**Plan de pruebas de software**

***[HappyLife]***

***Fecha: [5-09-2024]***

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| <29-08-2024> | <1.0> | Inicio de la creación de plan de pruebas | José Oporto |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

**1** **Introducción 6**

1.1 Resumen ejecutivo 6

1.2 Alcance de las pruebas 6

*1.2.1* *Elementos de pruebas 6*

*1.2.2* *Pruebas funcionales 6*

*1.2.3* *Riesgos 6*

1.3 Definiciones, acrónimos y abreviaciones 8

1.4 Referencias 8

**2** **Requerimientos para pruebas 9**

2.1 Casos de uso 9

*2.1.1* *Vista global 9*

*2.1.2* *Caso de uso 1 9*

*2.1.3* *Caso de uso 2 9*

*2.1.4* *Caso de uso 3 9*

2.2 Requerimientos funcionales

2.3 Requerimientos No-Funcionales 9

**3** **Estrategia de Pruebas 10**

3.1 Tipos de pruebas 10

*3.1.1* *Pruebas funcionales 10*

*3.1.2* *Pruebas de rendimiento (Performance) 11*

*3.1.3* *Pruebas de seguridad y de acceso a datos 11*

*3.1.4* *Herramientas involucradas 11*

**4** **Recursos 12**

4.1 Profesionales 12

4.2 Ambiente de pruebas 13

*4.2.1* *Preparación del ambiente de pruebas 13*

*4.2.2* *Diseño del ambiente de pruebas 13*

*4.2.3* *Diseño ambiente de pruebas 15*

*4.2.4* *Integración del ambiente de pruebas y configuración 16*

*4.2.5* *Generación de datos 16*

**5** **Actividades e Hitos del Plan de Pruebas 16**

**6** **Entregables 19**

6.1 Plan de pruebas 19

*6.1.1* *Criterio de entrada para el “Plan de pruebas” 19*

*6.1.2* *Criterio de salida para el “Plan de pruebas” 19*

*6.1.3* *Criterio de suspensión y resumisión 19*

6.2 Resultados de las pruebas 19

6.3 Reporte de defectos 19

**7** **Anexos 20**

7.1 A: Tareas del proyecto 20

7.2 B: Pruebas de rendimiento (*performance*) 21

7.3 C: Pruebas de seguridad y de control de acceso 22

1. **Introducción**
   1. **Resumen ejecutivo**

*El propósito de este plan de pruebas es validar que la aplicación HappyLife funcione correctamente, proporcionando recomendaciones personalizadas de ejercicio, registro de progreso en salud y acceso a expertos. La aplicación está diseñada para apoyar a los usuarios en la gestión de su salud y bienestar mediante una plataforma móvil y web. Este plan busca asegurar que todas las funcionalidades estén operativas, cumpliendo con los requisitos especificados y garantizando una experiencia de usuario satisfactoria.*

* 1. **Alcance de las pruebas**
     1. *Elementos de pruebas*

* *Web services ( Consultar stock y traer registros de usuarios)*
* *Base de datos*
  + 1. *Pruebas funcionales*

**Recomendaciones de Ejercicio Personalizadas:** Verificar que las recomendaciones sean adecuadas según los datos del usuario.

**Registro y Monitoreo del Progreso de Salud:** Validar la entrada y visualización correcta de datos como peso, presión arterial y glucosa.

**Acceso a Expertos:** Confirmar que los usuarios premium puedan conectarse con entrenadores y médicos.

**Notificaciones Inteligentes:** Probar el envío y recepción de recordatorios y alertas.

Los datos de entrada son los utilizados por las transacciones involucradas. Cada argumento de entrada puede seleccionar uno de los siguientes datos de prueba, dependiendo este del resultado que se desea obtener (esperado), verificando así el comportamiento del componente a probar usando distintos valores de entrada:

* Valores normales para cada transacción.
* Valores límites para cada transacción.
* Valores de borde.
* Valores ilegales.
  + 1. **Riesgos**
* Documentación incompleta o errónea.
* Inconsistencia entre requerimientos y casos de uso.
* Problemas con versiones de componentes.
* Fallos en hardware y software.
* Configuración incorrecta de herramientas de testing automatizado.
  + - 1. Matrices de riesgos
         1. Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | Se Consultará stock de los productos y dira cuanto se tiene de cada uno | alta | El programa consulta y arroja el nombre del producto el número y cuántos hay de cada uno |
| 2 | Se preguntará el listado de clientes para ver quienes tienen descuento | alta | El programa junta todos los usuarios disponibles con sus atributos de manera óptima |
| 3 | se pagará con transbank | alta | Nula |
| 4 | se revisará un producto y se apretara y se mandará al carrito | Alta | Se manda al carrito de compras luego de leer la descripción y se compra |
| 5 | se iniciara sesión introduciendo nombre y clave si se escribe mal una letra o mayúscula lanzará error | Media | Se inicia sesion de manera correcta sin ningun inconveniente |
| 6 | se cierra sesion | media | Se cierra la sesion y no queda ningún indicio de uso anterior |

* 1. **Definiciones, acrónimos y abreviaciones**

* 1. **Referencias**

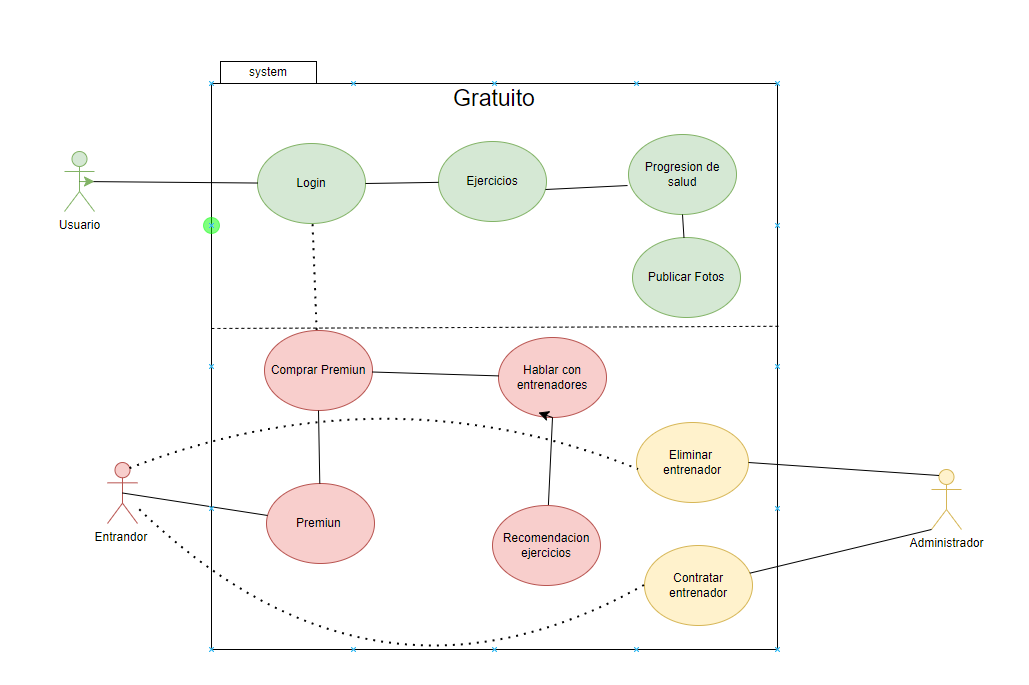
**Norma ISO/IEC 25000:2005**

**Norma ISO/IEC 15408**

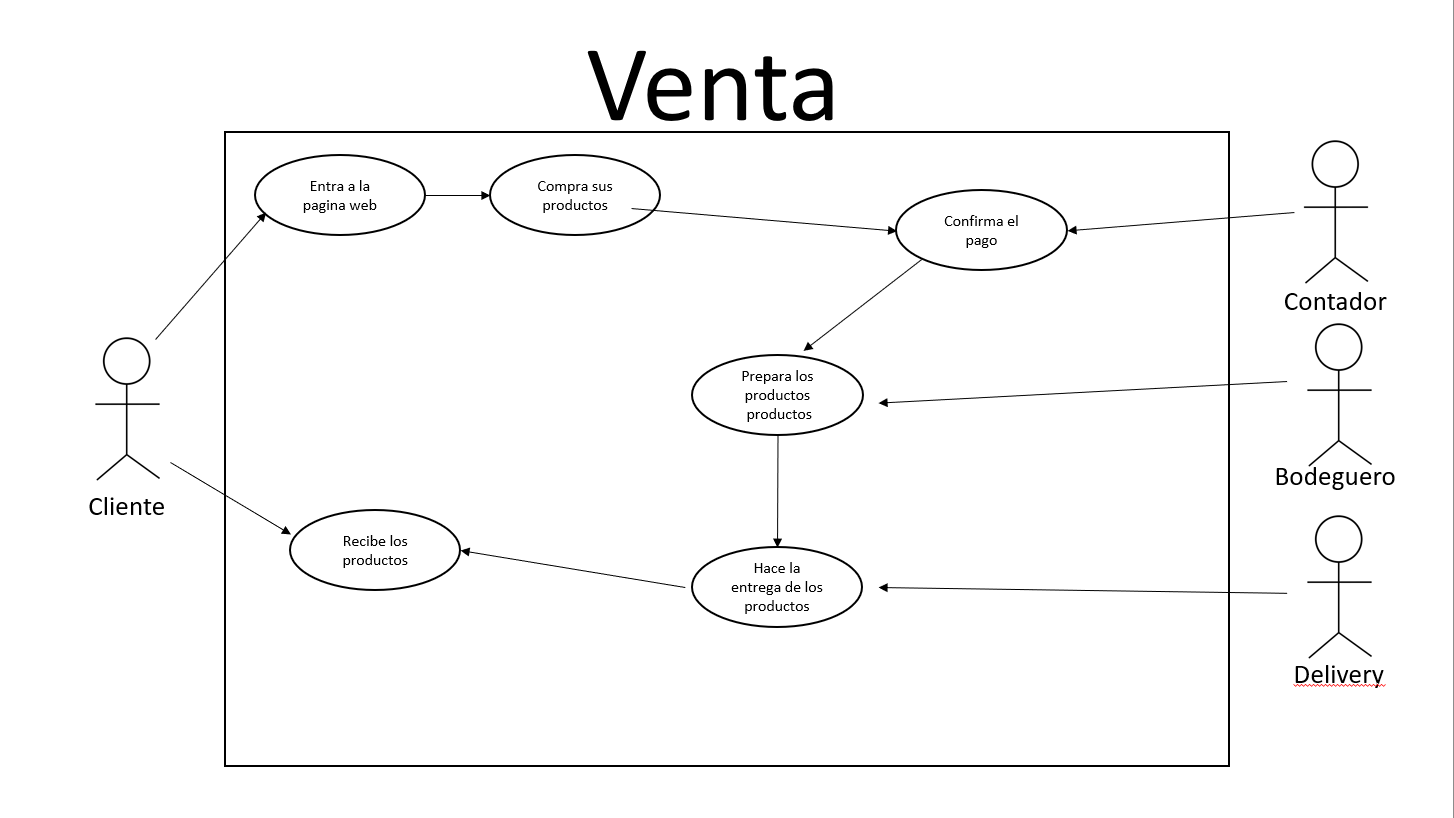
1. **Requerimientos para pruebas**

La siguiente lista identifica los ítems (casos de uso, requerimientos funcionales y requerimientos no funcionales) que se han identificado como requerimientos a ser probados.

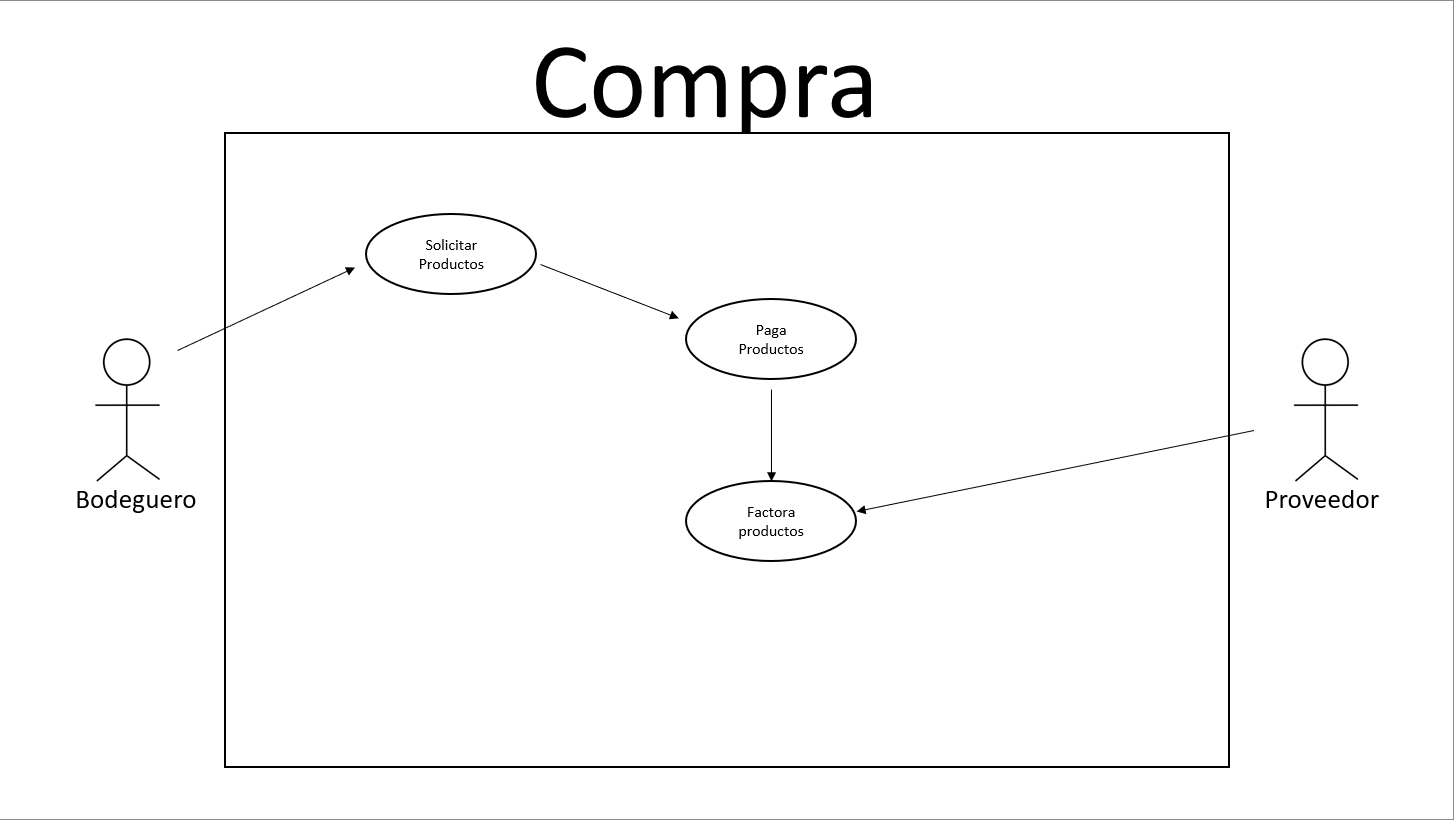
* 1. **Casos de uso**

****

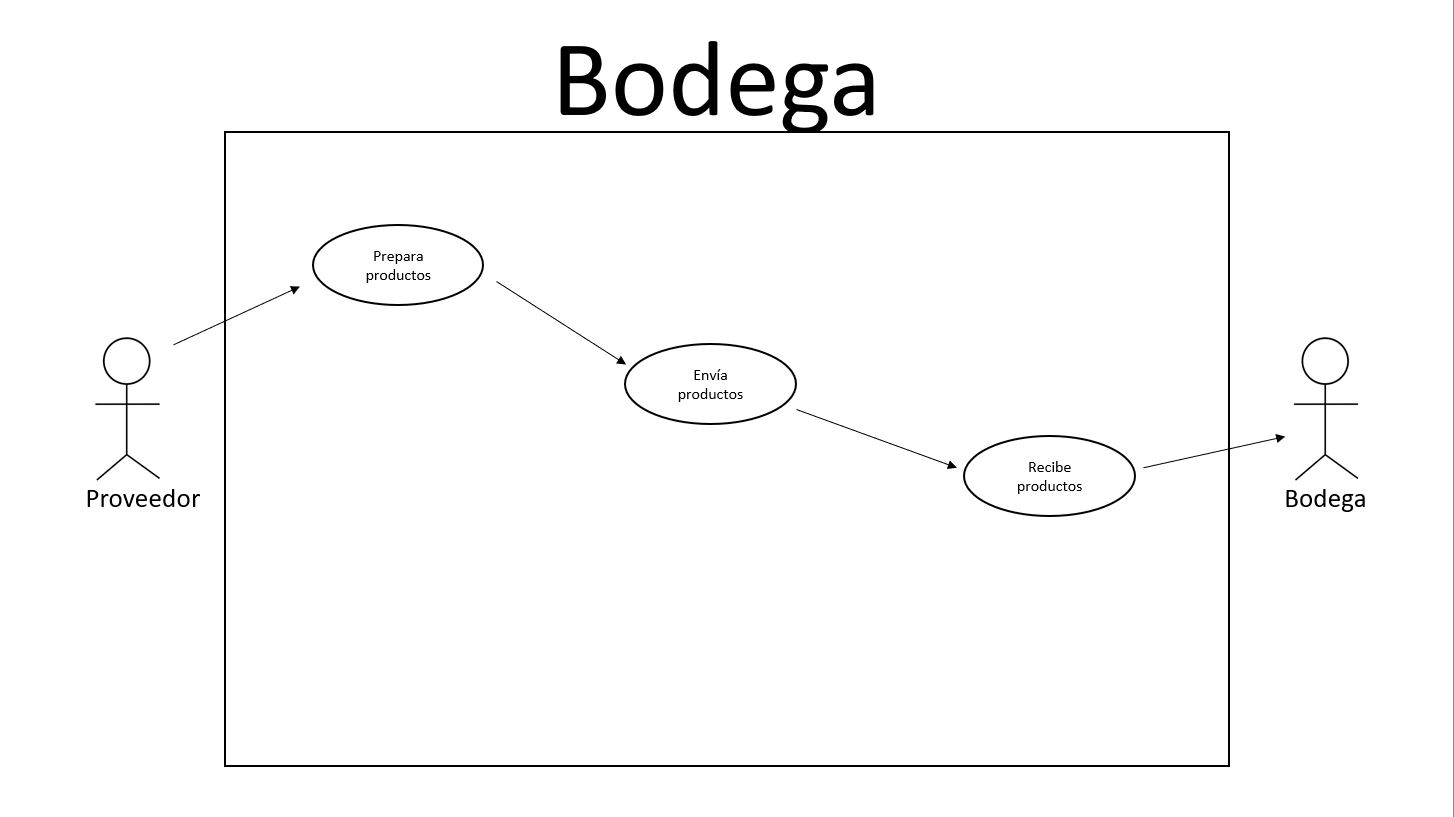
* + 1. *Caso de uso 1: Venta*



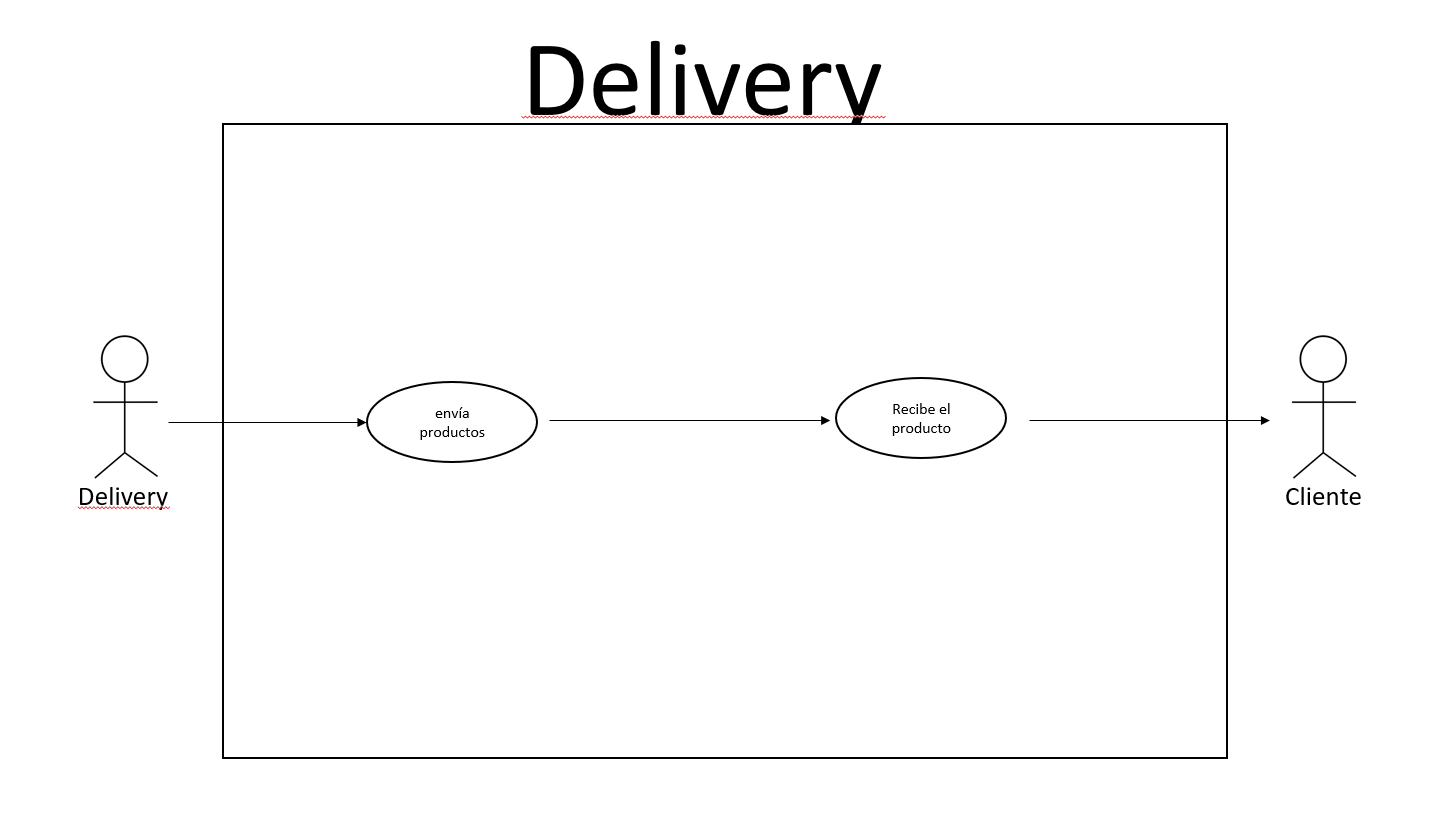
* + 1. *Caso de uso 2: Compró*



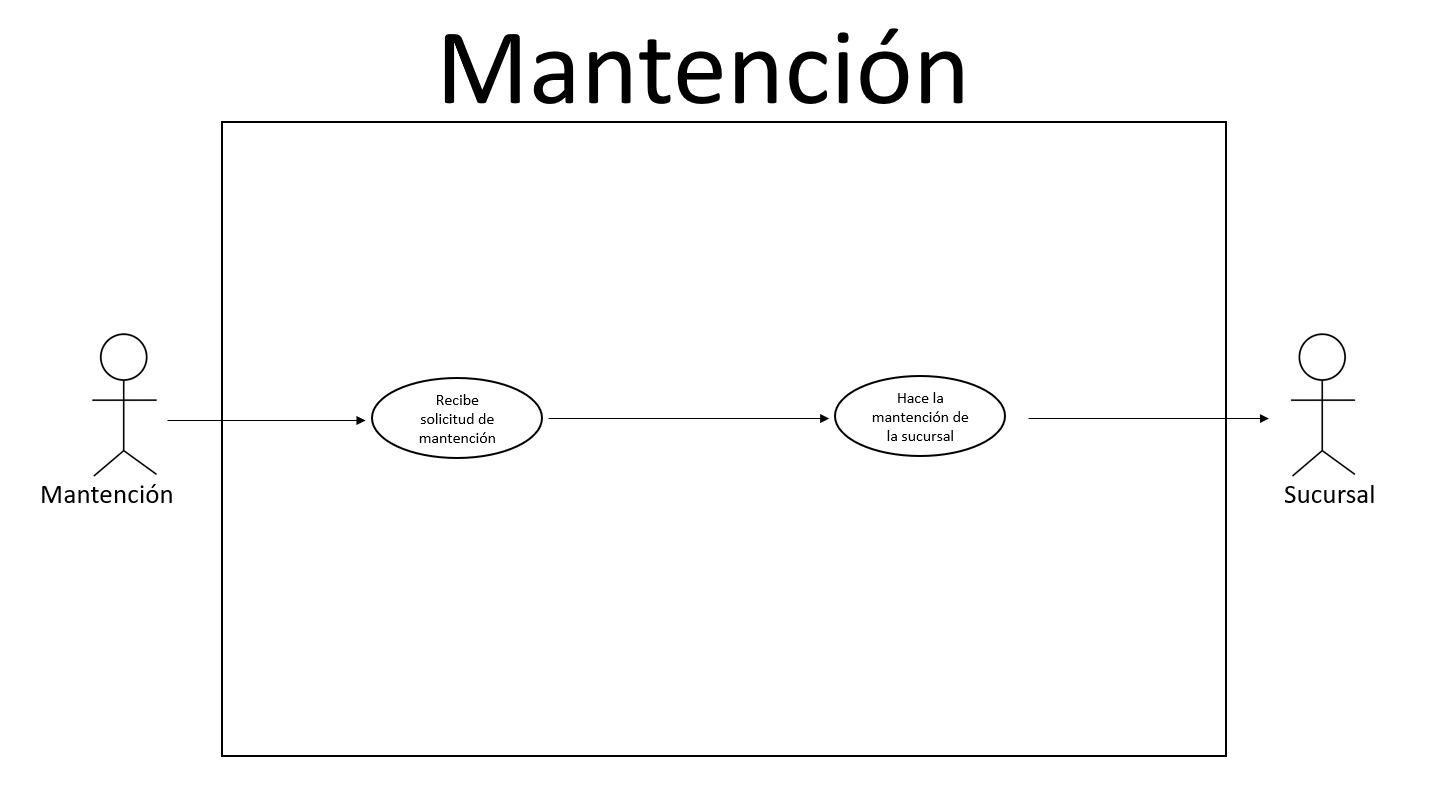
* + 1. *Caso de uso 3: Bodega*



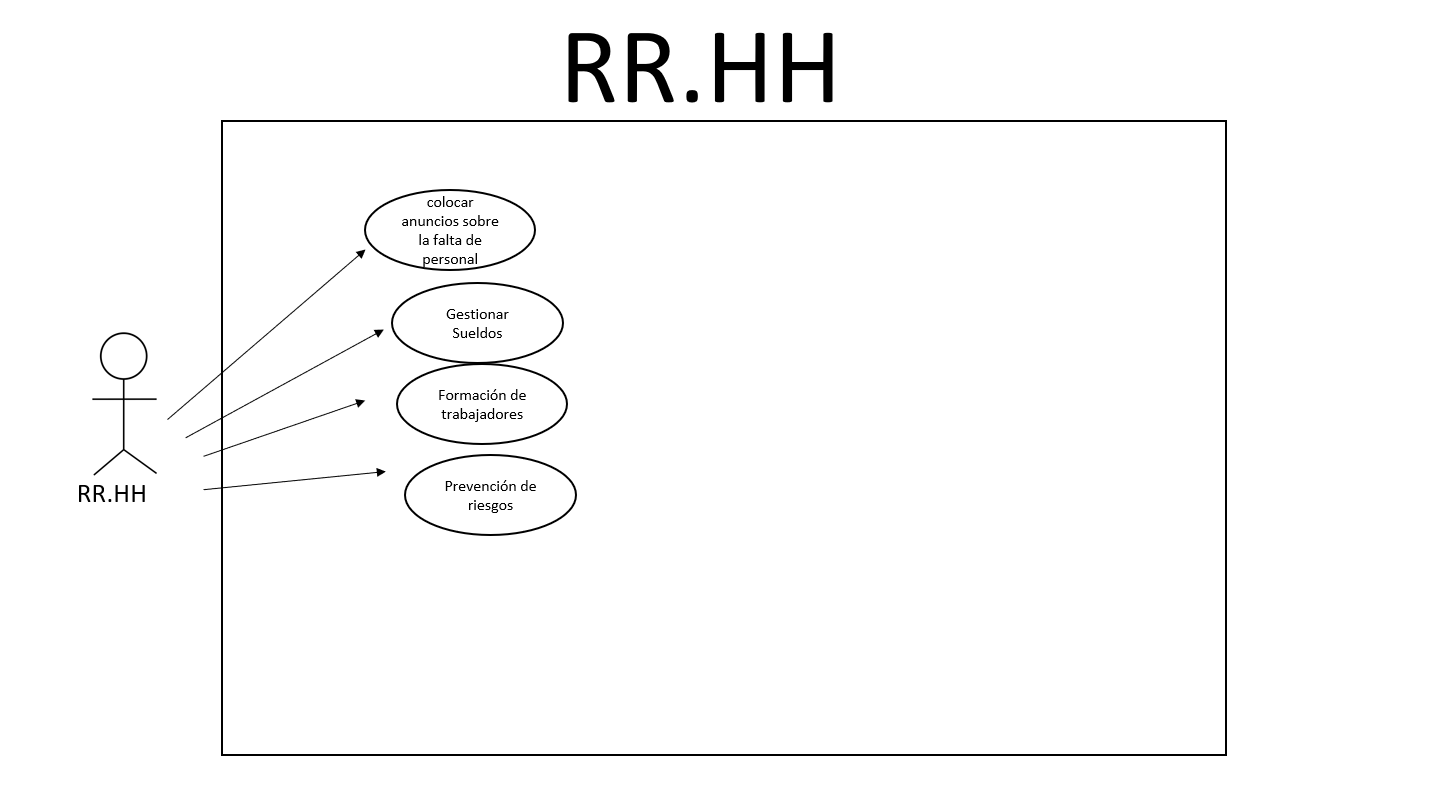
* + 1. *Caso de uso 4: delivery*



* + 1. *Caso de uso 5: Contador*
    2. *Caso de uso 6:Mantención*



* + 1. *Caso de uso 7: Recursos Humanos*



* 1. **Requerimientos funcionales**
     1. *Registro de Usuario*

Descripción: Los usuarios podrán registrar en la página web.

* + 1. *Registro de Usuario*

Descripción: El sistema será capaz de generar un informe sobre las ventas del mes

* + 1. *Generar informe desempeño*

Descripción: El sistema será capaz de generar un informe del desempeño de las tiendas

* + 1. *Generar informe de venta*

Descripción: El sistema será capaz de generar informe de ventas sobre las tiendas

* + 1. *Verificar Stock*

Descripción: los bodegueros y vendedores pueden verificar stock de productos.

* + 1. *Compra de Productos*

Descripción: Los usuarios podrán comprar productos desde la pagina web

* + 1. *Cambiar Contraseña*

Descripción: Los usuarios Podrán Cambiar la contraseña.

* + 1. *Eliminar Usuario*

Descripción: El administrador podrá eliminar usuarios.

* + 1. *Agregar Productos*

Descripción: El administrador y vendedores podrán agregar productos a la pagina web.

* + 1. *Modificar Productos*

Descripción: El administrador y vendedores podrán modificar detalles de los productos en la página.

* + 1. *Eliminar Productos*

Descripción: El administrador y vendedores podrán eliminar productos de la página.

* 1. **Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica)**
     1. *Componente 1 (puede tener uno o más casos de uso)*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 |  |  |  |

* + 1. *Componente 2 (puede tener uno o más casos de uso)*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
|  |  |  |  |

1. **Estrategia de Pruebas**

A través de las semanas el proyecto se llevó de la mejor manera posible así que luego de esto se empezaron aplicar las pruebas, Primero se probará la página web como tal se empezaron a ver los productos y ver si abren y dan una descripción del mismo , luego de esto se probaran los webservices aplicados en este trabajo tal como Revisión de stock que corresponde a Bodega donde se piden productos y esta se conecta a la base y los da revisar este proceso es crucial , Luego de esto se revisará listado de usuarios registrados en la página esto se hace ya que la página al registrarse aplicará descuentos a lo usuarios ya registrados y luego la prueba de transbank y así concluye la estrategia de prueba.

* 1. **Tipos de pruebas**
     1. *Pruebas funcionales*

El *testing* funcional se realizará sobre los requerimientos funcionales antes descritos y sus casos de uso. Estas pruebas tienen por finalidad comprobar la funcionalidad de la aplicación a partir de datos válidamente seleccionados sobre las transacciones del sistema.

Este tipo de comprobación se basa en las técnicas de caja negra, que permiten probar la aplicación (y sus procesos interinos) vía GUI.

| Objetivo de la prueba: | Asegurar la funcionalidad del conjunto de casos, incluyendo la navegación en la aplicación, el ingreso de datos, el proceso y la recuperación (resultados).  Que la navegación a través de los casos de prueba refleje apropiadamente las reglas del negocio y los requerimientos, incluyendo ventana a ventana, campo a campo y usando los métodos de acceso correctamente (tecla *tab*, movimiento del *mouse*, etc.)  Que los objetos de las ventanas y sus características, tales como menús, tamaño, posición, estados y foco, estén de acuerdo a los estándares. |
| --- | --- |
| Técnica a utilizar: | Ejecutar cada caso de uso, su flujo y funcionalidad usando tanto datos válidos como inválidos para comprobar lo siguiente:   * Que los resultados esperados ocurren cuando los datos válidos son utilizados. * Que el mensaje de error es apropiado cuando se utilizan datos inválidos. * Que cada regla de negocio se utiliza apropiadamente. * Crear y modificar los procedimientos de prueba para cada ventana, para comprobar los estados de los objetos y de la aplicación. |
| Criterio de validación: | * Todas las pruebas planificadas se ejecutaron correctamente. * Todos los defectos identificados han sido asignados. * Cada ventana debe ser verificada para mantener la consistencia con la versión maestra y comprobar que esté dentro de los estándares aceptables. |
| Consideraciones especiales: | *están han sido probadas en las pruebas de la 2.1* |
| Observaciones: |  |

* + 1. *Pruebas de rendimiento (Performance)*

Un detalle de este tipo de *testing*, adjunta en el Anexo B de este documento.

* + 1. *Pruebas de seguridad y de acceso a datos*

Un detalle de este tipo de *testing*, adjunta en el Anexo C de este documento.

* + 1. *Herramientas involucradas*

*[Esta sección es solo opcional y aplica en el caso que se utilicen herramientas de testing.]*

*[La siguiente tabla contiene algunos ejemplos.]*

|  | **Herramienta** | **Vendedor/Tienda** | **Versión** |
| --- | --- | --- | --- |
| *Guía de Referencia* | *Rational Unified Process* | *Rational Software Corp.* | *5.5* |
| *Administración de Testing* | *Rational TestManager incluida en el Rational TeamTest* | *Rational Software Corp.* | *7.5* |
| *Seguimiento de Defectos* | *Rational ClearQuest TT Edition incluida en el Rational TeamTest* | *Rational Software Corp.* | *2.0* |
| *Testing Funcional* | *Rational Robot incluido en el Rational TeamTest* | *Rational Software Corp.* | *7.5* |
| *ASQ Tool (performance testing)* | *No definido* |  |  |
| *Test Coverage Monitor or Profiler* | *No definido* |  |  |
| *Herramienta de DBMS* | *Microsoft Access* | *Microsoft Software Corp.* | *97* |
| *SQL Server* | *Microsoft Software Corp.* | *7.0* |

1. **Recursos**
   1. **Profesionales**

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Rol** | **Recursos mínimos recomendados**  **(Número de personas full-time)** | **Responsabilidades específicas / Comentarios** |
| Jefe Proyecto | Mateo Vergara | Responsabilidades   * Construir la carta gantt * Definir las metas * Actúa como un puente de comunicación entre equipos * Mantener una visión general * Dar los roles a cada integrante del equipo |
| Testeador | Yaniro Barrera | Responsabilidades:   * Ejecutar los casos de prueba. * Guardar el estado de los resultados. * Recuperación de errores. * Generar peticiones de cambios en la documentación. |
| Analista | Jean Grasset | Responsabilidades   * Conocimiento en seguridad * Conocimiento de la Arquitectura de software * Conocimiento de la base de datos * Análisis y Modelado de procesos de datos |
| Desarrollador | Byron Saavedra | Responsabilidades:   * Diseño y desarrollo de programas informáticos. * Diseñador de la página web * Diseñador de la base de datos |

* 1. **Ambiente de pruebas**

Se identifican los requerimientos de hardware, software y de comunicación necesarios para crear y dar soporte permanente al Ambiente de pruebas. Las actividades de instalación y configuración para el conjunto de los componentes del Ambiente de pruebas, deberán ser planificadas y calendarizadas. Se requiere que este ambiente sea seguro, estable y dedicado exclusivamente para las pruebas del sistema.

* + 1. *Preparación del ambiente de pruebas*

Las pruebas unitarias y de regresión deberán ser ejecutadas dentro del Ambiente de desarrollo, las pruebas de aceptación del usuario y del sistema se ejecutarán en este Ambiente de pruebas. Este ambiente deberá representar una configuración idéntica al Ambiente de producción o al menos, una versión en menor escala. Esto se requiere debido a que se debe replicar el rendimiento de la línea base y las medidas de mejoramiento relacionadas.

* + 1. *Diseño del ambiente de pruebas*

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Procesador | *Intel Core i5\_9300* |  |
| Memoria RAM | *64 MB (128 MB ideal)* |  |
| Espacio en Disco | *500 MB* |  |
| Tipo Monitor y Resolución | *Ge force GTX 1650 / 1366 x 768* |  |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red | *microsoft kernel debug network adapter* |  |
| Modem |  |  |
| Mouse | *Red dragon centrophorus* |  |
| Tipo Enlace |  |  |
| **Servidores** |  |  |
| ***Base de Datos (pruebas)*** |  |  |
| Procesador | *Intel Core i5\_9300* |  |
| Memoria RAM | *64 MB (128 MB ideal)* |  |
| Espacio en Disco | *500 MB* |  |
| Tipo Monitor y Resolución | *Ge force GTX 1650 / 1366 x 768* |  |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red | *microsoft kernel debug network adapter* |  |
| Modem |  |  |
| Mouse | *Red dragon centrophorus* |  |
| Tipo Enlace |  |  |
| ***Aplicación (pruebas)*** |  |  |
| Procesador | *Intel Core i5\_9300* |  |
| Memoria RAM | *64 MB (128 MB ideal* |  |
| Espacio en Disco | *500 MB* |  |
| Tipo Monitor y Resolución | *Ge force GTX 1650 / 1366 x 768* |  |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red | *microsoft kernel debug network adapter* |  |
| Módem |  |  |
| Mouse | *Red dragon centrophorus* |  |
| Tipo Enlace |  |  |
| **Impresoras** |  |  |
| Marca y Modelo | Hp Smart Tank 515 |  |
| Tipo | Impresión a Color |  |
| Resolución | 1200x1200 |  |
| Rendimiento | 12000 |  |
| Dedicación |  |  |
| **RED** |  |  |
| Topología | Red de estrella |  |
| Medio | Inalámbrico |  |
| Velocidad | 1 ms |  |
| Protocolo |  |  |
| Módems |  |  |
| Conexión Internet |  |  |
| **Sistema de Respaldo / Restauración** |  |  |
| Unidad (Modelo y Marca) | Western Digital/WDBUZG0010BBK-WESN |  |
| Capacidad | 1tb |  |
| Ubicación | Externa |  |
|  |  |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo | Windows |  |
| Herramienta de *testing* | *Selenium, Postman* |  |
| Herramienta de Modelamiento | *Data Modeler* |  |
| BDMA | *Access* |  |
| Browser | *Google Chrome* |  |
| Software de Escritorio | *Microsoft Office* |  |
| **Base de Datos (pruebas)** |  |  |
| Sistema Operativo | Microsoft edge |  |
| Software de Red | Microsoft Office |  |
| Dominio/Cuenta |  |  |
|  |  |  |
| **Aplicación (pruebas)** |  |  |
| Sistema Operativo | Microsoft edge |  |
| Software de Red | Microsoft Office |  |
| Dominio/Cuenta |  |  |
|  |  |  |
| **Repositorios** |  |  |
| Servidor | DELL - SERVIDOR - TOWER - 1 INTEL XEON E3-1200 SERIES E3-1220V6 / 3.06 GHZ - 8 GB DDR SRAM - 2 TB HARD DRIVE CAPACITY |  |
| Dominio / Cuenta | Local |  |
|  |  |  |
| **Seguridad** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* + 1. *Diseño ambiente de pruebas*

El siguiente diagrama muestra la arquitectura del Ambiente de pruebas requerido para realizar las pruebas. La arquitectura del ambiente de pruebas debe ser, en la medida de lo posible, similar a la arquitectura definida para el sistema en producción.

* + 1. *Integración del ambiente de pruebas y configuración*

Para esta actividad se requerirá la participación de profesionales de en cuanto a instalación, configuración y puesta en marcha del Ambiente de Pruebas. Principalmente se requiere del responsable de la Red y Administración de Bases de Datos, de tal forma de obtener un ambiente lo más consistente y similar al de producción, con las bases de datos creadas y el software configurado para asegurar que el sistema funciona de acuerdo a diseño.

Las actividades generales a ser consideradas son:

| **Actividad** | **Responsable** | **Fecha Estimada** | **Fecha Real** | **Observaciones** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

* + 1. *Generación de datos*

Los datos generados fueron basados según las necesidades de los requerimientos, como por ejemplo datos de los clientes, de compra, productos en stock, reportes, etc.

* + - 1. Muestra de producción

Para que la muestra de datos sea realmente representativa, se deberá elegir una fecha testigo adecuada y que posibilite la mayor cobertura de datos. En este sentido se ha elegido como fecha testigo el

Para la obtención de datos por esta vía, se deberán definir las restricciones (por motivos de confidencialidad) y generar algún utilitario para filtrar los datos de tal forma de obtener la mayor variabilidad de datos posible.

Además se deben considerar los siguientes aspectos para asegurar que estos datos funcionen correctamente en el Ambiente de Pruebas, si corresponde:

* Archivos Maestros al Inicio del Día
* Tablas de Parámetros
* Interfaces de Entrada
* Archivos de Movimientos del día o del periodo

Todos estos aspectos se deben considerar en los distintos ambientes donde los datos van a ser utilizados en las transacciones o actualizaciones.

1. **Actividades e Hitos del Plan de Pruebas**

|  | **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** |
| --- | --- | --- | --- | --- | --- |
|  | **Plan de Pruebas** |  |  |  |  |
|  | Identificar el Proyecto | Equipo de trabajo | Aquí se idealiza el proyecto | 1/04/2022 | 01/04/2022 |
|  | Definir Estrategia | Equipo de trabajo | Aquí se propone la estrategia para realizarlo | 01/04/2022 | 01/04/2022 |
|  | Estimar Actividades | Equipo de trabajo | Cuánto esfuerzo necesitará cada una de estas actividades | 02/04/2022 | 02/04/2022 |
|  | Identificar Recursos | Equipo de trabajo | que cosas recursos o medios se necesitan para realizarlo de manera óptima | 02/04/2022 | 02/04/2022 |
|  | Documentar el “Plan de Pruebas” | Equipo de trabajo | Aqui haremos la documentación de nuestro proyecto | 03/04/2022 | 03/04/2022 |
|  | Agenda de Actividades | jefe de proyecto | Aquí se agenda las actividades para seguir con la siguiente fase | 04/04/2022 | 04/04/2022 |
|  | Revisar el “Plan de Pruebas” | analista | Se analiza si este plan estara bien para hacer un proyecto optimo | 04/04/2022 | 04/04/2022 |
|  | **Diseño de las Pruebas** |  |  |  |  |
|  | Analizar Requerimientos | analista | Se analizan los requerimientos de music pro para saber qué se necesita | 06/04/2022 | 06/04/2022 |
|  | Especificar Procedimientos de prueba | analista | Se piensa que pruebas se realizará | 06/04/2022 | 06/04/2022 |
|  | Especificar casos de prueba | analista | Se dice como se harán | 07/04/2022 | 07/04/2022 |
|  | Revisar Cobertura de los requerimientos de prueba | analista | y se cubren para ver si se pueden realizar | 07/04/2022 | 07/04/2022 |
|  | **Implementación de las Pruebas** |  |  |  |  |
|  | Establecer Ambiente de Implementación | equipo de programacio | juntar al equipo par empezar a implementar pruebas | 10/04/2022 | 10/04/2022 |
|  | Desarrollar los Procedimientos de Prueba | equipo de programación | aqui se empiezan los borradores de los procedimientos | 11/04/2022 | 12/04/2022 |
|  | Probar y depurar los procedimientos de prueba | equipo de programación | tener claros que pruebas se aplicaran | 12/04/2022 | 14/04/2022 |
|  | Modificar los procedimientos de prueba | equipo de programación | Se mejoran las pruebas ya establecidas | 15/04/2022 | 17/04/2022 |
|  |  |  |  |  |  |
|  | **Ejecución de las pruebas** |  |  |  |  |
|  | Ejecutar pruebas | Equipo de trabajo | Se prueban el sistema con las pruebas planteadas | 18/04/2022 | 18/04/2022 |
|  | Comprobar resultados esperados | equipo de programacio | Se ve si los resultados de las pruebas son optimos | 18/04/2022 | 18/04/2022 |
|  | Investigar resultados inesperados | analista | ver si hay algun error | 18/04/2022 | 18/04/2022 |
|  | Registrar defectos (log) | analista | Se guardan los defectos del proyecto para corregirlo | 18/04/2022 | 18/04/2022 |
|  | Re-Ejecutar las pruebas | equipo de programación | Se vuelven a hacer pruebas para ver si siguen los mismos errores | 20/04/2022 | 20/04/2022 |
|  | **Evaluación de las pruebas** |  |  |  |  |
|  | Revisar el Log de pruebas | analista | Se revisa el logaritmo para seguir haciendo lo mismo | 22/04/2022 | 22/04/2022 |
|  | Evaluar cobertura de los casos de prueba | Equipo de trabajo | se analiza si la cobertura era la correcta | 23/04/2022 | 23/04/2022 |
|  | Evaluar defectos | Equipo de trabajo | Se anotan los defectos para saber que no se pudo corregir | 24/04/2022 | 26/04/2022 |
|  | Reportar defectos | Equipo de trabajo | Se reportan estos defectos en un informe | 27/04/2022 | 28/04/2022 |

1. **Entregables**

Plan de Prueba / Industria Music Pro / 16-05-2022

BDD / Industria Music Pro / 16-05-2022

Requerimientos funcionales / Industria Music Pro / 16-05-2022

Scripts / Industria Music Pro / 16-05-2022

Diagrama 4+1 / Industria Music Pro / 16-05-2022

* 1. **Plan de pruebas**
     1. *Criterio de entrada para el “Plan de pruebas”*

*Se dará inicio a las pruebas una vez el sistema este creado junto a sus web services y su base de datos*

* + 1. *Criterio de salida para el “Plan de pruebas”*

*El Plan de pruebas se terminará una vez los fallos sean mínimos o nulos y el sistema se pueda ejecutar de buena manera.*

* + 1. *Criterio de suspensión y remisión*
* *El sistema no cumple con los requerimientos exigidos desde un principio*
* *Sistema Inestable*
  1. **Resultados de las pruebas**

Las pruebas estarán registradas en la planilla de plan de pruebas a través del método de prueba llamado “caja negra” donde haremos el registro de servicios web de consultar stock y los usuarios registrados. a través de estos registros podremos establecer un mejor desarrollo y optimización de nuestros servicios web.

* 1. **Reporte de defectos**

3.3.5 Planilla Registro de Defectos Ejemplo.

Anexos

* 1. **A: Tareas del proyecto**

La siguiente lista muestra las tareas relacionadas con el “Plan de pruebas”:

| *✔* | “Plan de pruebas” (preliminar, al inicio del proyecto) |
| --- | --- |
| *✔* | Identificar requerimientos para el testing |
| *✔* | Identificar los riesgos, cuantificar impacto |
| *✔* | Desarrollar la estrategia de pruebas |
| *✔* | Identificar los recursos para las pruebas |
| *✔* | Generar “Plan de Pruebas” detallado |
| *✔* | Diseño general de las pruebas |
|  | Análisis de carga |
| *✔* | Identificar y describir los casos de prueba |
| *✔* | Identificar y estructurar los procedimientos de prueba |
| *✔* | Revisar y accesar la cobertura de las pruebas |
| *✔* | Implementar las pruebas |
| *✔* | Grabar o programar los *scripts* de las pruebas, si aplica |
| *✔* | Identificar las funcionalidades a probar, específicos en el modelo de diseño e implementación |
| *✔* | Establecer el conjunto de datos externos |
| *✔* | Ejecutar las pruebas |
| *✔* | Ejecutar los procedimientos de prueba |
| *✔* | Evaluar la ejecución de las pruebas |
| *✔* | Comprobar los resultados |
| *✔* | Investigar los resultados inesperados |
| *✔* | Registro de defectos, Informe de Resultados |
| *✔* | Evaluar las pruebas |
| *✔* | Evaluar la cobertura de los casos de prueba |
| *✔* | Evaluar la cobertura del código |
|  | Analizar defectos |

* 1. **B: Pruebas de rendimiento (*performance*)**

Para esto, se definen las transacciones de acuerdo a los casos de uso específicos que se espera que un actor del sistema realice usando un conjunto de datos para agregar o modificar transacciones.

| Objetivo de la prueba: | Comprobar la conducta de rendimiento para las transacciones seleccionadas o funcionalidades bajo las siguientes condiciones:  - Una carga de trabajo normal.  - Una sobrecarga de trabajo. |
| --- | --- |
| Técnica a usar: | Usar los procedimientos de pruebas desarrollados para el testing funcional.  Modificar los archivos de datos para aumentar las transacciones o los *script* de robotización para incrementar el número de iteraciones de cada transacción.  Los *script* deberán correr en una máquina (la mejor referencia es un solo usuario y una única transacción) y repetirla con múltiples clientes (virtuales o reales). |
| Criterio de validación: | Una Transacción / Un Usuario: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro del tiempo esperado (por transacción en forma independiente).  Múltiples Transacciones / Múltiples Usuarios: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro del tiempo estimado. |
| Consideraciones especiales: | La extensión del testing de rendimiento requiere tener en *background* la carga de trabajo en el servidor.  Existen varios métodos que se pueden usar para realizar esto como por ejemplo:  Gatillar transacciones directamente al servidor, normalmente en forma de llamadas de SQL.  Crear una carga de usuarios virtuales para simular (normalmente varios cientos) los clientes. Para esto se utilizan herramientas de emulación de terminales remotas para lograr esta carga. Esta técnica también puede usarse para someter a la red a un alto tráfico.  Usar múltiples clientes físicos, cada uno corriendo los Test scripts para agregar una carga al sistema.  El testing de rendimiento debería realizarse en una máquina dedicada o en un tiempo dedicado. Esto permite un control total y una exacta medición.  Las bases de datos utilizadas para realizar el testing de rendimiento deberán ser del tamaño equivalente a las de producción o a escala similar. |
| Observaciones: |  |

* 1. **C: Pruebas de seguridad y de control de acceso**

Se recomienda que el Administrador de la Red y del Sistema planifiquen algunas pruebas en este sentido.

Este *testing* se enfoca en dos áreas claves de la seguridad:

* Seguridad a nivel de la Aplicación, incluyendo acceso a los datos o funciones de negocio, y
* Seguridad a nivel del Sistema, incluyendo el autenticación (*login*) y/o acceso remoto al sistema.
* La seguridad a nivel de la aplicación, asegura que, sobre la base de la seguridad deseada, se restringen a los usuarios a ciertas funciones o casos de uso específicos o se les limita el acceso a datos disponibles para ellos.
* La seguridad a nivel de sistema, asegura que sólo los usuarios definidos en el sistema son capaces de acceder a la aplicación y sólo a través de entradas apropiadas.

| Objetivo de la prueba: | Seguridad a Nivel de Aplicación: comprobar que un usuario puede acceder sólo a las funcionalidades y datos para las cuales ese tipo de usuario tiene permiso.  Seguridad a Nivel de Sistema: comprobar que sólo esos usuarios con acceso al sistema y aplicación tienen permitido el acceso. |
| --- | --- |
| Técnica a usar: | Nivel de Aplicación: Identifique y liste cada tipo de usuario y las funcionalidades y datos de cada tipo para las cuales tiene permiso.  Cree pruebas para cada tipo de usuario y verifique cada permiso creando transacciones específicas para cada usuario.  Modifique los tipos de usuarios y vuelva a ejecutar los casos de prueba para los mismos usuarios. En cada caso verifique si las funcionalidades y los datos están correctamente disponibles o denegados.  Acceso a Nivel de Sistema: vea las consideraciones especiales más abajo. |
| Criterio de validación: | Para cada tipo de usuario conocido, las funcionalidades y los datos correctos debieran estar disponibles y todas las transacciones ejecutadas debieran ejecutarse de acuerdo a lo esperado. |
| Consideraciones especiales: | El acceso al sistema debería ser comprobado con el administrador de la red o del sistema.  Este *testing* quizás pueda requerir de la participación del administrador de la red o del sistema. |
| Observaciones: |  |