de avances y la interfaz de usuario, funcionen de acuerdo con los requisitos.
* **Pruebas de usabilidad**: Evaluar la facilidad de uso y la experiencia del usuario.
* **Pruebas de rendimiento**: Medir el rendimiento de la aplicación en términos de velocidad, tiempo de respuesta y uso eficiente de los recursos, especialmente con un gran volumen de datos.
* **Pruebas de seguridad**: Verificar la seguridad de los datos del usuario, incluyendo la protección de información personal y los datos de salud.
* **Pruebas de integración**: Validar que los distintos módulos y servicios de la aplicación se integren correctamente entre sí.
* **Pruebas de compatibilidad**: Verificar que la aplicación funcione correctamente en diferentes dispositivos, sistemas operativos y navegadores.

**Exclusiones**:

* El plan de pruebas no cubre el hardware asociado con los dispositivos de seguimiento físico (como pulseras o relojes de fitness), ya que estas pruebas no se consideran parte del alcance de la aplicación.
* No se realizarán pruebas de marketing o de experiencia de usuario a gran escala (por ejemplo, pruebas A/B para estrategias de marketing).

Nuevas Funcionalidades a Probar

**Inicio de Sesión**

* Acceso rápido con gmail

**Registro de Usuario**

* Creación de cuenta con campos mínimos (email, contraseña, etc).
* Confirmación de cuenta creada exitosamente.

**Recuperación de Contraseña**

* Opción sencilla para recuperar acceso en caso de olvido de contraseña.
* Mensaje de confirmación de envío de instrucciones de recuperación.

Pruebas de Regresión

# Pruebas de Regresión (Desde el Punto de Vista del Usuario)

**Perfil de Usuario**

* Visualización correcta de datos personales, sin errores en la carga.
* Capacidad de actualizar información (nombre, edad, etc.) sin problemas tras el inicio de sesión.

**Recomendaciones de Nutrición y Ejercicio**

* Generación y visualización de sugerencias personalizadas sin interrupciones.
* Precisión en las recomendaciones basadas en los datos de perfil, sin pérdida de datos.

**Sincronización de Datos en Firebase**

* Almacenamiento seguro y consistente de datos personales y preferencias.
* Sincronización en tiempo real sin duplicaciones ni pérdida de datos.

**Interfaz de Usuario (UI)**

* Navegación fluida y sin errores después del inicio de sesión o actualización de perfil.
* Respuesta de botones e interacciones de usuario sin retrasos.

**Notificaciones y Recordatorios**

* Generación de recordatorios de rutina y nutrición sin interrupciones.
* Entrega puntual y precisa de notificaciones de acuerdo a la configuración del usuario.

Funcionalidades a No Probar

**Integración Completa con la API de Gemini**

* Razón: La integración avanzada con la API de Gemini para recomendaciones personalizadas aún no está totalmente implementada y será probada en una fase futura.
* Riesgo: Las recomendaciones nutricionales y de ejercicio pueden ser básicas o genéricas, lo que podría impactar la personalización esperada por el usuario.

**Configuración Avanzada de Notificaciones**

* Razón: La funcionalidad de ajuste detallado en la frecuencia y contenido de notificaciones aún no está finalizada y no es crítica para la versión inicial.
* Riesgo: Los usuarios recibirán notificaciones estándar sin opciones de personalización, lo cual podría reducir la satisfacción en términos de flexibilidad.

Enfoque de Pruebas (Estrategia)

# Tipos de Pruebas a Realizar

**Pruebas Funcionales:** Validar que cada funcionalidad principal (autenticación, perfil, sincronización de datos) funcione según lo esperado desde la perspectiva del usuario.

**Pruebas de Interfaces (UI):** Verificar la navegación, usabilidad, y respuesta de elementos interactivos para garantizar una experiencia fluida e intuitiva.

**Pruebas de Regresión:** Asegurar que las nuevas implementaciones (principalmente en autenticación) no afecten funcionalidades existentes, como la sincronización de datos y la navegación de usuario.

**Pruebas No Funcionales:** Evaluar aspectos de rendimiento básico, como tiempos de respuesta en la autenticación y sincronización, que impactan la experiencia del usuario.

# 

# 

# 

# Configuraciones a Probar

**Firebase para Pruebas de Almacenamiento y Sincronización:** Validar que los datos de usuario se almacenen y sincronicen correctamente con Firebase en tiempo real.

**Datos de Prueba de Autenticación:** Crear usuarios de prueba con diferentes perfiles para validar casos de autenticación y registro.

**Configuraciones de Perfil Simplificadas:** Usar datos básicos en las preferencias de perfil y de configuración inicial, evitando complejidades que se implementarán en fases posteriores.

**Datos Simulados para Regresión:** Asegurarse de que cualquier cambio en autenticación no afecte otros datos, utilizando configuraciones de usuario básicas para probar la estabilidad de la sincronización.

# Nivel de Pruebas de Regresión

**Cobertura de Pruebas de Regresión Limitada:** Se ejecutarán pruebas de regresión en funcionalidades fundamentales (perfil, notificaciones básicas y sincronización con Firebase), evitando pruebas exhaustivas en áreas que no se modifican directamente en esta fase.

**Validación de Integridad de Funciones Clave:** Probar la integridad de funciones clave tras cada ajuste o actualización para asegurar que las funciones básicas continúen operando sin errores.

**Acceso a Firebase:** Configuración de un entorno de pruebas en Firebase para manipular y verificar datos sin impactar el entorno de producción.

Manejo de Errores: Captura y revisión de mensajes de error, especialmente en autenticación, para proporcionar retroalimentación clara al usuario en caso de fallos.

Criterios de Aceptación o Rechazo

 Criterios de Aceptación o Rechazo

**Cobertura Completa de Pruebas Funcionales**

Todas las pruebas funcionales deben completarse con al menos un 95% de éxito en los casos de prueba críticos, incluyendo autenticación, perfil y sincronización de datos en Firebase.

**Cobertura de Pruebas de Interfaces (UI)**

Comprobación de la interfaz para asegurar navegación sin errores en 100% de las pantallas principales.

Verificación de respuesta correcta de elementos interactivos (botones, enlaces) en al menos el 95% de los casos probados.

**Cobertura de Pruebas de Regresión**

Validación de que las funcionalidades existentes no se vean afectadas tras la implementación de la autenticación y perfil de usuario, con un 95% de éxito en las pruebas de regresión.

Los errores que aparezcan en estas pruebas deben corregirse o documentarse como riesgos asumidos en esta fase inicial.

**Estabilidad de Sincronización con Firebase**

Almacenamiento y recuperación de datos en Firebase debe completarse sin fallos en el 100% de las pruebas, garantizando la integridad de los datos del usuario.

**Porcentaje de Defectos Corregidos**

Todos los defectos críticos y de alta prioridad identificados durante las pruebas deben corregirse antes de la finalización del plan de pruebas.

Al menos el 90% de los defectos de prioridad media deben corregirse o marcarse para revisión en una fase futura si no impactan directamente la funcionalidad esencial.

**Pruebas de Rendimiento Básico en Autenticación**

Los tiempos de respuesta para inicio de sesión y registro deben mantenerse en menos de 2 segundos en dispositivos de prueba, para garantizar una experiencia fluida.

**Retroalimentación de Usuario Simulado**

Cualquier problema identificado durante pruebas de usabilidad con usuarios simulados debe ser evaluado y ajustado, especialmente en flujos de autenticación y configuración de perfil.

Criterios de Suspensión

**1. Defectos Críticos Bloqueantes**

**Condición:** La ejecución de las pruebas se detendrá si se identifica un defecto crítico que bloquee el acceso a la aplicación o funciones clave como el inicio de sesión o carga de la interfaz principal.

**Ejemplo de Criterio:** Si un error en la autenticación impide a los usuarios acceder a la aplicación.

**Acción:** Suspensión inmediata hasta que el equipo de desarrollo solucione el defecto.

**2. Alto Porcentaje de Casos Fallidos en Funciones Clave**

**Condición:** Las pruebas serán suspendidas si más del 30% de los casos de prueba en autenticación o sincronización de datos con Firebase fallan.

**Ejemplo de Criterio:** Fallos en inicio de sesión o problemas al recuperar y guardar datos en Firebase.

**Acción:** Suspensión de pruebas para la revisión y corrección de defectos en estas áreas.

**3. Fallas Repetitivas en Pruebas de Regresión**

**Condición:** Las pruebas se detendrán si se identifican fallas recurrentes en funcionalidades previamente estables, con un porcentaje de fallos superior al 25%.

**Ejemplo de Criterio:** Problemas persistentes en la navegación entre pantallas o en el acceso a configuraciones.

**Acción:** Suspensión hasta verificar y solucionar los errores en las funciones afectadas.

**4. Problemas de Rendimiento Grave**

**Condición:** Las pruebas se suspenderán si los tiempos de respuesta en funciones clave (inicio de sesión, carga de perfil, sincronización de datos) superan los 5 segundos de forma recurrente.

**Ejemplo de Criterio:** Retrasos considerables en la carga de datos del perfil de usuario.

**Acción:** Detención de pruebas y optimización del rendimiento en las funciones afectadas.

**5. Problemas Críticos en Seguridad**

**Condición:** La ejecución de las pruebas se detendrá si se detectan vulnerabilidades críticas en la seguridad de los datos del usuario o en los permisos de acceso a Firebase.

**Ejemplo de Criterio:** Fallos en la protección de datos personales que podrían comprometer la privacidad del usuario.

**Acción:** Suspensión inmediata hasta que se implementen las correcciones de seguridad necesarias.

Criterios de Reanudación

**1. Resolución de Defectos Críticos Bloqueantes**

**Condición:** Las pruebas pueden reanudarse una vez que los defectos críticos que impiden el acceso o afectan la funcionalidad clave (como autenticación o carga de la interfaz principal) han sido solucionados.

**Requisito:** El equipo de desarrollo debe confirmar que los defectos han sido corregidos y realizar pruebas de validación iniciales para verificar la funcionalidad.

**Evidencia:** Documentación de los errores corregidos, junto con los resultados de pruebas unitarias en el área afectada.

**2. Reducción de Casos Fallidos en Funciones Clave**

**Condición:** La reanudación de las pruebas será posible si el porcentaje de fallos en autenticación o sincronización de datos con Firebase se reduce a menos del 10%.

**Requisito:** El equipo de QA debe realizar una prueba de verificación para asegurar que los fallos se han corregido adecuadamente.

**Evidencia:** Reporte de casos de prueba exitosos en funciones clave, con registro de resultados corregidos.

**3. Estabilidad en Pruebas de Regresión**

**Condición:** Las pruebas se reanudarán cuando los fallos recurrentes en funciones previamente estables se reduzcan a menos del 10%.

**Requisito:** Las funciones afectadas deben pasar una prueba de regresión adicional para confirmar su estabilidad.

**Evidencia:** Informe de regresión exitoso con detalles de las correcciones y la estabilidad comprobada de cada función.

**4. Optimización de Rendimiento**

**Condición:** Las pruebas podrán reanudarse si los tiempos de respuesta en funciones críticas (inicio de sesión, sincronización de datos) se mantienen por debajo de los 2 segundos en dispositivos de prueba.

**Requisito:** El equipo de desarrollo debe realizar pruebas de rendimiento en el entorno de prueba para confirmar la mejora.

**Evidencia:** Resultados de pruebas de rendimiento documentados, mostrando tiempos de respuesta dentro de los límites establecidos.

**5. Garantía de Seguridad en el Manejo de Datos**

**Condición:** Las pruebas pueden reanudarse tras la resolución de vulnerabilidades críticas en la seguridad de los datos o permisos de acceso en Firebase.

**Requisito:** El equipo de desarrollo debe implementar y validar las correcciones de seguridad.

**Evidencia:** Informe de pruebas de seguridad que confirme la corrección de vulnerabilidades y la protección de los datos del usuario.

Entregables

* Documento de plan de pruebas
* Matriz de casos de pruebas

Recursos

# Requerimientos de Entornos – Hardware

* conectividad a internet
* mouse
* teclado
* pc

# 

# Requerimientos de Entornos – Software

**Entorno de Desarrollo y Pruebas**

**Ionic Framework:** Versión 7 para desarrollo y pruebas de la aplicación móvil, con soporte para Angular y sus dependencias.

**Angular CLI:** Última versión compatible para compilar y ejecutar la aplicación en modo de prueba.

**Node.js:** Versión 18 o superior, necesario para ejecutar y gestionar el entorno de desarrollo de Ionic y Angular.

**Firebase SDK:** Configurado para integrar las pruebas con Firebase, permitiendo el acceso a la base de datos y autenticación.

Accesos a Sistemas en Entorno de Pruebas

**Bases de Datos de Prueba**

**Instancia de Firebase Firestore:** Base de datos no relacional en Firebase para almacenar datos simulados del usuario, perfil y preferencias.

**Firebase Authentication:** Para gestionar y verificar las pruebas de inicio de sesión, registro y autenticación de usuarios en el entorno de pruebas.

**Herramientas para la Ejecución y Automatización de Pruebas**

**Postman:** Para probar y verificar manualmente las solicitudes a la API de Firebase y, si es necesario, a la API de Gemini.

**Jest o Jasmine:** Frameworks de pruebas unitarias e integración, configurados para Angular y compatibles con Ionic, para automatizar las pruebas.

**Google Chrome DevTools:** Para inspeccionar, depurar y probar la interfaz de usuario en un entorno de navegador simulado.

**Instalación de Software en Computadoras Asignadas**

**Visual Studio Code o IDE Similar:** Editor de código para acceder y editar el código fuente del proyecto.

**Git:** Para clonar el repositorio del proyecto y gestionar versiones durante las pruebas.

**Documentación y Gestión de Casos de Prueba**

**Jira :** Herramienta para la gestión y seguimiento de tareas de prueba, asignación de casos y reporte de incidencias.

**Google Sheets o Excel:** Para documentar resultados de pruebas y almacenar datos de pruebas manuales, si no se cuenta con un sistema de gestión de pruebas dedicado.

Herramientas de Pruebas Requeridas

**Herramientas de Pruebas Manuales**

**Google Chrome DevTools:** Herramienta de inspección para analizar y probar la interfaz de usuario en diferentes tamaños de pantalla y ajustar elementos visuales.

**Software para Pruebas de Rendimiento**

**Firebase Performance Monitoring:** Herramienta integrada en Firebase para monitorear el rendimiento de la aplicación en tiempo real, capturando métricas de carga y uso de la red en la aplicación móvil.

**Documentación de Resultados de Pruebas**

**Google Sheets o Microsoft Excel:** Para registrar manualmente resultados de pruebas, errores y observaciones, especialmente en pruebas que no se realicen mediante un software de gestión.

 Personal

* **1 Líder de Pruebas**
* **2 Analistas de Pruebas Funcionales**
* **1 Especialista en Automatización de Pruebas**
* **1 Especialista en Pruebas de Rendimiento y Seguridad**

Entrenamiento

**Capacitación en la Aplicación NUTRIMOVE**

**Objetivo:** Familiarizar al equipo de pruebas con las funcionalidades clave de la aplicación, como autenticación, perfil de usuario, recomendaciones nutricionales y ejercicio.

**Duración:** 1 día

**Contenido:**

* Revisión del flujo de usuario y navegación por la interfaz.
* Descripción de las funcionalidades y comportamiento esperado en cada módulo.

**Participantes:** Todo el equipo de pruebas.

**Entrenamiento en Herramientas de Gestión de Pruebas**

**Objetivo:** Asegurar que los analistas de pruebas y el líder de pruebas puedan usar eficientemente herramientas de gestión como Jira o Trello para documentar y asignar casos de prueba, y reportar incidencias.

**Duración:** 1 día

**Contenido:**

Creación y asignación de casos de prueba.

Seguimiento y documentación de defectos.

Generación de reportes de pruebas.

**Participantes:** Líder de Pruebas y Analistas de Pruebas Funcionales.

**Capacitación en Herramientas de Pruebas Automatizadas**

**Objetivo:** Capacitar al especialista en automatización en el uso de herramientas como Cypress, Protractor o Jest para desarrollar y ejecutar pruebas automatizadas.

**Duración:** 2 días

**Contenido:**

* Configuración de entornos de prueba automatizada en Angular/Ionic.
* Creación de scripts de prueba y mantenimiento de pruebas automatizadas.
* Ejecución de pruebas de regresión automatizadas.

**Participantes:** Especialista en Automatización de Pruebas.

**Entrenamiento en Pruebas de Rendimiento y Seguridad**

**Objetivo:** Entrenar al especialista en el uso de herramientas como Firebase Performance Monitoring para evaluar el rendimiento, y realizar pruebas de seguridad básica en la aplicación.

**Duración:** 1 día

**Contenido:**

* Monitoreo de rendimiento en Firebase.
* Pruebas de carga y optimización de tiempos de respuesta.
* Pruebas de seguridad en datos y autenticación.

**Participantes:** Especialista en Pruebas de Rendimiento y Seguridad.

Planificación y Organización

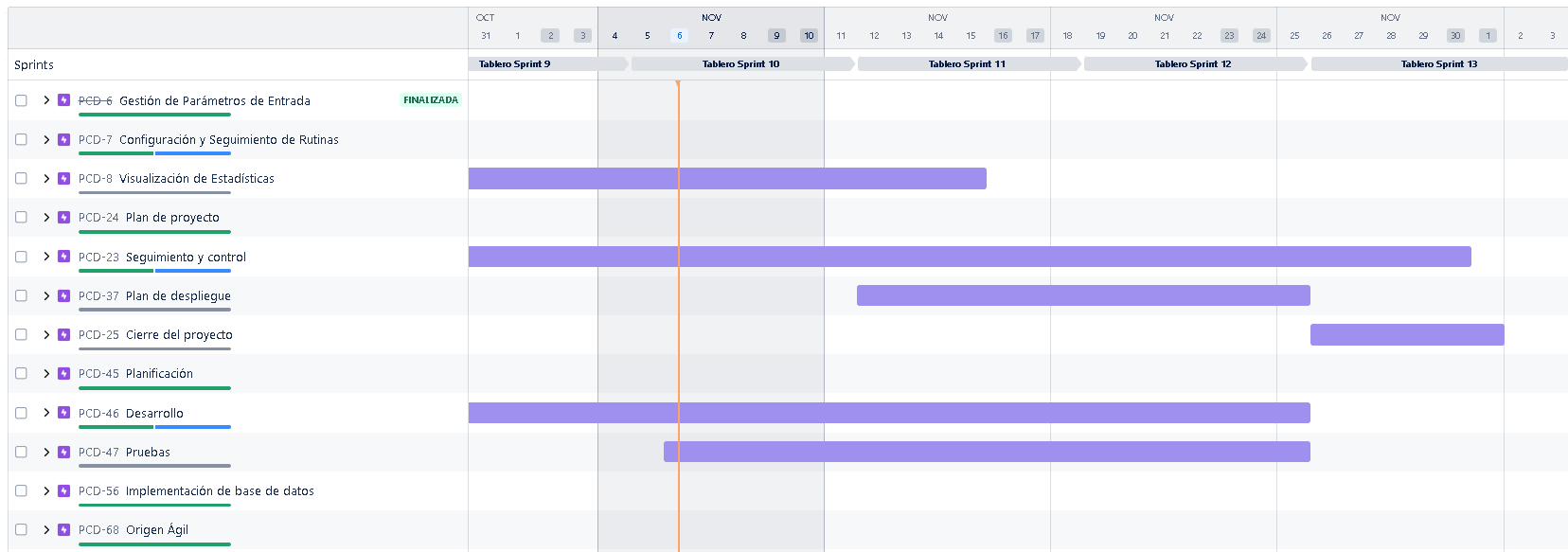
# Procedimientos para las Pruebas

Para la **metodología de pruebas nos basaremos en la matriz de casos de prueba**, esta plantilla será central en la creación, ejecución y seguimiento de los casos de prueba, asegurando que todos los aspectos del caso, desde entradas hasta resultados y dependencias, estén bien documentados. Usaremos esta estructura para definir claramente cada prueba y facilitar el análisis de resultados y el seguimiento de errores.

Matriz de Responsabilidades

| Actividad | Alejandro Vasquez | Felipe Prieto | Miguel Ravello | Patricio Catejo |
| --- | --- | --- | --- | --- |
| Definición del Plan de Pruebas | R | R | A | C |
| Preparación de Casos de Prueba | R | C | R | I |
| Ejecución de Pruebas Funcionales | R | R | C | I |
| Documentación y Reporte de Defectos | C | R | A | R |
| Automatización de Pruebas | R | C | R | I |
| Pruebas de Rendimiento y Seguridad | C | R | R | I |
| Revisión y Validación de Resultados | R | C | A | R |
| Reporte de Resultados Finales | C | R | A | R |
| Revisión de Criterios de Aceptación/Rechazo | R | R | C | A |
| Actualización de Documentación y Procedimientos | R | C | R | A |

Cronograma



Premisas

**Tiempo y Plazos**

* Las pruebas están programadas para completarse en 22 días. Revisiones semanales identificarán bloqueos para mantener el cronograma.

**Recursos Disponibles**

* Los miembros asignados (Alejandro Vasquez, Felipe Prieto, Miguel Ravello, Patricio Catejo) estarán disponibles según el cronograma.
* Se cuenta con acceso a dispositivos de prueba, emuladores y herramientas como BrowserStack y Android Studio.

**Metodología de Pruebas**

* Enfoque en pruebas funcionales, de regresión y de rendimiento básico.
* Metodología escalonada, comenzando con pruebas funcionales, seguidas de automatización y regresión.

**Herramientas de Pruebas y Gestión**

* Uso de Jira/Trello para gestionar y documentar casos de prueba y defectos.
* Cypress, Jest y Postman serán las herramientas principales parautomatización y pruebas manuales.

**Entorno de Pruebas en Firebase**

* Un entorno seguro y aislado de producción en Firebase estará disponible para pruebas.

**Pruebas de Seguridad y Rendimiento Básico**

* Se harán verificaciones de rendimiento y seguridad usando Firebase Performance Monitoring y Lighthouse, con posibles optimizaciones futuras.

**Corrección Prioritaria de Defectos Críticos**

* Los defectos críticos se corregirán de inmediato para avanzar en las pruebas y cumplir los plazos

Dependencias y Riesgos

**Dependencias con Desarrollos**

* **Descripción**: Las pruebas dependen de la finalización de desarrollos clave, como el módulo de autenticación y perfil.
* **Riesgo**: Retrasos en el desarrollo pueden demorar el inicio o continuidad de las pruebas.
* **Plan de Mitigación**: Coordinar con el equipo de desarrollo para obtener actualizaciones y probar módulos tan pronto estén listos.
* **Contingencia**: Reasignar tareas de prueba hacia componentes menos dependientes en caso de retraso.

**Dependencias con Otros Proyectos**

* **Descripción**: Otros proyectos en la organización pueden limitar los recursos compartidos (como equipos de prueba o herramientas).
* **Riesgo**: Posible escasez de recursos o interferencias en el acceso a herramientas.
* **Plan de Mitigación**: Programar el uso de recursos con anticipación y establecer acuerdos con otros proyectos.
* **Contingencia**: Utilizar entornos alternativos de prueba (emuladores locales) si los recursos compartidos no están disponibles.

**Disponibilidad de Recursos Humanos y Técnicos**

* **Descripción**: El equipo de pruebas depende de la disponibilidad continua de miembros y herramientas.
* **Riesgo**: Ausencia de personal clave o fallas en herramientas esenciales.
* **Plan de Mitigación**: Mantener una planificación clara de recursos y un sistema de respaldo para herramientas de prueba.
* **Contingencia**: Redistribuir responsabilidades en el equipo y usar herramientas alternativas en caso de ausencia o falla.

**Restricciones de Tiempo**

* **Descripción**: El cronograma de 22 días puede resultar ajustado si surgen problemas imprevistos.
* **Riesgo**: Falta de tiempo para completar todas las pruebas planificadas.
* **Plan de Mitigación**: Priorizar pruebas críticas y evaluar el progreso semanalmente para ajustes.
* **Contingencia**: Focalizarse en pruebas de alto impacto y aceptar el riesgo de dejar pruebas menores para una fase posterior.

**Premisas Inexactas o Incorrectas**

* **Descripción**: Alguna premisa inicial puede no cumplirse, como la estabilidad del entorno de pruebas en Firebase.
* **Riesgo**: El entorno de pruebas podría no estar disponible o funcionar incorrectamente, afectando las pruebas.
* **Plan de Mitigación**: Realizar pruebas preliminares en el entorno para validar su estabilidad.
* **Contingencia**: En caso de problemas en Firebase, usar bases de datos simuladas o locales para pruebas parciales.