**Plan de Pruebas de Software**

***Optimizador de cobranza***

***Fecha: [10/09/2024]***

**Tabla de contenido**

[**Historial de Versiones 3**](#_heading=h.ttcvvxga34ui)

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

[**Aprobaciones 3**](#_heading=h.z06oqwmlemwh)

[**Resumen Ejecutivo 4**](#_heading=h.e3u1ktvwul99)

[**Alcance de las Pruebas 4**](#_heading=h.2et92p0)

[Elementos de Pruebas 4](#_heading=h.slcht3xx97qu)

[Nuevas Funcionalidades a Probar 4](#_heading=h.n058nkcgvt2m)

[Pruebas de Regresión 5](#_heading=h.pds47bn2w1zp)

[Funcionalidades a No Probar 5](#_heading=h.appmcx7bd6t8)

[Enfoque de Pruebas (Estrategia) 5](#_heading=h.tf64cuy2b6ib)

[**Criterios de Aceptación o Rechazo 6**](#_heading=h.17dp8vu)

[Criterios de Aceptación o Rechazo 6](#_heading=h.3rdcrjn)

[Criterios de Suspensión 6](#_heading=h.91zgksa6cwg9)

[Criterios de Reanudación 7](#_heading=h.3168ddm8ahit)

[**Entregables 7**](#_heading=h.35nkun2)

[**Recursos 7**](#_heading=h.3xopa79bv5u2)

[Requerimientos de Entornos – Hardware 7](#_heading=h.44sinio)

[Requerimientos de Entornos – Software 7](#_heading=h.mpvdtr82sdry)

[Herramientas de Pruebas Requeridas 8](#_heading=h.akdfmtuqyz39)

[Personal 8](#_heading=h.3j2qqm3)

[Entrenamiento 8](#_heading=h.7d5b0t3jsyjj)

[**Planificación y Organización 9**](#_heading=h.4i7ojhp)

[Procedimientos para las Pruebas 9](#_heading=h.2xcytpi)

[Responsabilidades 9](#_heading=h.i0i58ici4u39)

[Premisas 9](#_heading=h.s1q4szcjxlrp)

[Dependencias y Riesgos 10](#_heading=h.qsh70q)

[**Referencias 10**](#_heading=h.tgakyfq4xpf)

## Historial de Versiones

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 10/09/2024 | 1.0 | Víctor Silva | Alloxentric | Inicio del documento |
| 10/09/2024 | 2.0 | Víctor Silva | Alloxentric | Finalización del documento |

## Información del Proyecto

| Empresa / Organización | Alloxentric |
| --- | --- |
| Proyecto | Optimizador de cobranzas. |
| Fecha de preparación | 12/08/2024 |
| Cliente | Max Kreimerman |
| Patrocinador principal | Max Kreimerman |
| Product Owner | Carlos Correa |

## Aprobaciones

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Víctor Silva | Desarrollador | TI |  |  |
| Martín Soto | Desarrollador | TI |  |  |

## Resumen Ejecutivo

En este documento se detalla los diferentes puntos que tendrá la aplicación de las pruebas en el software con el propósito de verificar que las funcionalidades implementadas en el sistema funcionen correctamente.

Además se desglosan todos los elementos necesarios para realizar el testeo del software y tener una mejor gestión e ir monitoreando de que las pruebas se ejecuten de la mejor manera, donde se incluye los requerimientos tanto de software como de hardware para una correcta ejecución de pruebas junto con especificar las herramientas para las pruebas, los módulos y funcionalidades que se deben testear.

## Alcance de las Pruebas

#### Elementos de Pruebas

El proceso de pruebas contiene los siguientes elementos a probar:

Módulos:

* Conexiones del sistema.
* Login
* Registro
* Inicio
* Cobranza
* Procesamiento
* Informes.

Elementos:

* Implementación de modelos de machine learning.
* Funcionalidades principales.
* Seguridad.

#### Nuevas Funcionalidades a Probar

* Ejecución del modelo K- Means para agrupar los diferentes clientes
* Ejecución del modelo LSTM para entregar los resultados.

#### Pruebas de Regresión

* Procesamiento de archivos de cobranza..
* Tiempo en la generación de resultados a través del modelo.
* Autenticación del sistema con keycloak que ya está implementado en el sistema de Alloxentric.

#### Funcionalidades a No Probar

En teoría no debería quedar ningún módulo o funcionalidad sin probar ya que puede que alguna afecta mínimamente o de forma crítica al funcionamiento del software.

Aunque es probable que los módulos de Login y Registro no sean considerados como prioridad debido a que la empresa ya cuenta con un sistema de autenticación dentro de los sistemas que son utilizados para las labores de cobranza o de otros proyectos.

#### Enfoque de Pruebas (Estrategia)

**Pruebas Funcionales:**

* Verificar que todas las funcionalidades principales del optimizador de cobranzas funcionen como se espera, como la importación de datos, procesamiento de estrategias de cobranza, generación de reportes, y alertas automáticas.
* Validar los flujos de trabajo completos de usuario (casos de uso) incluyendo creación, edición, eliminación y ejecución de estrategias de cobranza.

**Pruebas de Integración:**

* Probar la integración del optimizador de cobranzas con otros sistemas relevantes, como bases de datos, sistemas de facturación, plataformas de pago, y sistemas de CRM.
* Asegurar que los modelos se integren de la mejor manera en el funcionamiento del sistema para comprobar que se cumplan las tareas a realizar por el trabajadores.

**Pruebas de Desempeño:**

* Medir el tiempo de respuesta y asegurar que se cumplan los tiempos de respuesta aceptables para las operaciones críticas (como la generación de reportes y procesamiento de estrategias).
* Poder optimizar recursos para que el sistema no tenga problemas de rendimiento.

**Pruebas de Usabilidad:**

* Asegurar que la interfaz de usuario sea intuitiva, fácil de usar y cumpla con las expectativas del usuario final.
* Obtener retroalimentación de usuarios finales sobre la experiencia de uso y ajustar la interfaz según sea necesario.

**Pruebas de Seguridad:**

* Verificar que los datos confidenciales de los clientes estén protegidos contra accesos no autorizados.
* Realizar pruebas de penetración para identificar posibles vulnerabilidades en el sistema.
* Asegurar que la autenticación y autorización se implementen de acuerdo con las políticas de seguridad.

## Criterios de Aceptación o Rechazo

#### Criterios de Aceptación o Rechazo

Para tomar en cuenta los criterios de aprobación es importante que se completen todas las pruebas realizadas con la finalidad de que el proyecto se pueda usar de manera adecuada. Estas pruebas principalmente se realizan mediante la implementación de código especializado, en donde se utilizan herramientas como pytest para realizar el proceso de pruebas, por lo que es importante que al ejecutar esas pruebas se vayan mostrando en la consola de la herramienta de código abierto (Visual Studio Code) para verificar que pruebas pasen y se marquen como aprobadas . En cambio si el sistema detecta los casos de prueba como fallidos se deberá corregir lo que sea necesario para que todas las pruebas sean aprobadas.

#### Criterios de Suspensión

Es posible que el proceso de pruebas se suspenda en caso de que el código implementado especialmente para realizar pruebas esté mal organizado o tenga errores de sintaxis junto con que los casos fallidos sean difíciles de corregir.

Por otra parte pueden haber problemas de conexión del backend, ya que si no está funcionando o levantado correctamente será difícil realizar pruebas si este tipo de elementos está mal implementado.

#### Criterios de Reanudación

El proceso de pruebas se reanuda cuando el código ya sea del software o de las pruebas no tengan errores de sintaxis o de implementación, sino que tenga que mostrar por consola si un caso de prueba es aprobado o si ha fallado para sus posteriores correcciones..

## Entregables

* Documento de Plan de Pruebas.
* Planilla de casos de pruebas.
* Mensajes por consola.
* Reportes de las pruebas con la herramienta pytest.
* Registro de defectos.
* Evidencias.

## Recursos

#### Requerimientos de Entornos – Hardware

* **Servidor de Aplicaciones**: Hardware necesario para alojar el servidor de backend.
  + Requisitos mínimos: Procesador 4 núcleos, 8 GB de RAM, 100 GB de espacio en disco (SSD).
* **PCs para Testing**: Equipos con navegadores actualizados y conectividad de red.
  + Requisitos mínimos: Procesador 4 núcleos, 8 GB de RAM.

#### Requerimientos de Entornos – Software

**Backend:**

* Herramienta para construir y gestionar APIs rápidas y eficientes.
* Ejecución automatizada de los endpoints de la API y la lógica de negocio.
* Entorno para desarrollar y ejecutar el backend.

**Frontend:**

* Framework para desarrollar la interfaz de usuario.
* Herramientas de desarrollador de Google Chrome, Firefox o Edge para pruebas de frontend y debugging.

**Base de Datos:**

* Base de datos NoSQL utilizada para almacenar la información.

#### Herramientas de Pruebas Requeridas

**Pytest**: Herramienta de pruebas para ejecutar tests automatizados sobre los endpoints de la API y la lógica de negocio.

**FastAPI:** Herramienta para construir y gestionar APIs rápidas y eficientes.

**Python:** Lenguaje de programación para desarrollar y ejecutar el backend.

**Vue JS:** Framework para visualizar y donde se desarrolla la interfaz de usuario.

**Navegador:** Herramientas de navegación para probar las interacciones de usuario con las diferentes funcionalidades, en este caso se utiliza principalmente Google Chrome.

**Mongodb Compass:** Herramienta gráfica para administrar la base de datos y realizar consultas manuales. Además de ver de que los datos se están guardando correctamente.

#### Personal

* Equipo de desarrollo (2)

#### Entrenamiento

Para esto es necesario levantar el proyecto en entorno local al menos para que todas las funcionalidades se estén ejecutando correctamente. Primero se ejecuta el backend con FastAPI para acceder a la consola respectiva y además el frontend con Vue JS.

En la consola de FastAPI se realizan las respectivas consultas mediante los métodos GET, POST, PUT o DELETE de forma manual para verificar que los datos se estén guardando correctamente y que se puedan visualizar en el frontend. Además de levantar la base de datos para ir gestionando los datos.

## Planificación y Organización

#### Procedimientos para las Pruebas

Para este proceso de pruebas se tomará en cuenta el modelo Agile, ya que el proyecto en sí, está siendo gestionado con la metodología ágil por lo que generalmente se realizarán pruebas al final de cada sprint para verificar que las funcionalidades que se realizan en cada uno funcionen correctamente antes de pasar al siguiente sprint para evitar problemas a futuro.

Incluso la aplicación se probará cuando esté en su fase final con el objetivo de asegurar de que todo rinda de la mejor manera, tomando en cuenta que en el sprint 3 estará destinado a las pruebas de seguridad y en los testeos.

Además se plantea hacer un seguimiento de los riesgos en la mitad de cada sprint para ir resolviendo problemas que pueden surgir del software.

#### Responsabilidades

Las personas integrantes del equipo de desarrollo también estarán a cargo de ejecutar las respectivas pruebas:

* Victor Silva (Equipo de desarrollo) - R
* Martín Soto (Equipo de desarrollo/Scrum Master) - R

#### Premisas

**Limitaciones de tiempo:** se espera que el proceso de pruebas se ejecute dentro del plazo establecido del proyecto y que además en el cronograma o sprint planning estén definidas las fechas en donde se realizarán las pruebas y así evitar atrasos en la entrega final.

**Recursos:** Se asume que el equipo de desarrollo cuenta con recursos open source capacitados ya sea de código abierto como de testing para realizar y ejecutar las pruebas.

**Herramientas de pruebas:** Se espera que se utilicen las herramientas adecuadas y especializadas para realizar el proceso de pruebas como en este caso pytest para la entrega de resultados de que si un caso de prueba aprobó o no.

#### Dependencias y Riesgos

* Dependencias con Desarrollo.
* Restricciones de tiempo.
* Disponibilidad de información para entrenar modelos de machine learning.

## Referencias

* Plan de gestión de riesgos.
* Planilla plan de pruebas.
* Sprint Planning (cronograma del proyecto completo).
* Cronograma de pruebas.
* Endpoints.
* Registro de defectos.