PLUTÓN SAC

PLAN DE LA GESTIÓN DE LA CONFIGURACIÓN

****

**Informe de Proyecto**

**Gerente:** LENIS ROSSI WONG PORTILLO

**Elaborado por:**

* Salazar Herrera, Oscar Miguel
* Sanchez Ríos, Gregory
* Hurtado Saldaña, Jesús Daniel
* Romero Llamoca, Carlos Antonio
* Paredes Inga, Bryan Gerardo
* Pérez Acosta, Roddy David
* Guzman Neyra, Paulo Renato
* Ccana Romero, Jeison Kevin

**Lima - Perú**

**2023**

# Historial de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 04/09/2023 | 1.0 | Grupo 4 | Plutón SAC | Creación del documento |
| 11/09/2023 | 1.1 | Gregory Sanchez | Plutón SAC | Levantamiento de observaciones |
| 20/09/2023 | 1.2 | Grupo 4 | Plutón SAC | Identificación de ítems |
|  | 1.3 |  |  |  |
|  |  |  |  |  |

ÍNDICE

[Historial de versiones 3](#_7av2uniankgs)

[**1. Introducción 4**](#_mek6wsbhw1br)

[1.1 Situación actual 4](#_p2030x3oi77w)

[1.2 Problemática 4](#_vpg5b3dxrl2z)

[1.3 Finalidad 5](#_5khf2j734xk4)

# Introducción

## 1.1 Situación actual

La consultora Plutón actualmente viene desarrollándose en el rubro de Desarrollo de Software con una experiencia de más de 15 años en el mercado, transformando industrias, creando experiencias digitales y reinventando el rendimiento de las plataformas empresariales. Actualmente prestamos nuestros servicios a empresas como BCP, INTERBANK, SCOTIABANK, TOTTUS, entre otros.

Algunos de los proyectos que se desarrollan en PLUTÓN son:

Proyectos de Software:

* Sistema de control de inventario para supermercados.
* Sistema de compra de boletos para viajes terrestres.
* Migración de servicios on premise a la nube.
* Sistema de gestión hipotecaria.
* Sistema de control de código de barras
* Sistema de reconocimiento facial.
* Sistema de contratación de personal para supermercados.
* Sistema de citas médicas.

Software en mantenimiento:

* Sistema de control de inventario para supermercados.
* Sistema de compra de boletos para viajes terrestres.
* Sistema de gestión hipotecaria.

## 1.2 Problemática

* La versión actual del código se sobreescribe por una anterior.

*Sobreescritura de una versión actual por una anterior.*

Implementar un sistema de control de versiones: Utilizar una herramienta de control de versiones como Git es fundamental. Asegúrese de que todos los miembros del equipo lo utilicen correctamente para mantener un historial completo de cambios y evitar sobrescribir versiones anteriores.

* Una actualización crítica se descarta de la versión final.

*Descarte de actualización crítica en la versión final:*

Establecer políticas de revisión de código: Implemente un proceso de revisión de código en el que los miembros del equipo verifiquen que las actualizaciones críticas se incluyan en la versión final. Utilice herramientas de revisión de código como GitHub o GitLab para facilitar este proceso.

* Se hacen cambios en una versión incorrecta del código.

*Cambios en una versión incorrecta del código:*

Establece procedimientos claros para la asignación de tareas y la gestión de versiones. Asegúrese de que los desarrolladores trabajen en la versión correcta del código.

Utilice etiquetas o marcadores en su sistema de control de versiones para identificar las versiones actualizadas y las ramificaciones de forma clara.

* No se logra determinar qué versiones de archivos van en una entrega.

Documenta de manera clara y precisa los archivos y componentes que forman parte de cada entrega. Esto puede incluir un archivo de registro o una lista de control de cambios.

Utilice herramientas de gestión de configuración para rastrear y controlar qué versiones de archivos se incluyen en cada entrega.

Establece un proceso formal de revisión antes de realizar entregas para asegurarte de que todos los elementos necesarios estén incluidos y en las versiones correctas.

Automatiza el proceso de empaquetado y entrega para reducir la posibilidad de errores humanos.

* La necesidad de un repositorio central en la empresa es la congestión y la posible falta de escalabilidad a medida que múltiples equipos y proyectos compiten por el acceso y la gestión de los recursos de almacenamiento y la colaboración, lo que puede ralentizar el flujo de trabajo, aumentar los conflictos en la fusión de código y dificultar la administración eficiente de versiones y la coordinación entre equipos, lo que a su vez puede impactar negativamente en la productividad y la agilidad de desarrollo de la empresa, por ese motivo

## 1.3 Finalidad

Debido a la problemáticas que se presentaron anteriormente, nuestra consultora plantea poner en marcha el plan de Gestión de Software, para así, mejorar la eficiencia y la calidad en el desarrollo y mantenimiento de software. La empresa alcanzará los siguientes objetivos:

Gestionar versiones de software de manera efectiva: La implementación de un sistema de gestión de versiones como Git y políticas de revisión de código permitirá mantener un historial completo de cambios, evitar el sobrescritura y garantizar que las actualizaciones críticas se incluyan en la versión final. Esto ayudará a mantener un control riguroso sobre las versiones, lo que es esencial para la estabilidad y la continuidad de los proyectos.

Asegurar la integridad de las entregas: Evitar cambios en versiones incorrectas del código y garantizar que todas las entregas contengan las versiones correctas de archivos y componentes es crucial para proporcionar software confiable a los clientes por lo que al documentar y automatizar el proceso de empaquetado y entrega, se reduce la posibilidad de errores humanos y asegurar la consistencia en cada entrega.

Mejorar la satisfacción del cliente: Al abordar estas problemáticas, PLUTÓN SAC puede brindar un servicio de desarrollo y mantenimiento de software más confiable y eficiente a sus clientes. Esto se traduce en una mayor satisfacción del cliente, ya que se minimizan los problemas relacionados con la gestión de versiones y la calidad del software entregado.

Optimizar la productividad interna: La implementación de procedimientos claros para la asignación de tareas y la gestión de versiones, junto con herramientas adecuadas, mejora la productividad de los desarrolladores al reducir la pérdida de tiempo causada por la sobrescritura de código y la falta de claridad en las entregas.

1. IDENTIFICACIÓN DE ITEM

2.1. Clasificación

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Item** | **Item** | **Proyecto** | **Extensión** |
| E | Plan de proyecto (Project Charter) | SAMC | docx |
| E | Cronograma del proyecto | SAMC | xlsx |
| E | Repositorio GitHub | SAMC | git |
| E | Estructura del Repositorio | SAMC | git |
| E | Documento Historia de Usuario #1 | SAMC | docx |
| E | Documento Historia de Usuario #2 | SAMC | docx |
| E | Documento Historia de Usuario #3 | SAMC | docx |
| E | Documento Historia de Usuario #4 | SAMC | docx |
| E | Documento Historia de Usuario #5 | SAMC | docx |
| E | Documento Historia de Usuario #6 | SAMC | docx |
| E | Documento Historia de Usuario #7 | SAMC | docx |
| E | Documento Historia de Usuario #8 | SAMC | docx |
|  | Documento de Caso de Uso Ver y editar Perfil | SAMC |  |
|  | Documento de Caso de Uso Registrar Paciente | SAMC |  |
|  | Documento de Caso de Uso Reservar cita | SAMC |  |
|  | Documento de Caso de Uso Recibir Recordatorio | SAMC |  |
|  | Documento de Caso de Uso Visualización de horarios y especialidades | SAMC |  |
|  | Documento de Caso de Uso Cancelación o reprogramación de citas | SAMC |  |
|  | Documento de Caso de Uso Visualizar citas programadas | SAMC |  |
|  | Documento de Caso de Uso Atender cita | SAMC |  |
|  | Prototipo de Sistema | SAMC |  |
|  | Documentación de Especificación de la BD | SAMC |  |
|  | Documentación de la Arquitectura de Software | SAMC |  |
|  | Código de requisito 1 | SAMC |  |
|  | Código de requisito 2 | SAMC |  |
|  | Código de requisito 5 | SAMC |  |
|  | Reporte del Desarrollo del Software | SAMC |  |
|  | Reporte del primer Sprint | SAMC |  |
|  |  |  |  |
|  | Documento de Caso de Uso Reservar cita | SAMC |  |
|  | Documento de Caso de Uso Visualizar citas programadas | SAMC |  |
|  | Documento de Caso de Uso Atender cita | SAMC |  |
|  | Código de requisito 3 | SAMC |  |
|  | Código de requisito 7 | SAMC |  |
|  | Código de requisito 8 | SAMC |  |
|  | Reporte de análisis estatico código 2 | SAMC |  |
|  | Reporte del desarrollo del software | SAMC |  |
|  | Reporte del segundo sprint | SAMC |  |
|  |  |  |  |
|  | Documento de Caso de Uso Recibir Recordatorio | SAMC |  |
|  | Documento de Caso de Uso Verificar Aseguramiento | SAMC |  |
|  | Código de requisito 4 | SAMC |  |
|  | Código de requisito 6 | SAMC |  |
|  | Reporte de análisis estatico código 3 | SAMC |  |
|  | Reporte del desarrollo del software | SAMC |  |
|  | Reporte del tercer sprint | SAMC |  |
|  | Acta de cierre del proyecto | SAMC |  |

2.2. Nomenclatura

2.3. Diseño de Repositorio

2.4. Linea Base