**Plan de Pruebas de Software**

***[Proyecto “Skip”]***

***Fecha: [30/10/2024]***

**Tabla de contenido**

[**Resumen Ejecutivo 4**](#_heading=h.q1x03ye9pksn)

[**Alcance de las Pruebas 7**](#_heading=h.awwxxtv4jgxd)

[**Nuevas Funcionalidades a Probar 9**](#_heading=h.13au8em2lhc3)

[**Pruebas de Regresión 13**](#_heading=h.na92vfu232id)

[**Funcionalidades a No Probar 16**](#_heading=h.tykwtigxseqt)

[**Enfoque de Pruebas (Estrategia) 17**](#_heading=h.y0vupt3d27e5)

[**Criterios de Aceptación o Rechazo 20**](#_heading=h.y3nz4sg7u10c)

[**Criterios de Suspensión 23**](#_heading=h.fbj5mw7c4xnq)

[**Criterios de Reanudación 25**](#_heading=h.e63xz0uv7g7e)

[**Entregables 27**](#_heading=h.w8n2kdmzwt7l)

[**Requerimientos de Entornos – Hardware 29**](#_heading=h.44sinio)

[**Requerimientos de Entornos – Software 31**](#_heading=h.huof44tkrrfw)

[**Herramientas de Pruebas Requeridas 33**](#_heading=h.g9yykn20u3bg)

[**Personal 36**](#_heading=h.6mj731i7s0x6)

[**Entrenamiento 37**](#_heading=h.frmj6oowco4t)

[**Planificación y Organización 39**](#_heading=h.a482m6x6w99d)

[**Matriz de Responsabilidades 41**](#_heading=h.juduw1z83eiz)

[**Cronograma 42**](#_heading=h.5qxai4gxcjcc)

[**Premisas 42**](#_heading=h.8dkl3g45sczy)

[**Dependencias y Riesgos 44**](#_heading=h.rxc3usl2ijpx)

[**Referencias 46**](#_heading=h.o3b1q8pc97e)

[**Glosario 47**](#_heading=h.yyadwnqo4a1q)

**Historial de Versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 30-10-2024 | 1.0 | Lucas Carrasco | Skip | Plan de Pruebas de Proyecto Skip |
| 14-11-2024 | 1.1 | Lucas Carrasco | Skip | Caso de pruebas Proyecto Skip |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

| Empresa / Organización | Skip |
| --- | --- |
| Proyecto | Análisis Predictivo |
| Fecha de preparación | 30-10-2024 |
| Cliente | Rocio Contreras y Comité |
| Patrocinador principal | Duoc UC |
| Gerente / Líder de Proyecto | Lucas Carrasco |
| Gerente / Líder de Pruebas de Software | Lucas Carrasco |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Lucas Carrasco | Todas los Cargos | Gerente General | 30-10-2024 |  |

### Resumen Ejecutivo

**Propósito del Plan de Pruebas**

El propósito de este plan de pruebas es establecer un marco detallado para evaluar y validar el correcto funcionamiento y la precisión del sistema de Análisis Predictivo de Tendencias en Skip. Este documento tiene como objetivo definir las estrategias de prueba, establecer los recursos y metodologías necesarios, y garantizar que todas las funcionalidades críticas sean validadas antes del despliegue en producción. Se enfoca en asegurar que el sistema cumpla con los requisitos de negocio y técnicos, minimizando riesgos y garantizando un rendimiento óptimo y seguro.

**Tipo de Plan**

Este es un Plan Maestro de Pruebas, ya que abarca todas las fases y tipos de pruebas que se realizarán durante el ciclo de vida del desarrollo del sistema. Incluye tanto pruebas funcionales como no funcionales y abarca las diferentes etapas desde pruebas unitarias hasta pruebas de aceptación.

**Alcance del Plan de Pruebas**

El plan de pruebas cubre todos los módulos críticos del sistema de Análisis Predictivo, desde la ingesta y preprocesamiento de datos hasta la visualización de resultados en el dashboard, pasando por la implementación del modelo predictivo. Los módulos clave incluidos son:

1. **Ingesta y Preprocesamiento de Datos**: Pruebas de integración con sistemas de almacenamiento, manejo de datos faltantes y valores atípicos, y correcta transformación de variables categóricas.
2. **Modelo Predictivo**: Validación de precisión del modelo, análisis de hiperparámetros, y verificación de segmentación de clientes en categorías como "nuevos", "perdidos", "crecientes", entre otros.
3. **Visualización y Dashboard**: Pruebas de presentación de datos, interactividad del dashboard y usabilidad de las visualizaciones.
4. **Seguridad y Rendimiento**: Validación de medidas de seguridad y encriptación de datos, pruebas de carga y escalabilidad.

El alcance está alineado con el Plan de Proyecto de Software, cuyo objetivo es ofrecer un sistema robusto y eficiente para la predicción de comportamientos de clientes, proporcionando insights accionables a Skip.

**Restricciones**

El proyecto enfrenta algunas restricciones, tales como:

* **Recursos**: Disponibilidad limitada de datos históricos completos para ciertas variables, lo que podría afectar la precisión de los resultados.
* **Presupuesto**: Limitación en los recursos económicos destinados a la infraestructura de pruebas, lo que implica la necesidad de optimizar el uso de herramientas y entornos.
* **Tiempo**: Cronograma ajustado para las pruebas de rendimiento y de seguridad, que debe completarse antes de la fecha de entrega del proyecto.

**Alcance del Esfuerzo de Pruebas**

El esfuerzo de pruebas incluye un conjunto de pruebas que abarcan:

* **Pruebas Funcionales**: Validación de todas las funcionalidades del sistema, incluyendo la correcta segmentación de clientes y el procesamiento de datos en tiempo real.
* **Pruebas de Integración**: Evaluación de la interacción entre diferentes módulos del sistema, asegurando que la ingesta de datos, el modelo predictivo, y el dashboard funcionen correctamente en conjunto.
* **Pruebas de Seguridad**: Verificación de la protección de los datos de clientes, incluyendo encriptación y acceso controlado al sistema.
* **Pruebas de Rendimiento**: Pruebas de carga y estrés para asegurar que el sistema sea capaz de manejar grandes volúmenes de datos y usuarios concurrentes sin afectar el rendimiento.

**Otros Aspectos**

Este plan considera el uso de herramientas de pruebas automáticas y de análisis de datos como RStudio para las pruebas de precisión y efectividad del modelo. Además, se emplearán pruebas manuales para la verificación de la interfaz y la experiencia del usuario en el dashboard.

**Objetivos de Calidad**

La meta final es garantizar que el sistema de Análisis Predictivo de Tendencias ofrezca resultados precisos, visualizaciones claras y un rendimiento confiable, contribuyendo así a la toma de decisiones estratégicas en Skip. Al seguir la metodología CRISP-DM, el plan de pruebas garantiza que cada fase del ciclo de vida del proyecto ha sido validada con el objetivo de maximizar la calidad y confiabilidad del sistema.

### Alcance de las Pruebas

**Elementos de Pruebas**

Listado de todos los módulos, componentes o elementos que se van a probar. Si es de alto nivel, se listan las áreas funcionales (módulos o procesos que cubre el Testing)

**Listado de Módulos y Componentes para Pruebas**

**1. Módulo de Ingesta y Preprocesamiento de Datos**

* **Conexión a fuentes de datos**: Pruebas de integración con sistemas de almacenamiento de datos (PostgreSQL).
* **Extracción y carga de datos**: Validación de la correcta extracción y carga de datos desde las fuentes externas.
* **Limpieza de datos**: Verificación del manejo de datos faltantes, valores atípicos y duplicados.
* **Transformación de variables**: Pruebas para la conversión de variables categóricas a numéricas, normalización y estandarización de datos.
* **Verificación de integridad de datos**: Aseguramiento de que los datos están completos y sin errores antes de la entrada al modelo.

**2. Módulo de Modelado Predictivo**

* **Entrenamiento del modelo**: Evaluación de la correcta implementación de algoritmos de machine learning (regresión logística, árboles de decisión, etc.).
* **Selección y ajuste de hiperparámetros**: Validación de la optimización de parámetros para mejorar la precisión del modelo.
* **Evaluación del rendimiento del modelo**: Pruebas de métricas de desempeño, como R2, RMSE, accuracy, y precisión en la segmentación de clientes.
* **Validación cruzada y pruebas de generalización**: Asegurar que el modelo no está sobreajustado y se generaliza adecuadamente a nuevos datos.
* **Segmentación de clientes**: Pruebas de clasificación de clientes en segmentos específicos (nuevos, perdidos, decrecientes, crecientes, reactivados) y verificación de la precisión de esta segmentación.

**3. Módulo de Visualización y Dashboard**

* **Visualización de resultados**: Pruebas de presentación clara y precisa de los datos procesados y los resultados de la segmentación de clientes.
* **Interactividad del dashboard**: Validación de la interactividad, incluyendo filtros por fechas, categorías de clientes, y variables específicas.
* **Usabilidad y diseño de interfaz**: Pruebas de usabilidad para asegurar que los tomadores de decisiones puedan navegar fácilmente y encontrar insights clave.
* **Gráficas y tablas**: Aseguramiento de que las visualizaciones (gráficos de líneas, barras, tablas) se actualizan correctamente y representan datos precisos.
* **Actualización en tiempo real**: Verificación de la actualización dinámica del dashboard conforme a nuevos datos o cambios en los parámetros de visualización.

**4. Módulo de Seguridad y Privacidad de Datos**

* **Encriptación de datos**: Validación de que los datos sensibles están encriptados tanto en tránsito como en reposo.
* **Control de acceso**: Pruebas de autenticación y autorización para garantizar que solo usuarios autorizados puedan acceder al sistema y a los datos de clientes.
* **Cumplimiento de normativas de privacidad**: Aseguramiento de que el sistema cumple con las regulaciones de protección de datos y privacidad aplicables a Skip.

**5. Módulo de Rendimiento y Escalabilidad**

* **Pruebas de carga**: Evaluación de cómo el sistema maneja grandes volúmenes de datos y varios usuarios simultáneos sin pérdida de rendimiento.
* **Pruebas de estrés**: Verificación de la estabilidad del sistema bajo condiciones extremas, asegurando que no haya fallos o tiempos de respuesta excesivos.
* **Pruebas de tiempos de respuesta**: Aseguramiento de que las consultas y el procesamiento de datos se realicen en tiempos aceptables para los usuarios.

**6. Módulo de Integración y Comunicaciones**

* **Integración entre módulos**: Validación de la interacción entre el módulo de ingesta de datos, el modelo predictivo y el dashboard.
* **Comunicaciones entre API Librerías R y bases de datos**: Pruebas para asegurar que la API Librerías R puede comunicar eficientemente los resultados del modelo al dashboard y que las consultas de datos funcionan sin errores.
* **Sincronización de datos**: Verificación de que los datos se actualicen y sincronicen adecuadamente entre los distintos módulos.

### Nuevas Funcionalidades a Probar

Es un listado de lo que se va a probar “Desde el punto de vista del Usuario”.

**Listado de Funcionalidades para Pruebas desde el Punto de Vista del Usuario**

**1. Acceso a la Plataforma y Seguridad**

* **Inicio de Sesión Seguro**: El usuario debe poder iniciar sesión con su nombre de usuario y contraseña, asegurando que solo personas autorizadas acceden a los datos.
* **Recuperación de Contraseña**: Posibilidad de restablecer la contraseña en caso de olvido, garantizando un proceso simple y seguro.
* **Roles y Permisos**: Verificación de que los usuarios tengan acceso limitado según su rol, mostrando solo la información y funcionalidades que les correspondan.

**2. Visualización de Datos y Resultados**

* **Dashboard Interactivo**: El usuario debe ver un dashboard fácil de usar, donde pueda navegar por los datos de manera intuitiva y visualizar los resultados en tiempo real.
* **Filtros Personalizables**: Posibilidad de filtrar los datos por diferentes criterios (por ejemplo, tipo de cliente, fecha) para personalizar la vista y obtener insights específicos.
* **Gráficos Informativos**: Verificación de que los gráficos, como barras, líneas y tablas, sean fáciles de interpretar y muestren información clara y relevante.
* **Resumen de Insights Clave**: Pruebas de la funcionalidad que resalta puntos clave o insights importantes, como tendencias de clientes crecientes o decrecientes, de manera destacada para facilitar la toma de decisiones.

**3. Segmentación de Clientes**

* **Clasificación de Clientes**: Validación de que el sistema clasifique a los clientes en segmentos específicos (nuevos, perdidos, crecientes, decrecientes, reactivados), permitiendo al usuario visualizar los segmentos y su evolución en el tiempo.
* **Alertas de Comportamiento**: Pruebas para asegurar que el sistema emita alertas o notificaciones cuando se detecten cambios significativos en los segmentos de clientes, como el aumento de clientes perdidos o una tendencia en crecimiento.

**4. Análisis Predictivo de Comportamiento**

* **Predicciones de Tendencias**: El usuario debe poder ver las predicciones sobre el comportamiento futuro de los clientes, basadas en patrones detectados en los datos históricos.
* **Visualización de Resultados de Predicción**: Pruebas para verificar que los resultados de las predicciones sean claros y permitan al usuario entender fácilmente las tendencias anticipadas, como posibles aumentos o disminuciones en el uso de ciertos productos.
* **Simulación de Escenarios**: Pruebas de la funcionalidad que permite a los usuarios ajustar parámetros (por ejemplo, promociones o descuentos) y ver cómo estos cambios impactarían en la predicción de tendencias.

**5. Navegación y Experiencia de Usuario**

* **Interfaz Intuitiva**: Evaluación de la facilidad de uso y navegación de la plataforma, asegurando que los usuarios puedan acceder rápidamente a las distintas funcionalidades sin capacitación técnica.
* **Tiempo de Respuesta**: Pruebas para asegurar que el sistema responde rápidamente a las consultas y cambios de filtros, sin retrasos perceptibles para el usuario.
* **Consistencia en el Diseño**: Verificación de que el diseño visual (colores, estilos, iconos) sea coherente y permita una experiencia uniforme a través de todas las secciones de la plataforma.

**6. Actualización y Precisión de los Datos**

* **Datos en Tiempo Real**: Pruebas para asegurar que los datos en el dashboard se actualicen en tiempo real o con una frecuencia adecuada para reflejar cambios recientes en el comportamiento de los clientes.
* **Precisión de Datos**: Aseguramiento de que los datos mostrados en el sistema son precisos, eliminando errores de redondeo, duplicados o inconsistencias que puedan confundir al usuario.
* **Notificación de Actualizaciones de Datos**: Validación de una funcionalidad que notifique al usuario cuando se actualicen datos relevantes, permitiéndole saber cuándo están disponibles los datos más recientes.

**7. Exportación de Informes y Datos**

* **Generación de Reportes**: Pruebas para que el usuario pueda generar reportes personalizados en formatos como CSV, incluyendo los datos y gráficos seleccionados.
* **Exportación de Datos Filtrados**: Verificación de que el usuario pueda exportar datos según los filtros aplicados, para compartir o analizar en otras herramientas.
* **Resumen Ejecutivo**: Pruebas de una función que permita descargar un resumen ejecutivo con insights y tendencias clave, facilitando la comunicación con otros equipos o directivos.

### 

### Pruebas de Regresión

Listado de las funcionalidades no directamente involucradas en el desarrollo.

| Fase Proyecto (izquierdo) | Fase Pruebas (derecho) | Pruebas Realizadas | Pruebas | Resultado Obtenido |
| --- | --- | --- | --- | --- |
| Inicio del Proyecto | Pruebas de accesibilidad y seguridad de la plataforma | 100 | Cumplimiento de accesibilidad y seguridad | 100% |
| Comprensión de los Datos | Pruebas de interfaz y navegación para facilidad de uso | 100 | Facilidad de uso y navegación intuitiva | 100% |
| Análisis de los Datos | Pruebas de gestión y carga para manejar grandes volúmenes de datos | 100 | Manejo de grandes volúmenes de datos | 100% |
| Preparación de los Datos | Pruebas de notificaciones para alertas en tiempo real | 100 | Alertas en tiempo real | 100% |
| Modelado | Pruebas de reportes y exportación para precisión y análisis externo | 100 | Precisión y facilidad de exportación | 100% |
| Evaluación | Pruebas de actualización y rendimiento para eficiencia en tiempo real | 100 | Eficiencia en tiempo real | 100% |
| Despliegue | Pruebas de usabilidad y experiencia de usuario final | 200 | Alta satisfacción en experiencia de usuario | 100% |
| Monitoreo y Ajustes Finales | Pruebas de resiliencia del sistema | 200 | Capacidad de recuperación sin pérdida de datos | 100% |

### Funcionalidades a No Probar

Listado de las funcionalidades que NO se van a probar. Debe incluir información de las razones por las cuales no se van a probar y los riesgos que se están asumiendo.

**Funcionalidades Experimentales en Modelos Predictivos ( Explicabilidad del Modelo)**

* **Razón**: El enfoque está en precisión predictiva y no en explicabilidad o interpretabilidad de los modelos.
* **Riesgos**: Los usuarios no podrán entender completamente las razones detrás de las predicciones, lo cual puede afectar la confianza en los resultados.

**Compatibilidad con Navegadores Obsoletos (por ejemplo, Internet Explorer)**

* **Razón**: La plataforma está optimizada para navegadores modernos y el uso de navegadores obsoletos está en desuso en los entornos corporativos actuales.
* **Riesgos**: Usuarios que aún usen estos navegadores no podrán acceder a la plataforma, lo cual podría afectar a algunos usuarios corporativos que no hayan actualizado su software.

**Pruebas de Configuración de Red Avanzada (VPNs, Firewalls Personalizados)**

* **Razón**: La plataforma se utilizará principalmente en redes corporativas con configuraciones estándar.
* **Riesgos**: Usuarios que intenten acceder desde redes con configuraciones de seguridad altamente personalizadas podrían enfrentar problemas de acceso o restricciones inesperadas.

### Enfoque de Pruebas (Estrategia)

**Estrategia de Pruebas**

La estrategia de pruebas se definirá dentro del plan de pruebas para abarcar los tipos de pruebas, configuraciones, requisitos especiales, subconjuntos de datos y nivel de pruebas de regresión necesarias para asegurar la calidad y el rendimiento del sistema de análisis predictivo.

**1. Tipos de Pruebas**

* **Pruebas Funcionales**: Se realizarán pruebas funcionales para verificar que cada funcionalidad implementada cumpla con los requisitos del usuario, como la segmentación de clientes, predicciones de comportamiento y visualización de tendencias. Estas pruebas verificarán la correcta ejecución de las funcionalidades principales desde el punto de vista del usuario.
* **Pruebas de Desempeño**: Se evaluará el rendimiento del sistema bajo diferentes cargas de usuarios para asegurar tiempos de respuesta adecuados, especialmente en períodos de alta demanda. Se medirán métricas como tiempo de respuesta, uso de memoria y CPU.
* **Pruebas de Interfaces**: Se probarán las interacciones entre el sistema de análisis y los sistemas de terceros (PostgreSQL plataformas de visualización como RYR Studio) para asegurar que los datos fluyan correctamente y sin errores.
* **Pruebas de Seguridad**: Se incluirán pruebas de seguridad para proteger los datos sensibles de los clientes y asegurar que solo los usuarios autorizados puedan acceder al sistema.
* **Pruebas de Usabilidad**: Se evaluará la experiencia del usuario para garantizar que el sistema sea intuitivo y fácil de usar, con interfaces y gráficos claros y una navegación fluida.
* **Pruebas de Regresión**: Se realizarán pruebas de regresión en cada ciclo de desarrollo para asegurar que las nuevas funcionalidades no afecten negativamente las funcionalidades ya implementadas y verificadas.
* **Pruebas de Compatibilidad**: Se probará la compatibilidad del sistema en diferentes navegadores modernos (Chrome, Firefox, Safari, Edge) y dispositivos (PC, tablets, smartphones).

**2. Requerimientos Especiales de las Pruebas**

* **Datos de Pruebas**: Se requiere una base de datos de prueba representativa con datos históricos de clientes de Skip. Estos datos deben incluir registros de uso de productos como cupones electrónicos, TCT, TAE, y patrones de consumo que permitan evaluar las predicciones del sistema.
* **Acceso a Sistemas Externos**: Se necesita acceso a PostgreSQL para pruebas de integración. También se requiere acceso controlado a plataformas de visualización de datos (RYR Studio) para evaluar la visualización de resultados.
* **Entorno de Pruebas Aislado**: Las pruebas se realizarán en un entorno controlado, separado del entorno de producción, para evitar interferencias con los datos reales de la empresa.

**3. Configuraciones a Probar**

* **Navegadores**: Google Chrome, Mozilla Firefox, Microsoft Edge y Safari en sus versiones actuales.
* **Sistemas Operativos**: Windows 10 y 11, macOS, Android y iOS.
* **Dispositivos**: PCs, tablets y smartphones para asegurar la responsividad de la interfaz.
* **Conexiones de Red**: Pruebas en conexiones de alta velocidad (corporativas) y en redes móviles (4G y 5G) para verificar el rendimiento en diferentes condiciones de conectividad.

**4. Subconjuntos de Datos**

* **Datos Históricos Segmentados**: Se utilizarán subconjuntos de datos que representen diferentes segmentos de clientes (nuevos, perdidos, decrecientes, crecientes, reactivados) para evaluar la precisión de las predicciones.
* **Muestras de Alta Variabilidad**: Conjunto de datos con un amplio rango de consumo y uso de productos para asegurar que el modelo predictivo sea robusto y no esté sobreajustado a un tipo específico de comportamiento.
* **Datos de Pruebas para Casos Límites**: Subconjuntos de datos con patrones extremos, como clientes con muy altos o muy bajos niveles de consumo, para evaluar cómo responde el modelo a casos atípicos.

**5. Nivel de Pruebas de Regresión**

* **Regresión Total**: En cada nueva versión del sistema, se realizarán pruebas de regresión completas para verificar que las nuevas funcionalidades no afecten el funcionamiento de las ya existentes.
* **Regresión Selectiva**: Después de modificaciones menores, se aplicarán pruebas de regresión en módulos específicos, como la visualización de datos o la predicción de segmentos de clientes, asegurando que no se generen errores en esas áreas.
* **Automatización de Pruebas de Regresión**: Se automatizarán las pruebas de regresión para los módulos críticos, como la predicción de tendencias y la integración de datos. Esto permitirá identificar errores recurrentes rápidamente y mejorar la eficiencia del proceso de pruebas.

### Criterios de Aceptación o Rechazo

**Criterios de Aceptación o Rechazo**

Son los criterios que serán considerados para dar por completado el Plan de Pruebas de Software.

**Criterios de Finalización del Plan de Pruebas**

1. **Cobertura de Pruebas Unitarias**
   * Completar el 100% de las pruebas unitarias de los módulos clave, especialmente aquellos involucrados en el análisis predictivo, segmentación de clientes y visualización de tendencias.
   * Asegurar una cobertura de código mínima del 85% en pruebas unitarias para identificar y corregir cualquier posible error en el nivel de componentes individuales.
2. **Cobertura de Casos de Pruebas Funcionales**
   * Ejecutar el 100% de los casos de prueba funcionales para verificar la operatividad y funcionalidad de todas las características principales desde el punto de vista del usuario.
   * Alcanzar un porcentaje de éxito de al menos el 95% en los casos de prueba funcionales, permitiendo hasta un 5% de casos con defectos menores (que no afecten la funcionalidad crítica).
3. **Cobertura de Pruebas de Integración y de Interfaces**
   * Realizar pruebas de integración completas para cada sistema externo involucrado (PostgreSQL, RYR Studio).
   * Verificar la correcta comunicación entre los módulos del sistema y asegurar que los datos fluyan sin errores entre diferentes componentes.
   * Completar el 100% de las pruebas de interfaz, especialmente en los puntos críticos donde interactúan los sistemas externos con el sistema de análisis predictivo.
4. **Pruebas de Desempeño**
   * Cumplir con los requisitos de tiempo de respuesta establecidos (por ejemplo, menos de 2 segundos de tiempo de carga para la visualización de resultados en situaciones normales de carga).
   * Pasar las pruebas de carga con un mínimo de 90% de éxito en casos simulando el uso de múltiples usuarios simultáneos.
5. **Porcentaje de Defectos Corregidos**
   * Identificar y corregir el 100% de los defectos críticos y altos antes de dar por completado el plan de pruebas.
   * Resolver al menos el 95% de los defectos de prioridad media.
   * Registrar y documentar los defectos de baja prioridad para futuras versiones, siempre que no afecten la estabilidad ni la experiencia del usuario final.
6. **Cobertura de Requisitos**
   * Completar pruebas para el 100% de los requisitos definidos en el plan de desarrollo, tanto funcionales como no funcionales.
   * Asegurar que cada requisito tenga al menos un caso de prueba asociado y aprobado.
7. **Ejecución de Pruebas de Regresión**
   * Ejecutar pruebas de regresión completas y pasar con un 90% de éxito en los módulos principales, garantizando que las nuevas implementaciones no afecten funcionalidades previamente aprobadas.
8. **Pruebas de Seguridad y Usabilidad**
   * Completar el 100% de los casos de prueba de seguridad, asegurando que el sistema esté libre de vulnerabilidades críticas.
   * Realizar pruebas de usabilidad y obtener al menos un 90% de satisfacción en los resultados de las pruebas de experiencia de usuario.
9. **Aprobación de Resultados por el Equipo de QA y Stakeholders**
   * Obtener la aprobación final del equipo de QA y stakeholders principales, asegurando que todos los módulos críticos cumplan con los estándares de calidad esperados y que los objetivos del proyecto se hayan alcanzado.

### Criterios de Suspensión

Establece claramente bajo qué condiciones se detienen un conjunto de casos de pruebas.

**Condiciones de Detención de Casos de Pruebas**

1. **Defectos Críticos Bloqueantes**
   * Si se identifica un defecto crítico que bloquea la funcionalidad principal (por ejemplo, la incapacidad de cargar o procesar datos de clientes, fallos en la segmentación de usuarios, o errores en la predicción de tendencias) y este impide la ejecución de pruebas adicionales en otros módulos, se detendrán todos los casos de prueba relacionados hasta que el defecto sea corregido y verificado.
2. **Porcentaje de Casos de Prueba Fallidos**
   * Si más del 30% de los casos de prueba en un módulo o funcionalidad específica fallan durante la ejecución inicial, se pausará el avance de los casos de prueba para evaluar el alcance del problema. El equipo de desarrollo deberá corregir los defectos, y los casos de prueba deberán repetirse antes de continuar con la prueba de otros módulos.
3. **Fallos en Pruebas de Integración y de Interfaces**
   * Si se encuentran defectos que afectan la comunicación entre sistemas (por ejemplo, errores de integración con PostgreSQL, o RYR Studio), se detendrán las pruebas de integración y de interfaz. Solo se reanudarán cuando se haya solucionado el problema y validado la correcta interacción entre los sistemas.
4. **Incumplimiento en Pruebas de Desempeño**
   * Si el sistema no cumple con los criterios de desempeño en más del 20% de los casos (como tiempos de respuesta excesivos o errores al procesar grandes volúmenes de datos), se detendrán las pruebas de desempeño. Las pruebas se reanudarán una vez que se optimice el sistema y cumpla con los tiempos de respuesta requeridos.
5. **Defectos de Seguridad Críticos**
   * Si se detectan vulnerabilidades críticas que comprometan la seguridad del sistema o de los datos de los usuarios, todas las pruebas se detendrán hasta que los problemas de seguridad se solucionen y el sistema pase una nueva revisión de seguridad.
6. **Recursos Insuficientes**
   * Si durante el proceso de pruebas se llega a un punto donde faltan recursos críticos, como hardware, tiempo o personal de QA, que impidan continuar con la ejecución del plan de pruebas en su totalidad, las pruebas se suspenderán temporalmente hasta que se disponga de los recursos necesarios.
7. **Limitaciones en el Ambiente de Pruebas**
   * Si se detectan limitaciones en el entorno de pruebas (como bases de datos incompletas, problemas de configuración en servidores, o datos inconsistentes) que afecten los resultados, se detendrán las pruebas hasta que el entorno esté en condiciones adecuadas y haya sido validado para su continuidad.
8. **Exceso de Defectos Menores**
   * Si, al alcanzar un acumulado del 50% de defectos menores, se observa una tendencia constante de fallos en los mismos componentes, se detendrán las pruebas para realizar un análisis exhaustivo y una revisión de la calidad general del módulo afectado.

### Criterios de Reanudación

Luego de haber suspendido las pruebas, aquí se establece bajo qué criterios se reanudarán.

**Criterios para la Reanudación de Pruebas**

1. **Corrección de Defectos Críticos**
   * Los defectos críticos o bloqueantes identificados, que impedían la ejecución de otros casos de prueba, deben ser corregidos y validados en un entorno controlado. Esto incluye la validación de que la funcionalidad afectada opera correctamente sin impacto en otras áreas del sistema. Una vez confirmado, se reanudarán las pruebas en los módulos afectados.
2. **Reevaluación y Cumplimiento de Porcentaje de Casos Exitosos**
   * Si la suspensión fue debido a un alto porcentaje de casos fallidos, se deben corregir los defectos reportados y alcanzar un nivel de éxito superior al 70% en el módulo o funcionalidad específica. Solo tras cumplir este criterio se reanudarán las pruebas en ese módulo.
3. **Validación de Integración y Funcionalidad de Interfaces**
   * En caso de problemas de integración con otros sistemas, las pruebas se reanudarán cuando se verifique que la comunicación y transferencia de datos entre sistemas (como PostgreSQL, o RYR Studio) es estable y funcional, y que todos los datos están siendo procesados correctamente.
4. **Cumplimiento de Criterios de Desempeño**
   * Si las pruebas se suspendieron debido a problemas de desempeño, se deben realizar optimizaciones en el sistema hasta alcanzar los tiempos de respuesta esperados. Las pruebas se reanudarán solo cuando se verifique que el sistema cumple con los estándares de rendimiento bajo carga y en condiciones normales de operación.
5. **Solución de Vulnerabilidades de Seguridad**
   * Para reanudar las pruebas suspendidas debido a problemas de seguridad, se debe realizar una revisión exhaustiva y validación de que todas las vulnerabilidades críticas han sido corregidas. Se requiere la aprobación del equipo de seguridad para confirmar que el sistema es seguro para continuar con las pruebas.
6. **Disponibilidad de Recursos Necesarios**
   * Si la suspensión se debió a la falta de recursos (personal, tiempo, infraestructura), se reanudarán las pruebas una vez que se cuente con los recursos necesarios. Esto incluye asegurar que el equipo de QA tiene suficiente apoyo para llevar a cabo todas las pruebas restantes según el plan.
7. **Validación del Entorno de Pruebas**
   * En caso de limitaciones en el ambiente de pruebas, las pruebas se reanudarán una vez que el entorno haya sido completamente restaurado, incluyendo la integridad de las bases de datos y la configuración de los servidores. Es necesario verificar que el entorno de pruebas esté alineado con los requisitos para la ejecución de casos de prueba sin interrupciones.
8. **Revisión de Defectos Menores**
   * Si se suspendieron las pruebas debido a un exceso de defectos menores, se realizará una revisión exhaustiva y corrección de estos problemas, seguida de una validación para asegurar que no afectan la estabilidad del sistema. Al alcanzar un porcentaje aceptable de defectos menores, las pruebas podrán reanudarse.

### Entregables

Establece que se entregará como parte de la ejecución del plan

**Entregables del Plan de Pruebas**

1. **Documento de Plan de Pruebas**
   * Describe los objetivos, alcance, estrategia de pruebas, criterios de inicio y reanudación, cronograma, y todos los aspectos necesarios para la ejecución efectiva del plan.
2. **Casos de Pruebas**
   * Listado detallado de todos los casos de prueba diseñados, que incluye la descripción de cada prueba, el procedimiento a seguir, y los datos de entrada esperados. Incluye tanto pruebas funcionales como no funcionales.
3. **Especificación de Diseño de Casos de Prueba**
   * Documento que detalla el diseño de los casos de prueba, incluyendo objetivos específicos, precondiciones, y criterios de éxito/fallo para cada caso.
4. **Evidencias de Pruebas**
   * Capturas de pantalla, videos, y documentos de respaldo que muestren los resultados de cada caso de prueba ejecutado, tanto para casos exitosos como para fallidos.
5. **Logs de Errores**
   * Registro de todos los errores encontrados durante las pruebas, incluyendo detalles técnicos como trazas de error, mensajes de sistema, y contexto en el que se produjo cada error. Estos logs facilitarán el análisis y corrección de problemas.
6. **Reportes de Incidencias**
   * Reportes detallados que documentan cada defecto o incidencia encontrada, incluyendo su gravedad, impacto, prioridad, pasos para reproducir el error, y asignación a responsables para su resolución.
7. **Reportes de Resultados de Pruebas**
   * Informes periódicos que resumen los resultados de las pruebas ejecutadas, incluyendo métricas de casos exitosos y fallidos, porcentajes de cobertura de pruebas, y detalles de cualquier problema significativo que afecte el avance.
8. **Reportes Emitidos por Herramientas de Pruebas**
   * Reportes generados automáticamente por herramientas de pruebas, como logs de rendimiento, análisis de cobertura de código, resultados de pruebas automatizadas, entre otros.
9. **Reporte Final de Pruebas**
   * Informe consolidado que resume todos los resultados de las pruebas, análisis de la efectividad de los casos de prueba, métricas clave (como porcentaje de cobertura y tasa de defectos corregidos), y una evaluación final sobre la calidad del sistema en base a los resultados obtenidos.
10. **Evidencia de Pruebas de Regresión**
    * Documentación de la ejecución de pruebas de regresión en las funcionalidades existentes, confirmando que las correcciones y modificaciones no han afectado la estabilidad de componentes previamente probados.
11. **Checklist de Configuraciones de Entorno**
    * Listado de configuraciones específicas del entorno de pruebas (bases de datos, sistemas integrados, configuraciones de red, etc.) para asegurar que todos los casos se ejecutaron bajo las condiciones esperadas.

**Recursos**

### Requerimientos de Entornos – Hardware

Lista de los requerimientos de equipos, hardware y red necesarios para completar las actividades del Plan de Pruebas de Software

1. **Servidor de Pruebas**
   * **CPU**: Procesador de al menos 4 núcleos
   * **Memoria RAM**: 16 GB para manejar volúmenes de datos moderados
   * **Almacenamiento**: SSD de 512 GB para acceso rápido a datos, más un disco HDD de 1 TB para almacenamiento de datos históricos y resultados de pruebas
   * **Sistema Operativo**: Windows 10 Pro o Ubuntu 20.04 LTS
2. **Estaciones de Trabajo para QA**
   * **CPU**: Procesador de 4 núcleos
   * **Memoria RAM**: 8 GB para ejecutar herramientas de pruebas y simulaciones
   * **Almacenamiento**: SSD de 256 GB para almacenamiento local de herramientas y datos de prueba temporales
   * **Monitor**: Pantalla de resolución estándar (1366x768 o superior)
   * **Sistema Operativo**: Windows 10 Pro o Ubuntu 18.04 LTS
3. **Servidor de Base de Datos**
   * **CPU**: Procesador de 8 núcleos
   * **Memoria RAM**: 16 GB, suficiente para consultas moderadas y procesamiento de datos históricos
   * **Almacenamiento**: SSD de 512 GB para datos de alto rendimiento
   * **Sistema Operativo**: Ubuntu 18.04 LTS o Windows Server 2016
4. **Entorno de Pruebas de Desempeño**
   * **CPU**: Procesador de 4 núcleos para pruebas de carga ligera
   * **Memoria RAM**: 16 GB para simular un número reducido de usuarios concurrentes
   * **Almacenamiento**: SSD de 256 GB para almacenamiento temporal de resultados
   * **Sistema Operativo**: Windows 10 Pro o Ubuntu 18.04 LTS
5. **Dispositivos Adicionales**
   * **Conexión a Internet**: 100 Mbps para asegurar la comunicación entre los servidores de pruebas y estaciones de trabajo
   * **Backup Storage**: Sistema de respaldo externo con capacidad de 2 TB para copias de seguridad semanales de los datos de prueba y resultados
   * **Red de Pruebas Aislada**: Red virtualizada o enrutada dentro de una LAN existente, para realizar pruebas en un entorno seguro

### Requerimientos de Entornos – Software

Lista de los requerimientos de software necesarios para completar las actividades de prueba

**1. Accesos a Sistemas**

* **Entorno de Pruebas de PostgreSQL**: Acceso a una base de datos de pruebas en PostgreSQL, que replica la estructura de la base de datos de producción. Esto permitirá realizar consultas y pruebas sin afectar los datos reales.
* **Entorno de Pruebas de la Interfaz Web (Shiny)**: Acceso al entorno de pruebas de la interfaz desarrollada con Shiny y Shinydashboard, para verificar la funcionalidad de los dashboards y su interactividad.
* **Sistema de Gestión de Versiones (GitHub)**: Acceso al repositorio de GitHub donde se almacenan los scripts, configuraciones y documentación del proyecto, para facilitar la revisión y la colaboración en el código.

**2. Bases de Datos**

* **Conexión a Base de Datos de Pruebas**: Configuración de acceso seguro a una base de datos de pruebas en PostgreSQL, para realizar pruebas sin comprometer la integridad de los datos de producción.
* **Permisos de Lectura y Escritura Limitada**: En la base de datos de pruebas, los testers necesitan permisos para leer, escribir, y realizar pruebas de inserción o actualización, con datos que se puedan resetear.

**3. Instalación de Software en Computadores Asignados a Testers**

* **R y RStudio**: Entorno de desarrollo de R para ejecutar y probar los scripts de análisis y visualización de datos.
* **Paquetes de R**:
  + **shiny y shinydashboard**: Para probar la interfaz web y los dashboards.
  + **RPostgres**: Para verificar la conexión entre R y PostgreSQL.
  + dplyr, lubridate, readr: Necesarios para pruebas de limpieza y manipulación de datos.
  + **caret, rpart y ggplot2:** Para la validación de los modelos predictivos y la generación de gráficos de prueba.
* **Cliente de GitHub**: Para gestionar y sincronizar cambios en el repositorio de pruebas y garantizar que los testers trabajen con las últimas versiones del código.
* **Herramientas de Gestión de Incidencias**: Como JIRA o Trello, para el seguimiento de errores e incidencias encontrados durante las pruebas.

**4. Otros Requerimientos**

* **Permisos de Usuario**: Para los accesos específicos de entorno de pruebas, incluyendo autenticación para PostgreSQL y permisos de conexión a los dashboards de Shiny.
* **Documentación Técnica de la Base de Datos**: Para comprender la estructura de datos y facilitar la creación de pruebas basadas en datos específicos.

### Herramientas de Pruebas Requeridas

**1. Herramientas de Automatización de Pruebas**

* **Selenium**: Utilizado para la automatización de pruebas de la interfaz web desarrollada con Shiny y Shinydashboard, permitiendo simular interacciones de usuario y verificar el comportamiento de los dashboards en distintos navegadores.
* **R Scripts Automatizados**: Scripts en R diseñados para ejecutar pruebas repetitivas en el análisis de datos y modelado predictivo. Estos scripts evaluarán la precisión y el rendimiento del modelo en distintos subconjuntos de datos.

**2. Software de Gestión de Pruebas**

* **TestRail**: Herramienta de gestión de pruebas que permite organizar, documentar y hacer seguimiento a los casos de prueba, además de registrar los resultados de cada ejecución y detectar tendencias en las incidencias reportadas.
* **JIRA**: Utilizado para la gestión de incidencias y seguimiento de defectos. JIRA permite documentar errores encontrados en las pruebas, asignar responsables y dar seguimiento hasta su resolución.

**3. Herramientas de Análisis y Validación de Datos**

* **Data Validation Scripts en R**: Scripts específicos en R para la validación de integridad y consistencia de datos, asegurando que los datos utilizados en el modelo predictivo sean correctos y cumplan con los estándares de calidad.
* **pgAdmin (PostgreSQL)**: Herramienta de administración y consulta de bases de datos PostgreSQL para validar las consultas de extracción de datos, revisar integridad y realizar pruebas en las tablas de datos del proyecto.

**4. Herramientas de Registro y Monitoreo de Pruebas**

* **Log Analyzer (Paquete log4r en R)**: Utilizado para crear y analizar logs generados por los scripts de R durante las pruebas, permitiendo la revisión detallada de errores y la identificación de problemas en el flujo de procesamiento de datos.

**5. Técnicas Especiales de Pruebas**

* **Pruebas de Regresión**: Aplicadas cada vez que se realizan modificaciones al modelo predictivo o a la interfaz web, para asegurar que las funcionalidades existentes no se vean afectadas negativamente.
* **Pruebas de Carga y Rendimiento**: Realizadas sobre la interfaz Shiny y la base de datos PostgreSQL para evaluar cómo se comportan bajo condiciones de alta carga, simulando múltiples usuarios concurrentes o grandes volúmenes de datos.
* **Pruebas Exploratorias**: Permiten a los testers explorar libremente las funcionalidades de los dashboards y el sistema de predicciones para identificar defectos que podrían no estar cubiertos por casos de prueba formales.
* **Pruebas de Integración**: Se verificará que la conexión entre R y PostgreSQL, así como la integración entre los dashboards y la base de datos, funcionen correctamente y que los datos se reflejen en tiempo real en las visualizaciones.

**6. Metodologías y Marcos de Trabajo**

* **CRISP-DM (Cross Industry Standard Process for Data Mining)**: Se sigue esta metodología para estructurar el flujo de trabajo de pruebas en el análisis predictivo, asegurando que cada etapa del proceso de minería de datos esté debidamente probada.
* **Pruebas Basadas en Riesgos**: Focalizando los esfuerzos de prueba en las áreas de mayor impacto y riesgo, como la precisión del modelo predictivo y la visualización de datos en tiempo real en los dashboards, para asegurar la fiabilidad de los resultados.

### Personal

**1. Líder de Pruebas (1)**

* **Responsabilidades**:
  + Planificar y coordinar las actividades de prueba.
  + Asegurar que todas las pruebas se realicen de acuerdo con el cronograma y los estándares de calidad.
  + Revisar y aprobar los casos de prueba, así como supervisar los resultados de las pruebas.
  + Realizar informes de avance y presentar los resultados de pruebas a la dirección del proyecto.
* **Perfil**: Experiencia en gestión de pruebas y en la coordinación de equipos de testing, idealmente con conocimientos en análisis predictivo y pruebas de sistemas de datos.

**2. Analistas de Pruebas (4)**

* **Responsabilidades**:
  + Diseñar, ejecutar y documentar casos de prueba basados en los requisitos del usuario y del sistema.
  + Realizar pruebas funcionales y de regresión para verificar la exactitud y el rendimiento del modelo predictivo y la interfaz web.
  + Identificar y reportar defectos en las funcionalidades evaluadas.
* **Perfil**: Conocimiento en testing funcional, experiencia en pruebas de software con interfaces de usuario y familiaridad con herramientas como Selenium y TestRail.

### Entrenamiento

Para asegurar la efectividad de las pruebas en el sistema de análisis predictivo, se han identificado las siguientes necesidades de entrenamiento para el equipo de pruebas:

1. **Capacitación en el Sistema de Análisis Predictivo**:
   * Entrenamiento sobre el flujo y funcionalidades principales del sistema, orientado a comprender la lógica de los modelos predictivos, el procesamiento de datos y la interacción de usuarios finales con el sistema.
   * Familiarización con el uso de dashboards de análisis y las herramientas de visualización integradas (desarrollados con shiny y shinydashboard en R).
2. **Entrenamiento en Herramientas de Prueba**:
   * **PostgreSQL**: Capacitación básica para acceder y ejecutar consultas en la base de datos de pruebas. Esto incluye entender la estructura de la base de datos y realizar pruebas de integridad de datos.
   * **R y RStudio**: Entrenamiento en el uso de R y RStudio para la ejecución de scripts y pruebas de integración con el sistema. Incluye la capacitación en librerías específicas (como dplyr, caret, y ggplot2) usadas en el proyecto.
   * **GitHub**: Formación en el uso de GitHub para gestión de versiones y colaboración en la documentación de resultados de pruebas y generación de informes de errores.
   * **Herramientas de Automatización de Pruebas**: Si se emplean scripts automatizados, el personal de pruebas debe estar capacitado en la herramienta de automatización seleccionada para el proyecto.
3. **Técnicas de Pruebas de Rendimiento y Carga**:
   * Capacitación en herramientas de prueba de carga y rendimiento que se utilizarán para evaluar la escalabilidad del sistema. Se incluirán prácticas en la configuración de entornos para pruebas de estrés y análisis de resultados.

Estos entrenamientos buscan asegurar que el equipo de pruebas tenga las habilidades técnicas y conocimientos específicos para ejecutar pruebas efectivas y eficientes en el sistema y sus componentes.

### Planificación y Organización

**Procedimientos para las Pruebas**

Para la ejecución del plan de pruebas del sistema de análisis predictivo, se implementará la metodología de pruebas V-Model debido a su enfoque en la verificación y validación en cada etapa del desarrollo, asegurando así que cada fase de pruebas esté alineada con los requisitos establecidos desde el principio. Los procedimientos y tipos de pruebas incluyen:

1. **Pruebas Unitarias**:
   * **Objetivo**: Validar que cada módulo o función individual funcione según lo esperado, especialmente en las etapas iniciales del modelado predictivo.
   * **Procedimiento**: Los desarrolladores ejecutarán pruebas automatizadas en cada módulo de código mediante R y sus librerías de prueba.
   * **Criterios de Aceptación**: 100% de cobertura en funciones críticas y al menos 90% de casos exitosos.
2. **Pruebas de Integración**:
   * **Objetivo**: Verificar la correcta interacción entre módulos, incluyendo la conexión entre R y la base de datos en PostgreSQL.
   * **Procedimiento**: Simulación de flujos de datos y pruebas de interacción entre componentes como RPostgres para la conexión a la base de datos, shiny para la visualización de datos, y los dashboards en *shinydashboard*.
   * **Criterios de Aceptación**: Correcta transferencia y procesamiento de datos sin errores en al menos 95% de los casos.
3. **Pruebas de Sistema**:
   * **Objetivo**: Validar el sistema completo en un entorno de pruebas, asegurando que todas las funcionalidades del análisis predictivo funcionan según los requisitos.
   * **Procedimiento**: Ejecución de pruebas funcionales, pruebas de rendimiento y carga para asegurar la estabilidad del sistema bajo condiciones normales y extremas.
   * **Criterios de Aceptación**: Cumplimiento del 100% de los requisitos funcionales y una respuesta aceptable bajo carga.
4. **Pruebas de Regresión**:
   * **Objetivo**: Garantizar que nuevas modificaciones o correcciones no introduzcan defectos en las funcionalidades ya existentes.
   * **Procedimiento**: Pruebas automatizadas para verificar las funciones afectadas por cambios recientes y otras funcionalidades principales.
   * **Criterios de Aceptación**: Al menos 95% de los casos de regresión deben completarse sin errores.
5. **Pruebas de Aceptación del Usuario**:
   * **Objetivo**: Asegurar que el sistema cumpla con las expectativas y requisitos del usuario final.
   * **Procedimiento**: Presentación del sistema a usuarios clave para validar su usabilidad, precisión del análisis predictivo, y accesibilidad de los dashboards.
   * **Criterios de Aceptación**: Aprobación de usuarios clave en todos los módulos críticos del sistema.
6. **Pruebas de Documentación y Reportes**:
   * **Objetivo**: Confirmar que toda la documentación asociada (casos de prueba, logs de errores, reportes de incidencias) esté completa y organizada.
   * **Procedimiento**: Revisión de cada documento generado en el proceso de pruebas para asegurar su exactitud y facilidad de interpretación.
   * **Criterios de Aceptación**: Disponibilidad de todos los documentos y reportes requeridos en un repositorio de fácil acceso, como GitHub, con actualizaciones reflejadas en tiempo real.

Estos procedimientos buscan cubrir todos los aspectos críticos del sistema y sus funcionalidades, asegurando una ejecución completa y alineada con los objetivos y requisitos de calidad establecidos para el proyecto de análisis predictivo.

### Matriz de Responsabilidades



### V-Model Pruebas

Explicación del Modelo en V ajustado para el Proyecto de Análisis Predictivo en Skip

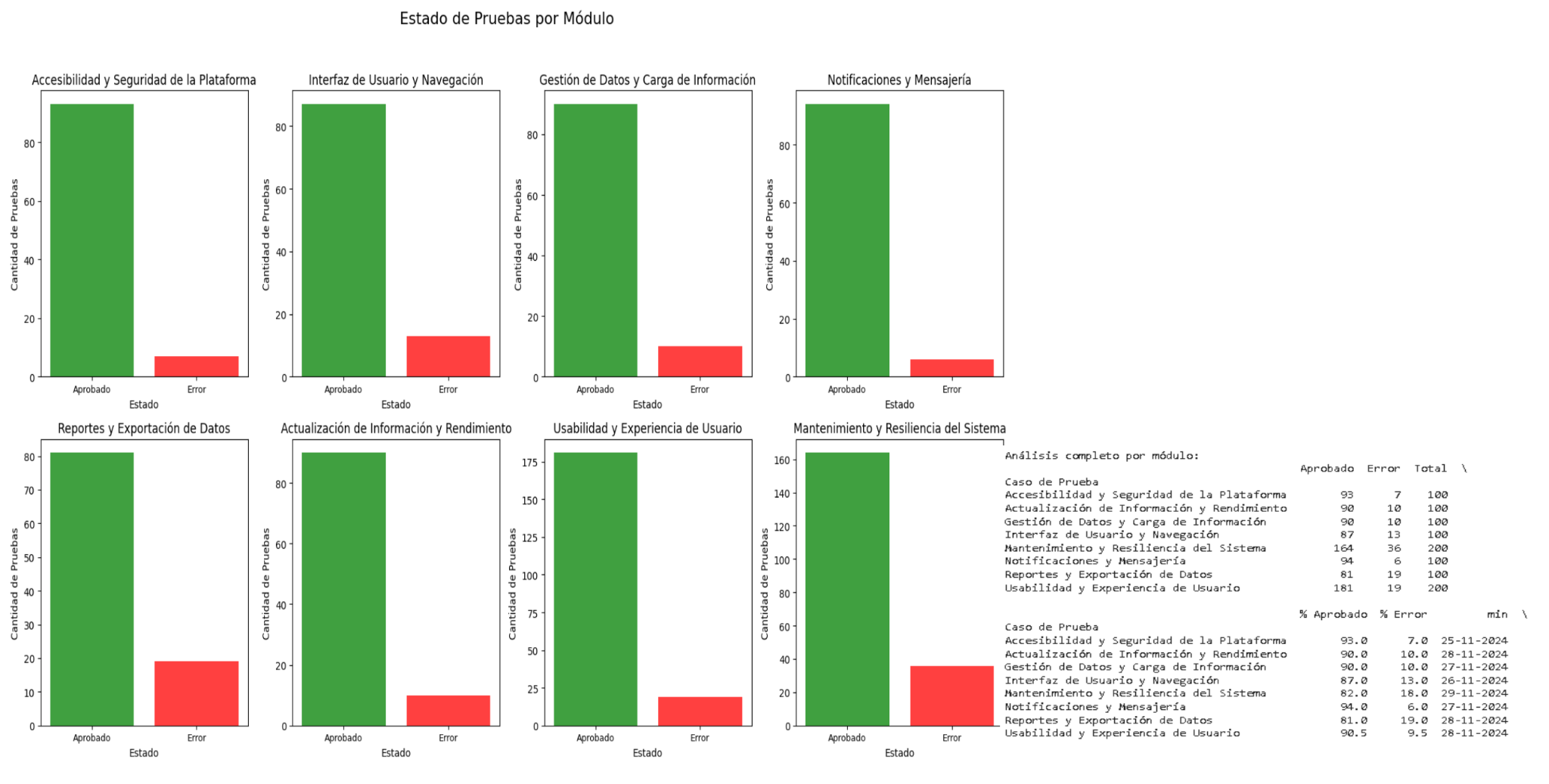
1. **Fase Inicio del Proyecto**
   * **Pruebas de Accesibilidad y Seguridad de la Plataforma**: En esta etapa se verifica que la plataforma cumpla con los requisitos de accesibilidad para todos los usuarios y con los estándares de seguridad para proteger los datos y el sistema de posibles vulnerabilidades.
2. **Fase Comprensión de los Datos**
   * **Pruebas de Interfaz de Usuario y Navegación**: Asegurar que la interfaz sea intuitiva y fácil de navegar para los usuarios encargados de analizar y comprender los datos. Se evalúa la disposición de elementos y la fluidez de la experiencia de usuario.
3. **Fase Análisis de los Datos**
   * **Pruebas de Gestión de Datos y Carga de Información**: Validación de la integridad de los datos y de que la plataforma pueda gestionar y cargar grandes volúmenes de información sin fallos ni pérdida de rendimiento.
4. **Fase Preparación de los Datos**
   * **Pruebas de Notificaciones y Mensajería**: Asegurar que las notificaciones y los sistemas de mensajería funcionen correctamente, proporcionando alertas en tiempo real o mensajes relevantes a los usuarios, en relación a la preparación de los datos y otros eventos importantes.
5. **Fase Modelado**
   * **Pruebas de Reportes y Exportación de Datos**: Validar que el modelo predictivo pueda generar reportes precisos y que los datos puedan ser exportados correctamente en formatos específicos para el análisis y la interpretación.
6. **Fase Evaluación**
   * **Pruebas de Actualización de Información y Rendimiento**: Asegurar que el sistema pueda actualizar los datos y los resultados del modelo de manera eficiente sin afectar el rendimiento general de la plataforma.
7. **Fase Despliegue**
   * **Pruebas de Usabilidad y Experiencia de Usuario**: Evaluación de la experiencia de usuario en el entorno de producción, asegurando que sea intuitiva, fácil de usar y satisfactoria, facilitando la adopción del sistema por los usuarios finales.
8. **Fase Monitoreo y Ajustes Finales**
   * **Resiliencia del Sistema**: En esta última fase se realiza la validación continua y se verifica que el sistema sea resiliente y pueda recuperarse ante posibles fallos o sobrecargas.

### Cronograma

Gráfico

Descripción generada automáticamente

### Estado de Pruebas por Módulo



### Modelo V Prueba Skip

Diagrama

Descripción generada automáticamente

### Caso de Pruebas

Calendario

Descripción generada automáticamente

### Pruebas

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Pruebas Prototipo

Interfaz de usuario gráfica

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

### 

### Premisas

Las siguientes premisas guiarán las tareas de pruebas de software para el sistema de análisis predictivo:

1. **Limitaciones de Tiempo**:
   * Las pruebas se ejecutarán dentro de un plazo de dos meses para cumplir con el cronograma del proyecto, lo que implica la priorización de pruebas críticas y la automatización de casos recurrentes para maximizar la eficiencia.
2. **Disponibilidad de Recursos**:
   * Se cuenta con un equipo reducido de testers, compuesto por un líder de pruebas y dos analistas, por lo que se priorizarán las pruebas automatizadas y se realizarán pruebas manuales solo para los módulos de mayor criticidad.
   * El entorno de pruebas estará disponible de forma continua, aunque la capacidad de la infraestructura será limitada durante los picos de prueba.
3. **Metodología de Pruebas**:
   * Se implementará la metodología V-Model, permitiendo la integración de la verificación y validación en cada fase del ciclo de pruebas, para asegurar la alineación con los requisitos y detectar errores tempranamente.
4. **Herramientas de Pruebas**:
   * Se utilizarán herramientas como testthat para pruebas unitarias en R, y GitHub para la gestión de versiones y documentación de errores. Además, shiny será clave para la ejecución y validación de pruebas de interfaz en los dashboards.
5. **Alcance de Pruebas**:
   * Se priorizarán las pruebas funcionales y de integración de los módulos críticos, dejando en segundo plano pruebas de componentes menos críticos debido a las limitaciones de tiempo y recursos.
6. **Cobertura de Pruebas de Regresión**:
   * Dado el tiempo limitado, solo los módulos más afectados por cambios serán cubiertos por pruebas de regresión completas, mientras que otros componentes menos críticos tendrán una cobertura de regresión parcial.
7. **Accesibilidad de Bases de Datos**:
   * La base de datos PostgreSQL se encontrará en un entorno controlado para pruebas, con acceso solo para los testers autorizados, garantizando la seguridad y consistencia de los datos durante las pruebas.

Estas premisas aseguran un balance entre el cumplimiento de los objetivos de calidad y la optimización de recursos dentro de las limitaciones existentes del proyecto.

### Dependencias y Riesgos

Los riesgos asociados con el proceso de pruebas de software en el proyecto de análisis predictivo son:

1. **Dependencias con Desarrollos**
   * **Descripción:** Retrasos en el desarrollo de módulos clave pueden afectar el inicio y continuidad de las pruebas.
   * **Probabilidad**: Media
   * **Impacto**: Alto
   * **Mitigación:** Asegurar una comunicación continua con el equipo de desarrollo y priorizar módulos críticos en entregas parciales.
   * **Contingencia:** Reasignar recursos a pruebas de otros módulos o a tareas de preparación mientras se completa el desarrollo.
2. **Dependencias con otros Proyectos**
   * **Descripción:** Otros proyectos en la misma infraestructura de pruebas pueden limitar el acceso a entornos de prueba o bases de datos compartidas.
   * **Probabilidad:** Baja
   * **Impacto:** Medio
   * **Mitigación:** Coordinar el uso de recursos y reservar tiempos específicos para evitar conflictos.
   * **Contingencia:** Planificar pruebas offline y gestionar datos de manera local si los entornos no están disponibles.
3. **Disponibilidad de Recursos**
   * **Descripción:** Limitaciones en el número de testers y especialistas en automatización pueden ralentizar las pruebas.
   * **Probabilidad:** Alta
   * **Impacto**: Alto
   * **Mitigación**: Optimizar el uso de recursos, priorizar pruebas críticas, y automatizar pruebas recurrentes.
   * **Contingencia:** Solicitar apoyo temporal de otros equipos o limitar el alcance de pruebas en módulos no críticos.
4. **Restricciones de Tiempo**
   * **Descripción**: El cronograma ajustado puede no permitir realizar pruebas exhaustivas.
   * **Probabilidad:** Media
   * **Impacto:** Alto
   * **Mitigación:** Priorizar casos de prueba de alto impacto y automatizar los que se repiten.
   * **Contingencia:** Posponer algunas pruebas de menor importancia o realizar pruebas post-producción en componentes menos críticos.
5. **Premisas que resulten no ser ciertas**
   * **Descripción:** Si alguna de las premisas, como la disponibilidad del entorno de pruebas o el cronograma de desarrollo, no se cumple, el plan de pruebas podría verse afectado.
   * **Probabilidad**: Media
   * **Impacto**: Medio
   * **Mitigación**: Revisar y validar las premisas regularmente con los stakeholders.
   * **Contingencia:** Reevaluar el plan de pruebas y hacer ajustes en el alcance o en los plazos de pruebas.

Cada riesgo tiene un plan de mitigación para minimizar su ocurrencia y un plan de contingencia para actuar si el riesgo se materializa, asegurando así el éxito del proceso de pruebas dentro de las limitaciones del proyecto.

### Referencias

A continuación se listan los documentos de apoyo que pueden citarse para ampliar el contenido del plan de pruebas del proyecto de análisis predictivo:

1. **Plan de Proyecto**: Detalla el alcance, objetivos, y cronograma general del proyecto, proporcionando el contexto para las pruebas de software.
2. **Especificaciones de Requerimientos**: Documento que define los requisitos funcionales y no funcionales del sistema, fundamental para desarrollar casos de prueba alineados con las necesidades del usuario.
3. **Diseño General**: Proporciona una visión de alto nivel de la arquitectura del sistema, incluyendo la estructura de los módulos y su interacción, lo cual es útil para entender el flujo de trabajo y los puntos críticos a probar.
4. **Diseño Detallado**: Describe las especificaciones técnicas y la lógica de cada módulo, permitiendo la creación de pruebas más precisas y orientadas a los componentes.
5. **Procedimientos y Estándares de Desarrollo**: Normas establecidas para el desarrollo de software, que aseguran una codificación consistente y de calidad, facilitando el proceso de pruebas.
6. **Procedimientos y Estándares de Pruebas**: Documentación que establece las directrices para la realización de pruebas de software, incluyendo tipos de pruebas, criterios de éxito, y métricas a evaluar.
7. **Metodologías Corporativas**: Directrices corporativas para la gestión y ejecución de proyectos, que pueden incluir metodologías de desarrollo ágil, estándares de calidad y políticas de seguridad.
8. **Procedimientos de Gestión de Incidentes**: Proceso para la identificación, documentación y resolución de incidentes encontrados durante el proceso de pruebas.
9. **Estándares de Documentación**: Reglas corporativas para la creación y mantenimiento de documentos técnicos y de pruebas, asegurando claridad y coherencia en la documentación del proyecto.
10. **Manual del Usuario**: Documento que describe la funcionalidad y usabilidad de la aplicación desde la perspectiva del usuario, sirviendo de referencia para validar la experiencia del usuario final en las pruebas.

### Glosario

A continuación se presentan algunas definiciones clave utilizadas en la documentación de pruebas de software y en el área de pruebas en general:

1. **Pruebas de Software**: Proceso de evaluar y verificar que el software funcione correctamente y cumpla con los requisitos especificados. Incluye la ejecución de casos de prueba y la identificación de defectos.
2. **Caso de Prueba**: Un conjunto de condiciones o variables bajo las cuales un evaluador determinará si un sistema o una aplicación cumple con los requisitos o no. Generalmente, incluye los pasos a seguir, los datos de entrada, las condiciones de prueba y los resultados esperados.
3. **Plan de Pruebas**: Documento que describe el alcance, enfoque, recursos y cronograma de las actividades de pruebas, incluyendo los objetivos, criterios de aceptación y los elementos a probar.
4. **Pruebas Funcionales**: Tipo de pruebas que valida que el software realiza todas las funciones especificadas en los requisitos funcionales. Se centra en las acciones realizadas por el usuario.
5. **Pruebas No Funcionales**: Pruebas que evalúan aspectos de rendimiento, usabilidad, seguridad, compatibilidad, entre otros, asegurando que el software cumpla con los requisitos no funcionales.
6. **Pruebas de Regresión**: Pruebas realizadas para verificar que las modificaciones en el software no introducen nuevos defectos en las funcionalidades existentes.
7. **Pruebas de Integración**: Proceso de probar módulos o componentes combinados para garantizar que funcionan correctamente en conjunto y que las interfaces entre ellos son adecuadas.
8. **Pruebas de Rendimiento**: Pruebas que evalúan cómo el sistema funciona bajo una carga específica de usuarios o transacciones, midiendo la velocidad, escalabilidad y estabilidad del sistema.
9. **Pruebas de Aceptación**: Pruebas realizadas para determinar si el sistema cumple con los criterios de aceptación y si es apto para su lanzamiento. Pueden ser realizadas por los usuarios finales o los clientes.
10. **Defecto (Bug)**: Un error o fallo en el software que provoca que funcione incorrectamente o de manera no intencionada. Puede referirse a una discrepancia entre el resultado esperado y el obtenido.
11. **Criterios de Finalización**: Condiciones que deben cumplirse para dar por completado el proceso de pruebas, como el porcentaje de pruebas exitosas o la cobertura de código alcanzada.
12. **Entorno de Pruebas**: Configuración de hardware y software en la que se ejecutan las pruebas de software. Incluye sistemas operativos, bases de datos, redes y otros elementos técnicos necesarios para las pruebas.
13. **Automatización de Pruebas**: Uso de herramientas y scripts para ejecutar pruebas automáticamente, reduciendo el tiempo y esfuerzo en pruebas repetitivas y mejorando la eficiencia.
14. **Criterios de Suspensión y Reanudación**: Condiciones que indican cuándo deben suspenderse las pruebas (por ejemplo, alta cantidad de defectos críticos) y cuándo deben reanudarse una vez solucionados los problemas.
15. **Log de Errores**: Registro detallado de los errores o defectos encontrados durante el proceso de pruebas, incluyendo su descripción, prioridad, y estado de resolución.
16. **Evidencia de Pruebas**: Documentación o capturas de pantalla que demuestran el resultado de las pruebas realizadas, útiles para validar el cumplimiento de los requisitos.
17. **Pruebas de Caja Negra**: Pruebas en las que el evaluador no tiene conocimiento de la estructura interna del código. Se enfoca en la funcionalidad del software y en los resultados que produce.
18. **Pruebas de Caja Blanca**: Pruebas en las que el evaluador tiene conocimiento del código y estructura interna del software, enfocándose en los caminos, bucles, y estructuras internas.
19. **Cobertura de Código**: Métrica que indica el porcentaje de líneas de código que han sido ejecutadas durante las pruebas, utilizado para medir la exhaustividad de las pruebas.
20. **Ciclo de Vida de las Pruebas**: Proceso completo de las pruebas de software, desde la planificación y diseño hasta la ejecución y cierre. Incluye fases como diseño de casos de prueba, ejecución de pruebas y reporte de resultados.