



Aseguramiento de la Calidad de Software.

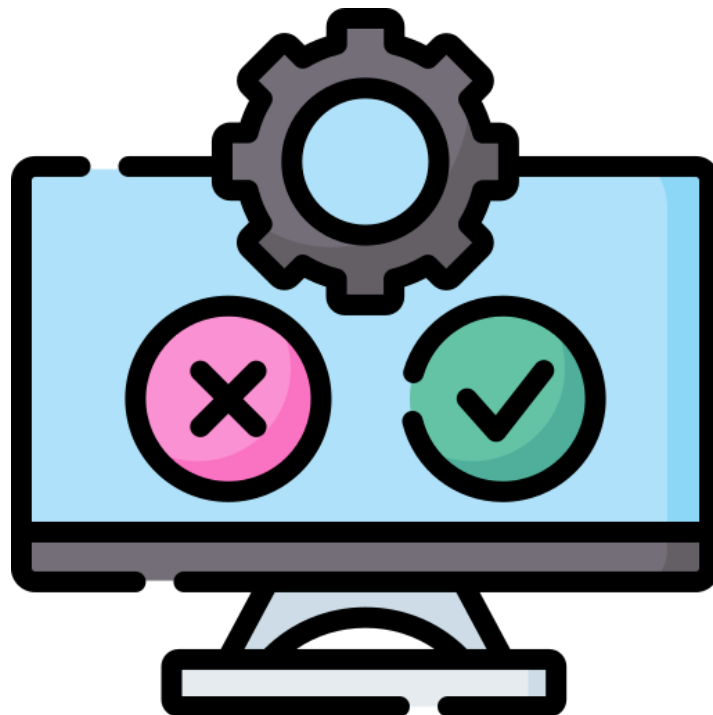
Desarrollar y ejecutar un plan de administración de la configuración del proyecto.

Alumno:

- Arce Montoya José Antonio (1271264).
- Gomez Betancourt Axel (1285032).

Grupo: 371

Profesora: Diana Cristina Ruiz Alvarez.





Introducción.

La administración de la configuración es una práctica fundamental en el desarrollo de software, que asegura que los productos sean desarrollados y configurados de una forma coherente y controlada a lo largo del tiempo.

Una buena administración de la configuración es importante, ya que estos ajustes permiten que el sistema funcione de una forma específica. Antes de lanzar la aplicación o producto, se deben configurar parámetros como la asignación de memoria, hardware, privilegios de acceso, red, etc. Si alguno de estos ajustes no se aplican adecuadamente o no son aplicados, la aplicación o producto será inestable o directamente no va a funcionar.

A lo largo de esta actividad diseñamos un plan de administración de la configuración con el objetivo de mejorar la calidad, trazabilidad y mantenibilidad del proyecto.



Objetivos del plan.

Como objetivo principal de este plan tenemos el garantizar que todos los cambios realizados en el proyecto sean de forma controlada, permitiéndonos así mantener una coherencia entre versiones.

Objetivos específicos:

- Asegurar que el software se pueda reproducir de forma consistente en diferentes entornos.
- Gestionar y controlar los cambios en el código.
- Usar herramientas de automatización de la gestión de la configuración, facilitando así el seguimiento y trazabilidad de los cambios.
- Establecer un proceso claro para realizar cambios en el sistema, incluyendo revisiones y aprobaciones.



Estructura del plan.

El plan de administración de la configuración se divide en los siguientes componentes.

- **Gestión de código fuente:** El código fuente será gestionado utilizando un sistema de control de versiones, durante el desarrollo de este proyecto se utilizará Git. Esto permitirá a los desarrolladores trabajar de manera colaborativa sin conflictos y la prevención de errores permitiendo recuperar versiones previas del proyecto.
- **Gestión de configuraciones:** Se gestionarán todos los archivos de configuración del proyecto, incluyendo los archivos de configuración del entorno de desarrollo, pruebas y producción. Esto asegurará que el proyecto pueda ser desplegado y ejecutado en diferentes entornos sin inconsistencias.
- **Control de versiones:** Todos los cambios en el código y la gestión de versiones será a través de Git, utilizando un repositorio centralizado, en este caso GitHub. Cada versión del código estará documentada con el propósito de poder retroceder a una versión anterior en caso de ser necesario.
- **Proceso de cambios:** Se incluirá revisiones y aprobaciones de los cambios antes de que se fusionen al repositorio principal. Cada desarrollador poseerá una rama donde realizarán las modificaciones, las cuales serán revisadas y aprobadas por el equipo antes de que se fusionen al repositorio principal.
- **Auditoría y trazabilidad:** Cada cambio será documentado en los mensajes de los commits de Git, se llevará un registro de todas las versiones y cambios importantes realizados durante el ciclo de vida del proyecto.



Procesos definidos.

- **Control de versiones.**

- Ramas para el desarrollo de nuevas funcionalidades o corrección de errores que serán evaluadas antes de fusionarse a la rama principal.
- Uso de pull request para garantizar que los cambios sean revisados por otros miembros del equipo antes de ser integrados.

- **Gestión de configuraciones.**

- Archivos de configuración para entornos de desarrollo, pruebas y producción se mantendrán separados para evitar conflictos.
- Usar Docker para garantizar que los entornos de desarrollo sean consistentes y reproducibles.

- **Proceso de revisión y aprobación.**

- Todos los cambios deben ser revisados y aprobados por al menos un miembro del equipo antes de ser fusionados.
- Se utilizarán code reviews en las pull requests para asegurar la calidad del código.

- **Auditoría:**

- Uso de herramientas de integración continua como Jenkins o Selenium para realizar pruebas automatizadas cada vez que se realice un cambio.
- Registro de todos los cambios en el repositorio con descripciones claras sobre lo que se ha modificado.



Herramientas.

Las herramientas que escogimos para llevar a cabo este plan de administración de la configuración nos ayudarán a gestionar el código, realizar integración continua y configurar el entorno.

Dichas herramientas permiten una mayor automatización, control de versiones, trazabilidad de los cambios y consistencia en los entornos de desarrollo y producción.

- Git: Git es una herramienta muy conocida y fundamental cuando hablamos de gestionar código y permitir colaboración entre varios desarrolladores. Cada cambio que realicemos en el código es documentado por medio de commits y pull, lo que garantiza la trazabilidad y facilita enormemente el trabajo en equipo.
- Jenkins: Esta herramienta será utilizada para la integración continua del proyecto. Cada vez que se realiza un commit, Jenkins ejecuta automáticamente pruebas unitarias y de integración, asegurándonos así de que los cambios no van a perjudicar al sistema.
- Puppet: Puppet será utilizado para gestionar y automatizar la configuración de los entornos de desarrollo y producción. Esta herramienta permitirá definir configuración de infraestructura como código, garantizando que todos los entornos sean consistentes y fáciles de replicar.



Conclusión.

La implementación de un plan de administración de la configuración es fundamental para garantizar que el proyecto se desarrolle de manera controlada, coherente y eficiente. Al utilizar herramientas como Git, Jenkins, Puppet y Selenium, nos aseguramos de que todos los miembros del equipo trabajen de manera colaborativa y que los cambios sean gestionados de forma adecuada, así como verificar que las integraciones no tengan conflictos o empeoren la versión actual.

Asimismo, a través del manejo de versiones se permite recuperar el trabajo realizado de manera rápida en caso de tener una pérdida y con la documentación de características agregadas por versión, el equipo tendrá contemplado qué aspectos trabajar en la versión recuperada.

De esta forma nos aseguramos de mejorar la calidad y mantenibilidad del software. Este plan ayudará a evitar errores y facilita el mantenimiento del producto.



Referencias.

- ¿Qué es la administración de la configuración?: explicación de la administración de la configuración del software: AWS. (s. f.). Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/configuration-management/#:~:text=La%20administraci%C3%B3n%20de%20la%20configuraci%C3%B3n%20es%20el%20proceso%20que%20garantiza,a%20lo%20largo%20del%20tiempo.>
- Ciberseg. (2022, April 20). Principales herramientas de gestión de la configuración que debes conocer. Ciberseguridad.
<https://ciberseguridad.com/herramientas/gestion-configuracion/>
- Atlassian. (n.d.). Gestión de configuración: definición y ventajas | Atlassian.
<https://www.atlassian.com/es/microservices/microservices-architecture/configuration-management>
- Administración de la configuración. Organización y control para lograr el éxito. (n.d.). SG Buzz. <https://sg.com.mx/content/view/412>
- ¿Qué es la gestión de la configuración? (n.d.).
<https://www.redhat.com/es/topics/automation/what-is-configuration-management>
- IBM. (2024, mayo 16). Gestión de la configuración. IBM. Recuperado 15 de noviembre de 2024, de
<https://www.ibm.com/mx-es/topics/configuration-management>