**Plan de Pruebas de Software**

***Mantenimiento al volante BUPESA***

***Fecha: [14/11/2024]***

**Tabla de contenido**

Historial de Versiones 4

Información del Proyecto 4

Aprobaciones 4

Resumen Ejecutivo 5

Alcance de las Pruebas 5

Elementos de Pruebas 5

Pruebas de Regresión 6

Funcionalidades a No Probar 7

Enfoque de Pruebas (Estrategia) 7

Criterios de Aceptación o Rechazo 8

Criterios de Aceptación o Rechazo 8

Criterios de Suspensión 8

Criterios de Reanudación 9

Entregables 9

Recursos 10

Requerimientos de Entornos – Hardware 10

Requerimientos de Entornos – Software 10

Herramientas de Pruebas Requeridas 11

Personal 11

Entrenamiento 12

Planificación y Organización 12

Procedimientos para las Pruebas 12

Dependencias y Riesgos 14

Referencias 15

Glosario 15

**Historial de Versiones**

| **Fecha** | **Versión** | **Autor** | **Descripción** |
| --- | --- | --- | --- |
| 14-11-2024 | 1.0 | Miguel Herrera. | Relleno del documento. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Información del Proyecto**

| Empresa / Organización | RH Software. |
| --- | --- |
| Proyecto | Mantenimiento al volante BUPESA. |
| Fecha de preparación | 14-11-2024 |
| Cliente | BUPESA |
| Gerente / Líder de Proyecto | Gustavo Rojas. |
| Gerente / Líder de Pruebas de Software | Miguel Herrera. |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Fecha** | **Firma** |
| --- | --- | --- | --- |
| Miguel Herrera | Gerente / Líder de Pruebas de Software | 14-11-2024 | MIG |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Resumen Ejecutivo**

Este documento clave en la gestión de calidad de un proyecto de software, está diseñado para estructurar y organizar todas las actividades de prueba que se realizan. Su propósito principal es asegurar que el producto cumpla con los requisitos de calidad y funcionalidad establecidos, y proporcionar una guía clara para el equipo de pruebas en cada fase del proceso de desarrollo.

**Alcance de las Pruebas**

**Elementos de Pruebas**

La funcionalidad del botón editar

La funcionalidad de eliminar bus

La funcionalidad de agregar bus

La funcionalidad de nuevo mantenimiento.

La funcionalidad de agregar usuario

La funcionalidad editar usuario.

La funcionalidad eliminar usuario.

El Login del sistema.

**Pruebas de Regresión**

Listado de las funcionalidades no directamente involucradas en el desarrollo, pero cuyos componentes están siendo afectados y por ende deben probarse para asegurar que continúan funcionando adecuadamente. Al igual que en el punto anterior, se describen desde el punto de vista del usuario.

**Funcionalidades a No Probar**

Funcionalidad de Imprimir: Usa una funcionalidad que es de JavaScript y no es necesario hacerle pruebas a esta funcionalidad ya que es externa.

**Enfoque de Pruebas (Estrategia)**

Se realizarán pruebas funcionales del sistema de mantenimiento BUPESA, como funcionalidades de un botón, validaciones del login o características propia del sistema que se realizó y se realizará un reporte en los casos de pruebas de sus resultados obtenidos.

**Criterios de Aceptación o Rechazo**

**Criterios de Aceptación o Rechazo**

### 1. Cobertura de Pruebas Unitarias

* **100% de cobertura en pruebas unitarias críticas**: Garantizar que todos los módulos y funciones críticas hayan sido probados al nivel de código, cubriendo todas las rutas lógicas relevantes.
* **Cobertura mínima aceptable**: Por lo general, se establece un porcentaje de cobertura (por ejemplo, 80-90%) en pruebas unitarias para todas las funcionalidades no críticas.

### 2. Ejecución de Casos de Prueba

* **100% de casos de prueba ejecutados**: Todos los casos de prueba planificados deben haberse ejecutado para asegurar una cobertura completa del sistema.
* **Porcentaje de casos exitosos**: Al menos un 90-95% de los casos de prueba deben haber tenido éxito, dependiendo de los estándares del proyecto.
* **Cobertura funcional completa**: Todas las funcionalidades y módulos descritos en el plan deben haber sido probados.

### 3. Cobertura de Código

* **Cobertura de código aceptable**: Se debe establecer un porcentaje objetivo (por ejemplo, 85% de cobertura de líneas de código) para asegurar que se ha cubierto adecuadamente la lógica de la aplicación.
* **Cobertura de todos los componentes**: Asegurar que cada componente y módulo ha sido probado en condiciones de operación normales y límites.

### 4. Reducción de Defectos

* **Defectos críticos corregidos**: Todos los defectos críticos y de alta prioridad deben haberse resuelto antes de finalizar las pruebas.
* **Reducción de defectos a un nivel aceptable**: Limitar el número de defectos menores a un umbral aceptable (por ejemplo, menos del 5% de defectos abiertos o de baja prioridad).
* **Validación de corrección**: Asegurar que todos los defectos corregidos hayan sido validados y verificados en pruebas posteriores.

### 5. Documentación y Reporte de Resultados

* **Generación de reportes finales**: Documentación completa de los resultados de las pruebas, informes de defectos y métricas de calidad obtenidas.
* **Entrega de documentación de cierre**: El plan de pruebas debe incluir una conclusión formal y cualquier documentación que certifique que el software ha cumplido con los criterios de calidad establecidos.

**Criterios de Suspensión**

**Existencia de Defectos Bloqueantes**

* Si se encuentra uno o más defectos críticos o bloqueantes que impiden la ejecución de casos de prueba adicionales, las pruebas se detendrán. Estos defectos pueden incluir fallos que afectan la funcionalidad principal, o que provocan caídas o bloqueos en el sistema.

**Inestabilidad del Sistema**

* En caso de que el sistema esté inestable o se presenten fallos recurrentes que no permiten ejecutar las pruebas con fluidez (como tiempos de respuesta inconsistentes, caídas frecuentes, o comportamientos anómalos), se detendrán las pruebas hasta que se garantice la estabilidad del sistema.

**Alto Riesgo de Seguridad**

* Si se identifican vulnerabilidades o riesgos de seguridad graves que puedan comprometer el sistema, los datos o la privacidad del usuario, se suspenderán las pruebas hasta que estos problemas se resuelvan y se pueda asegurar la seguridad en el entorno de pruebas.

**Criterios de Reanudación**

**Corrección de Defectos Bloqueantes**

Los defectos críticos o bloqueantes que llevaron a la suspensión de las pruebas han sido corregidos, probados y confirmados en un entorno controlado. Esto incluye verificar que los fallos críticos que impedían la ejecución de pruebas adicionales han sido resueltos satisfactoriamente.

**Estabilidad del Sistema Restablecida**

Se confirma que el sistema es estable y que las caídas, errores de rendimiento u otros comportamientos anómalos que llevaron a la suspensión ya no se presentan. Esto puede implicar pruebas iniciales de validación de estabilidad antes de retomar la ejecución completa de los casos de prueba.

**Resolución de Vulnerabilidades de Seguridad**

Las pruebas pueden reanudarse una vez que los riesgos de seguridad que llevaron a la suspensión han sido mitigados o eliminados. Se debe realizar una revisión de seguridad preliminar para confirmar que las vulnerabilidades detectadas se han resuelto.

**Entregables**

* Plan de pruebas de software.
* Casos de Pruebas.

**Recursos**

**Requerimientos de Entornos – Hardware**

PC con sistema operativo Windows.

**Requerimientos de Entornos – Software**

Visual Studio Code.

Microsoft Edge.

Django.

**Herramientas de Pruebas Requeridas**

Especifica las herramientas de software, metodologías o técnicas especiales empleadas en las pruebas, por ejemplo Herramientas de Automatización de Pruebas, Software de Gestión de Pruebas, entre otros.

**Personal**

Líder de Pruebas,

**Entrenamiento**

**Entrenamiento en la Funcionalidad del Sistema**

**Objetivo**: Capacitar al equipo para que comprenda el funcionamiento de la aplicación o sistema bajo prueba, su arquitectura, flujo de trabajo, y sus principales módulos.

**Planificación y Organización**

**Procedimientos para las Pruebas**

Especifica los procedimientos o metodología de pruebas a emplear durante la ejecución del plan de pruebas de software.

**Dependencias y Riesgos**

* Dependencias con Desarrollos.
* Disponibilidad de recursos.
* Restricciones de tiempo.
* Premisas que resultan no ser ciertas.