**DELU**

**Plan de Gestión de la Configuración del Software**

Integrantes

Calle Huamantinco, Luis Eduardo

Calongos Jara, Leonid

Flores Cóngora, Paolo Luis

Pariona Molina, Matthew Alexandre

Calderón Matias, Diego Alonso

Luján Vila, Frank José

Versión 1.0

**Control de versión**

| **Fecha** | **Autor** | **Descripción** |
| --- | --- | --- |
| 19/09/2024 | Matthew Pariona Molina | Estudio de la problemática de la empresa |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Introducción**
   1. Situación de la empresa y problemática

La empresa "DELU" ha estado trabajando en diversos proyectos durante los últimos años, mostrando un crecimiento significativo recientemente. Los proyectos que DELU posee son DeluStock, FitClue y ProfeSoft. **"DeluStock"** es una plataforma web diseñada para pequeñas y medianas empresas que facilita el control de productos, el registro de entradas y salidas, alertas de stock bajo y la generación de reportes. Actualmente, se encuentra en fase de mantenimiento, con algunos problemas de rendimiento identificados. La empresa está evaluando la incorporación de nuevas funcionalidades, como la integración con sistemas de pago y facturación electrónica. **"FitClue"** es una aplicación para dispositivos móviles enfocada en el bienestar y fitness, que ofrece planes de entrenamiento personalizados, seguimiento del progreso, recomendaciones nutricionales y clases en vivo con entrenadores. Actualmente, está en desarrollo, con el diseño de la interfaz de usuario casi finalizado. Sin embargo, la empresa debe optimizar la integración con dispositivos wearables y mejorar la experiencia del usuario. **“ProfeSoft”** es una plataforma web destinada a los estudiantes de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM. Permite evaluar anónimamente a los profesores, calificar y dejar comentarios sobre su desempeño. Su objetivo es proporcionar información útil para que los alumnos tomen decisiones informadas al elegir a sus docentes y fomentar la mejora en la calidad de la enseñanza.

A pesar del crecimiento, la gestión de desarrollo en DELU presenta varios desafíos. El desarrollador que estuvo trabajando solo durante estos años no utilizaba ningún sistema de control de versiones, lo que significa que no hay un historial organizado de cambios, versiones o colaboraciones. Además, no existía un plan claro para la colaboración en proyectos ni un enfoque para el trabajo en equipo, lo que dificulta la transición hacia un equipo más grande. Las pruebas del software y los despliegues en producción se hacían de manera manual, lo que introduce errores y riesgos en cada entrega, y podría ser un obstáculo importante para el crecimiento futuro de la empresa y la escalabilidad de sus proyectos.

* 1. El objetivo del plan

Para enfrentar la falta de control de versiones, la dificultad de trabajo en grupo y la gestión manual de despliegues y pruebas, “DELU” establecerá una organización en GitHub. Esta organización permitirá alojar todos los proyectos en repositorios individuales bajo un único nombre de organización. Con esta estructura, “DELU” podrá asignar roles administrativos adecuados, mejorar la colaboración entre el equipo de desarrollo y automatizar funciones y procesos clave, como el control de versiones, despliegues y pruebas, aumentando así la eficiencia y el rendimiento del equipo.

1. **Gestión de Configuración de Software**
   1. Roles y responsabilidades

| Roles | Responsabilidades |
| --- | --- |
| Gestor de la configuración | Define, aplica y supervisa el proceso de gestión de versiones y cambios en el software. |
| Inspector de Aseguramiento de Calidad | Auditar la configuración y asegurar que el software cumple con los estándares de calidad. |
| Desarrollador Principal | Implementar las configuraciones y coordinar los aspectos técnicos del desarrollo de software |
| Bibliotecario | Definir y mantener las bibliotecas usadas en la gestión de configuración. Asegurarse de que los aspectos prácticos de la gestión de configuración funcionen adecuadamente entre sí. |
| Comité de Control de Cambios | Evaluar registro de eventos, gestionar solicitudes de cambios relevantes y hacer seguimiento de su ciclo de vida. Proveer apoyo y coordinar con los otros comités de control de cambios. |

* 1. Herramientas, entorno e Infraestructura

| **CRITERIOS** | **Github** | **Gitlab** | **Bitbucket** |
| --- | --- | --- | --- |
| Gratuito | Gratuito | Gratuito | Gratuito |
| Curva de aprendizaje | Moderada | Moderada | Moderada |
| Centralizado | Sí | Sí | Sí |
| Distribuido | Sí | Sí | Sí |
| Control de roles y permisos | Sí | Sí | Sí |
| Automatización | Sí | Sí | Sí |
| Integraciones | Amplio soporte con herramientas de CI/CD externas (Jenkins, CircleCI, etc.) y GitHub Actions integrado | GitLab CI/CD nativo y robusto, permite pipelines complejos sin necesidad de herramientas externas | Integración fuerte con Jira y otros productos de Atlassian |

* 1. Herramienta elegida

Como herramienta de control de versiones utilizaremos GitHub. Si bien las tres alternativas son similares en funcionamiento y flujo de trabajo, GitHub aparece como la más popular y familiar para la mayoría de desarrolladores de hoy en día. Además, ofrece características sobresalientes cuando se trata de automatización y DevOps gracias a sus integraciones y el ecosistema que rodea GitHub actions, posicionándose como la herramienta ideal para nuestro trabajo.

