Nicolas Marin, José Oporto

Plan de pruebas

Software HappyLyfe

**Tabla de Contenido**

* 1. **Alcance de las pruebas**
* 1.1.1 Elementos de las pruebas
* 1.1.2 Pruebas funcionales
* 1.2.3 Pruebas no funcionales
* 1.1.4 Riesgos

1. **Descripción**

* 2.1 Descripción y contextualización de plan de prueba

1. **Módulos del software a probar**

* 3.1 Pruebas

1. **objetivo del plan de prueba**
2. **técnicas y estrategias de pruebas a utilizar**
3. Tipos de prueba a utilizar
4. **Detalle de Ejecución de las pruebas**
5. **Tipos de Prueba para realizar**
6. **Técnicas y estrategias de pruebas a utilizar**
7. **Roles y responsabilidades**
8. **Entorno y configuración de ambiente de pruebas**
9. **Calendarización de Pruebas**

**13 riesgos del proceso de Pruebas**

1. **Artefactos y Criterios de Aceptación**

1.1 Alcances de las pruebas

1.1.1 Elementos de las pruebas

**Interfaz de usuario**: Se ha verificado que la interfaz de usuario es intuitiva, funcional y accesible en una variedad de plataformas ya sea móvil o web

**Prueba:** La interfaz se ve bien y es intuitiva y se ve bien en móvil y página web

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Base de datos** : se prueba que los datos ingresados por el usuario coincidan con los datos almacenados en el sistema, incluidos registros de usuarios, ejercicio, métricas de salud y fotografías de progreso.

**Prueba**: El usuario al ingresar a la página se guardan sus datos después de llenar el formulario

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**Características principales**: Se están probando todas las funciones como recomendaciones personalizadas, registro de indicadores de salud, interacción con profesionales de la salud, gestión de fotografías y pagos.

**Servicios de terceros**: Se ha verificado la integración con servicios externos como sistemas de pago y almacenamiento en la nube.

**Seguridad**: pruebas para garantizar que la información personal y de salud esté adecuadamente protegida.

**Notificaciones** : Validación de funciones de recordatorios y notificaciones inteligentes

**1.1.2** Pruebas funcionales

**Recomendaciones de ejercicios personalizadas**: Se probará que el sistema pueda generar recomendaciones adaptadas a las características del usuario (edad, peso, nivel de exigencia).

**Registro y monitoreo de progreso**: Se validará que los usuarios puedan registrar y visualizar sus métricas de salud de manera precisa.

**El usuario puede ver sus métricas**

**Prueba**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**El usuario puede ingresar los sus datos y gráficos, pero no tiene variaciones de caracteres o input vacío**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Acceso a Profesionales de la salud**: Se verificará que los usuarios puedan comunicarse efectivamente con profesionales de la salud.

**Prueba:** El usuario todavía no puede comunicarse con algún profesional pero si puede ver visualmente

Patrón de fondo

Descripción generada automáticamente con confianza baja

**Gestión de fotos de progreso**: Se comprobará que los usuarios puedan tomar, almacenar y visualizar sus fotos a lo largo del tiempo.

**Prueba**: el usuario puede subir sus fotos, pero el error fue que no puede modificarlas ni eliminarlas

Imagen que contiene ropa

Descripción generada automáticamente

**Pagos y suscripciones**: Se evaluará que el sistema gestione correctamente los registros de nuevos usuarios y el procesamiento de pagos.

**El sistema todavía no puede realizar pagos**

**Notificaciones inteligentes**: Se validará que el sistema envíe notificaciones relevantes basadas en el comportamiento y progreso del usuario.

**El sistema todavía no puede enviar notificaciones**

**1.1.3** Riesgos

Dentro de nuestro software es necesario tener en consideración los siguientes riesgos

**Cambios en los requisitos**: Modificaciones en las especificaciones funcionales durante el desarrollo pueden llevar a errores en las pruebas y retrasar la entrega del producto

**Integración con servicios de terceros**: Si los servicios de terceros como servicios de pago o almacenamiento de datos fallan podrían retrasar el proceso de pruebas.

**Riesgo de rendimiento**: Si no se realizan pruebas de rendimiento adecuadas, el sistema podría tener fallos cuando muchos usuarios accedan simultáneamente.

**Falla en la seguridad de los datos personales de usuarios**: Si las pruebas de seguridad no detectan vulnerabilidades a tiempo, los datos sensibles de los usuarios podrían estar en riesgo.

2 descripción

2.1 Descripción y contextualización de plan de prueba

Este plan de pruebas se implementará para garantizar que el software “HappyLife” funcione según lo especificado, cumpliendo tanto con los requisitos funcionales como no funcionales dados por el cliente. Además de obviamente llevar a cabo Las pruebas que cubrirán desde la experiencia del usuario con el sistema hasta la correcta implementación de servicios externos junto con pruebas de rendimiento, seguridad, almacenamiento, etc.

3 módulos del software a probar

- Módulo de Recomendaciones de Ejercicio Personalizadas

-Módulo de Registro y Monitoreo de Salud

- Módulo de Acceso a Profesionales de la Salud

- Módulo de Gestión de Fotos de Progreso

-Módulo de Gestión de Pagos y Suscripciones

-Módulo de Notificaciones Inteligentes

-Integración con servicios de terceros (pagos, almacenamiento de fotos)

* 1. Casos de Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | Se hará registro de usuario e inicio de sesión para verificar que las validaciones estén correctas | alta | El programa verifica que las validaciones requeridas previamente validen correctamente al usuario que quiera iniciar sesión o registrarse |
| 2 | Se verificará que los datos personales ingresados por los clientes cumplen con los parámetros determinados | media | El programa verifica que los parámetros principales para el correcto funcionamiento del sistema como por ejemplo edad, peso, estatura están dentro de los rangos predefinidos en el codigo |
| 3 | se pagará con transbank | alta | Nula |
| 4 | Se seleccionará un nivel de dificultad de una rutina de ejercicio | alta | El programa le mostrara al usuario las rutinas de ejercicios específicos para esa dificultad |
| 5 | se mandara una solicitud de mensaje hacia el profesional de la salud deseado (kinesiólogo o nutricionista) | alta | El programa notificara al profesional que hay una solicitud de mensaje y este al contestar le deberá mandar una notificación al usuario de que le han respondido |
| 6 | se cierra sesión | media | Se cierra la sesión y no queda ningún indicio de uso anterior |
| 7 | Seleccionara la opción de progreso | alta | El programa debe mostrar en la interfaz el progreso registrado e ingresado por el usuario |
| 8 | Se hará una prueba de carga haciendo que varios usuarios se registren e inician sesión al mismo tiempo | alta | El programa debe estar funcional sin importar la cantidad de usuarios  (dependiendo de la capacidad de los servidores) |
| 9 | Cambio de secciones dentro de la pagina | baja | El tiempo de navegación entre secciones y opciones del sistema debe ser menor a 0.8 milisegundos |
| 10 | Revisar el perfil | media | El programa mostrara los datos ingresados por el usuario al registrarse |
| 11 | Modificar el perfil | alta | El cambio de algunos datos del perfil de usuario debe ser instantáneo |
| 12 | Subir foto de progres | alta | El programa guardara y mostrara al usuario las fotos del progreso que la suba |
| 13 | Se comprará la suscripción premium | alta | El programa mostrara al usuario las nuevas opciones habilitadas solamente a los usuarios con suscripción premium |
| 14 | Se dará permisos a las notificaciones del sistema | alta | El programa enviara una notificación al usuario en los casos de que se requiera enviar una |

1. **Objetivo plan de prueba**

* Los objetivos del plan de prueba son garantizar que el sistema cumpla con los requisitos funcionales especificados, proporcione un rendimiento estable bajo cargas variables, respete los estándares de seguridad y privacidad, sea accesible y fácil de usar y esté equipado con una variedad de tecnologías.

Puede integrarse y adaptarse fácilmente a su entorno.

Esto garantiza la calidad y el éxito durante todo el ciclo de vida de la aplicación.

5 **técnicas y estrategias de pruebas a utilizar**

Después de finalizar nuestro proyecto y que esté listo para que se le puedan realizar las pruebas correspondientes y así garantizar el correcto funcionamiento y calidad de nuestro sistema planteamos una estrategia de pruebas estructuradas en las cuales se llevaran a cabo las pruebas en varias fases, dividiéndose en frontend y backend

Como por ejemplo las Pruebas Funcionales en el Frontend se probarán todas las funcionalidades que interactúan directamente con el usuario. El objetivo es verificar que cada función del sistema cumple con los requisitos funcionales especificados. Estas pruebas se llevarán a cabo en una variedad de dispositivos y navegadores para asegurar una correcta compatibilidad

Luego del frontend nos enfocaremos en todo lo que es backend e integración de servicios los cuales interactúan con el backend verificando que las solicitudes y respuestas entre el sistema y la base de datos, así como con sistemas externos, funcionen correctamente.

También realizaremos pruebas de carga y de rendimiento para garantizar que el sistema pueda soportar un alto número de usuarios simultáneos sin comprometer el rendimiento.

* **5.1 Tipos de prueba a utilizar**
* 1. **Pruebas Funcionales**
* Estas pruebas se enfocarán en verificar que las funcionalidades del sistema cumplan con los requerimientos especificados. Se realizarán pruebas funcionales utilizando la técnica de caja negra, que evalúa la funcionalidad de la aplicación sin tener en cuenta su código interno.
* Objetivo: Validar que cada funcionalidad, desde las recomendaciones personalizadas hasta la gestión de fotos y pagos, funcione correctamente según los requisitos.
* Técnica: Ingresar datos válidos e inválidos en las distintas pantallas de la aplicación, verificando que el sistema reaccione de forma apropiada en cada caso.
* Criterios de validación:
* Todos los flujos de trabajo deben funcionar correctamente.
* Los mensajes de error deben mostrarse cuando se utilicen datos incorrectos.
* La navegación, interacción con elementos y transacciones deben funcionar como se espera.
* 2. **Pruebas de Usabilidad**
* Estas pruebas se centrarán en verificar que la interfaz sea intuitiva, fácil de usar y accesible para los usuarios.
* Objetivo: Garantizar que los usuarios puedan navegar sin problemas por la aplicación y realizar las acciones necesarias sin confusión.
* Técnica: Probar la navegación y la interfaz de usuario en dispositivos móviles con diferentes tamaños de pantalla y resoluciones. Involucrar usuarios reales para realizar pruebas de experiencia.
* Criterios de validación:
* La navegación debe ser intuitiva y clara.
* Los elementos de la interfaz (botones, menús, formularios) deben ser fácilmente accesibles y comprensibles.
* 3. **Pruebas de Seguridad**
* Dado que la aplicación maneja datos personales sensibles, estas pruebas verificarán que el sistema proteja adecuadamente la información del usuario.
* Objetivo: Asegurar que la aplicación siga las mejores prácticas de seguridad, garantizando que los datos de los usuarios no sean vulnerables.
* Técnica: Pruebas de penetración para identificar vulnerabilidades, pruebas de autenticación y autorización, encriptación de datos en tránsito y en reposo.
* Criterios de validación:
* Todos los accesos deben estar correctamente restringidos según los roles de usuario.
* La información personal debe estar cifrada y protegida.
* 4. **Pruebas de Rendimiento**
* Estas pruebas se centrarán en verificar que el sistema mantenga un rendimiento aceptable bajo diferentes condiciones de carga.
* Objetivo: Garantizar que la aplicación funcione de manera eficiente incluso bajo cargas pesadas, sin afectar negativamente la experiencia del usuario.
* Técnica: Realizar simulaciones con grandes volúmenes de usuarios concurrentes y medir tiempos de respuesta y uso de recursos.
* Criterios de validación:
* El sistema debe responder en menos de 2 segundos en el 95% de los casos.
* Debe soportar al menos 1000 usuarios simultáneos sin caídas.
* 5. **Pruebas de Integración**
* Las pruebas de integración evaluarán la correcta interacción entre los diferentes módulos y servicios de la aplicación.
* Objetivo: Asegurar que los diferentes componentes (backend, frontend, bases de datos, APIs) funcionen bien juntos y que los datos fluyan sin problemas entre ellos.
* Técnica: Probar la interacción entre el sistema de recomendaciones de ejercicios, la base de datos de usuarios y el módulo de pago, entre otros.
* Criterios de validación:
* Los datos deben sincronizarse correctamente entre todos los módulos.
* Los webs servicies y Apis deben devolver respuestas adecuadas en todo momento.
* 6. **Pruebas de Compatibilidad**
* Estas pruebas verificarán que la aplicación funcione correctamente en diferentes dispositivos y sistemas operativos.
* Objetivo: Garantizar que la aplicación sea compatible con una amplia gama de dispositivos y navegadores.
* Técnica: Realizar pruebas en diferentes dispositivos móviles con diversos sistemas operativos (iOS, Android) y resoluciones de pantalla.
* Criterios de validación:
* La aplicación debe comportarse de manera consistente en todos los dispositivos soportados.
* No debe haber errores de visualización o mal funcionamiento en ninguno de los entornos probados.
* 7. **Pruebas de Regresión**
* Cada vez que se realicen cambios o se introduzcan nuevas funcionalidades, se realizarán pruebas de regresión para asegurar que los cambios no afecten funcionalidades previamente verificadas.
* Objetivo: Asegurar que nuevas características no rompan funcionalidades anteriores.
* Técnica: Re ejecutar los casos de prueba existentes para verificar que el sistema siga funcionando como se esperaba.
* Criterios de validación:
* Ninguna funcionalidad previamente probada debe fallar tras los cambios.
* 8. **Pruebas de Mantenibilidad**
* Estas pruebas verificarán que el sistema sea fácil de mantener y que se puedan aplicar actualizaciones o parches sin interrumpir el servicio.
* Objetivo: Asegurar que el sistema pueda mantenerse y actualizarse fácilmente.
* Técnica: Probar la implementación de actualizaciones y parches sin afectar la operación del sistema.
* Criterios de validación:
* Los parches y actualizaciones deben implementarse sin interrupciones significativas en el servicio.
* Los tiempos de mantenimiento deben ser mínimos.

1. **roles y responsabilidades**

| **Rol** | **Responsables** | **Responsabilidades específicas / Comentarios** |
| --- | --- | --- |
| Jefe Proyecto | Salvador Allende | - Construir la carta Gantt - Definir las metas - Actuar como un puente de comunicación entre equipos - Mantener una visión general - Asignar roles a cada integrante del equipo |
| Testeador | Víctor Jara | - Ejecutar pruebas funcionales y no funcionales - Documentar errores y problemas encontrados - Informar sobre el estado de las pruebas al equipo |
| Analista | José Oporto | - Identificar requisitos de prueba - Crear casos de prueba - Revisar y validar resultados de pruebas - Asegurar que los requisitos estén correctamente cubiertos en las pruebas |
| Desarrollador | Nicolas Marín | |  | | --- | |  |  |  | | --- | | -Implementar correcciones basadas en los resultados de las pruebas - Colaborar con el analista para entender los requisitos de prueba - Realizar pruebas unitarias y de integración | |

1. Entorno y configuración de ambiente de pruebas
   1. **Ambiente de pruebas**

El entorno de pruebas debe garantizar que todas las funcionalidades del software HappyLife sean probadas en un ambiente seguro, estable y que simule las condiciones reales de uso. El ambiente debe incluir tanto configuraciones para pruebas en dispositivos móviles como en navegadores web.

* 1. ***Preparación del ambiente de pruebas***

Las pruebas unitarias y de integración se realizarán en un ambiente de desarrollo, mientras que las pruebas de aceptación y rendimiento se ejecutarán en un ambiente de pruebas que simula el entorno de producción. Este entorno debe replicar las condiciones de uso en cuanto a hardware y software para asegurar resultados precisos.

* 1. ***Diseño del ambiente de pruebas***

***Móvil***

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| Procesador | Snapdragon 429 | Probar diferentes procesadores |
| Memoria RAM | 4 GB (mínimo) |  |
| Espacio en Disco | 64 GB (mínimo) |  |
| Tipo de Dispositivo | Smartphone (Android/iOS) | Probar diferentes modelos y versiones |
| Resolución de Pantalla | Variada (HD, Full HD, 4K) |  |
| Red / Conexión | 4G/5G o Wi-Fi |  |

**Web**

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| Procesador | Intel Core i5 o equivalente |  |
| Memoria RAM | 8 GB (mínimo) |  |
| Espacio en Disco | 500 GB (mínimo) |  |
| Resolución de Pantalla | 1920 x 1080 (Full HD) | Probar en diferentes resoluciones. |
| Red / Conexión | Ethernet o Wi-Fi |  |
| Navegador | Chrome, Firefox, Safari, Edge | Verificar compatibilidad en diferentes navegadores. |

**Red**

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| Topología | Red de estrella |  |
| Medio | Inalámbrico y por cable |  |
| Velocidad | 1 ms (baja latencia) |  |
| Protocolo | TCP/IP |  |
| Conexión a Internet | Banda ancha (fibra óptica) |  |

**Seguridad**

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| Sistema de Respaldo | Unidad de respaldo externa |  |
| Sistema de Restauración | Plan de recuperación ante desastres |  |

**Pruebas (Móvil y Web): Configuraciones software que replican las condiciones de uso esperadas por los usuarios finales.**

**Red: Configuración de red similar a la que se usará en producción para garantizar que el rendimiento y la seguridad sean representativos.**

**Seguridad: Implementación de medidas de seguridad que aseguren la protección de los datos y la integridad del sistema durante las pruebas.**

1. calendarización de Pruebas

**8.1 Fase de Preparación**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Definición de Plan de Pruebas | Redacción del plan de pruebas detallado | Nicolás Marín | |  | | --- | |  |  |  | | --- | | 01/10/2024 | | 05/10/2024 | Pendiente |
| Configuración del Ambiente de Pruebas | Instalación y configuración de hardware y software | José Oporto | 06/10/2024 | |  | | --- | | 10/10/2024 |  |  | | --- | |  | | Pendiente |
| Creación de Casos de Prueba | Elaboración de casos de prueba basados en requisitos | Nicolás Marín | 11/10/2024 | |  | | --- | | 15/10/2024 |  |  | | --- | |  | | Pendiente |

**8.2 Fase de Pruebas Unitarias**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas Unitarias | Pruebas individuales de cada módulo | José Oporto | 16/10/2024 | 20/10/2024 | Pendiente |
| Revisión y Corrección de Errores | Análisis y corrección de fallos encontrados | José Oporto | |  | | --- | | 21/10/2024 |  |  | | --- | |  | | 23/10/2024 | Pendiente |

**8.3 Fase de Pruebas de Integración**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas de Integración | Pruebas de interacción entre módulos | Nicolás Marín | 24/10/2024 | 28/10/2024 | Pendiente |
| Revisión y Corrección de Errores | Análisis y corrección de fallos encontrados | Nicolás Marín | 29/10/2024 | 31/10/2024 | Pendiente |

**8.4 Fase de Pruebas Funcionales y de Usabilidad**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas Funcionales | Validación de funcionalidades del sistema | Nicolás Marín | 01/11/2024 | 07/11/2024 | Pendiente |
| Ejecución de Pruebas de Usabilidad | Evaluación de la interfaz y experiencia del usuario | Nicolás Marín | 08/11/2024 | 14/11/2024 | Pendiente |
| Revisión y Corrección de Errores | Análisis y corrección de fallos encontrados | Nicolás Marín | 15/11/2024 | |  | | --- | | 17/11/2024 |  |  | | --- | |  | | Pendiente |

**8.5 Fase de Pruebas de Rendimiento**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas de Rendimiento | Pruebas bajo diferentes cargas de usuarios | José Oporto | 18/11/2024 | 22/11/2024 | Pendiente |
| Análisis de Resultados | Evaluación del rendimiento y ajuste de configuraciones | José Oporto | 23/11/2024 | |  | | --- | |  |  |  | | --- | | 25/11/2024 | | Pendiente |

**8.6 Fase de Pruebas de Seguridad**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas de Seguridad | Evaluación de vulnerabilidades y protección de datos | Nicolás Marín | |  | | --- | | 26/11/2024 |  |  | | --- | |  | | 30/11/2024 | Pendiente |
| Revisión y Corrección de Errores | Análisis y corrección de fallos encontrado | Nicolás Marín | 01/12/2024 | 03/12/2024 | Pendiente |

**8.7 Fase de Pruebas de Aceptación**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Descripción** | |  | | --- | |  |  |  | | --- | | **Responsable** | | **Fecha de Inicio** | **Fecha de Finalización** | **Estado** |
| Ejecución de Pruebas de Aceptación | Validación final del sistema por usuarios finales | Nicolás Marín | 04/12/2024 | 08/12/2024 | Pendiente |
| Revisión Final y Reporte de Pruebas | Documentación de resultados y cierre del ciclo de pruebas | Nicolás Marín | 09/12/2024 | 11/12/2024 | Pendiente |

1. **riesgos del proceso de Pruebas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Riesgo** | **Descripción** | **Impacto** | |  | | --- | |  |  |  | | --- | | **Mitigación** | |
| Cambios en los Requisitos | Modificaciones durante el desarrollo afectan las pruebas. | Alto | Comunicación constante y actualización de casos de prueba. |
| Integración con Terceros | |  | | --- | | Terceros |  |  | | --- | | Dependencia de servicios externos que pueden fallar. | | Alto | Pruebas de integración continuas y plan de contingencia. |
| Rendimiento del Sistema | Problemas bajo carga alta. | Alto | Pruebas de carga y optimización durante el desarrollo. |
| Seguridad de los Datos | Vulnerabilidades que comprometen datos sensibles. | Alto | Pruebas de seguridad y auditorías regulares. |
| Ambiente de Pruebas Inadecuado | Configuración no fiel al entorno de producción. | Medio | Simulación precisa del entorno de producción |
| Falta de Recursos | Personal o recursos insuficientes retrasan el proceso. | Medio | Asignación adecuada de recursos y capacitación del equipo. |
| Errores en Casos de Prueba | Casos incompletos que no detectan fallos importantes. | Medio | Revisión y validación periódica de casos de prueba. |
| Retrasos en Resultados | Resultados tardíos afectan el cronograma del proyecto. | Bajo | Cronograma claro y comunicación eficiente. |

**10. Artefactos y Criterios de Aceptación**

Durante el proceso de pruebas de la aplicación Happy Life, se generan varios documentos y resultados clave para garantizar la calidad del sistema.

* **Plan de Pruebas**: Un documento que establece qué se va a probar, cómo se realizará y cuál será el enfoque. Se considera aprobado cuando cumple con los requisitos acordados por el equipo de desarrollo y demás involucrados.
* **Casos de Prueba**: Escenarios diseñados para validar funciones específicas de la aplicación. Se aceptan cuando aseguran que se han cubierto todos los requisitos funcionales y no funcionales del sistema.
* **Evidencias de Pruebas**: Resultados documentados que muestran el comportamiento del sistema al ejecutar los casos de prueba. Se consideran válidas cuando reflejan un comportamiento correcto sin errores críticos.
* **Informe de Defectos**: Un registro de los problemas detectados durante las pruebas. Este informe se da por cerrado cuando los defectos importantes han sido corregidos y verificados.
* **Informe de Rendimiento**: Análisis del comportamiento del sistema bajo diferentes condiciones de uso. Es satisfactorio cuando el sistema cumple con los tiempos de respuesta y rendimiento establecidos.
* **Informe Final de Pruebas**: Un resumen de todos los resultados obtenidos durante las fases de prueba. Se aprueba si demuestra que el sistema ha superado los criterios de aceptación fijados por el cliente.