

Evidencia de conocimiento: GA7-220501096-AA1-EV01
informe técnico de plan de trabajo para
construcción de software



Presentado por: Josué David Salamanca

PROFESOR: MIGUEL ANGEL SUAREZ

JUAN CAMILO OSPINA

INTRODUCCIÓN

En el desarrollo de software, la capacidad de gestionar y controlar los cambios en el código fuente es fundamental para garantizar la calidad y la eficiencia del proceso de creación. Las herramientas y tecnologías de versionamiento desempeñan un papel crucial en esta tarea, permitiendo a los equipos de desarrollo llevar un registro detallado de las modificaciones realizadas, facilitar la colaboración entre los miembros del equipo y asegurar la integridad del proyecto.

Este informe técnico tiene como objetivo proporcionar una visión general de los principales elementos de un lenguaje de programación, con un énfasis especial en Java, y analizar las herramientas y tecnologías de versionamiento más utilizadas en la industria del software. Al comprender los fundamentos de la programación orientada a objetos y las características clave de Java, junto con el conocimiento de las soluciones de control de versiones disponibles, los lectores podrán adquirir una perspectiva completa sobre la importancia del versionamiento en el desarrollo de aplicaciones de calidad.

A lo largo del documento, se abordarán temas como la estructura de las clases, la creación de objetos, la gestión de atributos y métodos, los modificadores de acceso y la sobrecarga de métodos, entre otros aspectos relevantes del lenguaje de programación Java. Posteriormente, se presentarán las principales herramientas y tecnologías de versionamiento, como Git, GitHub, GitLab, Apache Subversion y Mercurial, destacando sus características, ventajas y casos de uso.

Con esta información, los lectores estarán mejor preparados para comprender la importancia del control de versiones en el desarrollo de software y podrán aplicar estas prácticas en sus propios proyectos, contribuyendo a la mejora continua y la eficiencia de los procesos de creación de aplicaciones.

OBJETIVO

El objetivo principal de este informe técnico es proporcionar una comprensión sólida de los elementos fundamentales de un lenguaje de programación, con un enfoque específico en Java, y analizar las principales herramientas y tecnologías de versionamiento utilizadas en el desarrollo de software.

De manera más específica, este informe tiene los siguientes objetivos:

Explicar los conceptos clave de la programación orientada a objetos, incluyendo la definición de clases, la creación de objetos, el manejo de atributos y métodos, y las convenciones de nomenclatura en Java.

Describir los diferentes modificadores de acceso en Java (público, privado, protegido y por defecto) y su impacto en la restricción y control del alcance de los miembros de una clase.

Analizar la importancia de los métodos accesorios (get y set) y la sobrecarga de métodos como mecanismos para encapsular y manipular los datos de un objeto.

Proporcionar una visión general de las principales herramientas y tecnologías de versionamiento, como Git, GitHub, GitLab, Apache Subversion y Mercurial, destacando sus características, ventajas y casos de uso en el desarrollo de software.

Ilustrar cómo la integración de los conceptos de programación orientada a objetos y las prácticas de control de versiones pueden mejorar la calidad, la colaboración y la eficiencia en el proceso de desarrollo de aplicaciones.

Al cumplir con estos objetivos, este informe técnico busca equipar a los lectores con los conocimientos y herramientas necesarios para comprender la importancia del versionamiento en el desarrollo de software y aplicar estas prácticas de manera efectiva en sus propios proyectos.

SELECCIÓN DE HERRAMIENTAS DE VERSIONAMIENTO A UTILIZAR

Git:

- Git es un sistema de control de versiones distribuido que permite a los desarrolladores llevar un registro detallado de los cambios en el código fuente.
- Ofrece características avanzadas para la colaboración, ramificación, fusión y resolución de conflictos.
- Es ampliamente utilizado en la industria del software y se ha convertido en un estándar de facto.

GitHub:

- GitHub es una plataforma de desarrollo colaborativo que utiliza Git como sistema de control de versiones.
- Permite a los desarrolladores almacenar, compartir y colaborar en proyectos de software.
- Proporciona funcionalidades adicionales, como gestión de problemas (issues), revisión de código y despliegue.

GitLab:

- GitLab es otra plataforma de desarrollo colaborativo que también utiliza Git como sistema de control de versiones.
- Ofrece características similares a GitHub, como repositorios, control de versiones, herramientas de colaboración y despliegue.
- Cuenta con una comunidad activa y un conjunto de características avanzadas para proyectos de mayor envergadura.

Apache Subversion (SVN):

- Subversion es un sistema de control de versiones centralizado que permite a los desarrolladores llevar un registro de los cambios en el código fuente.
- Cuenta con una larga trayectoria y es una opción viable para aquellos equipos que prefieren un enfoque más tradicional.
- Proporciona características básicas de control de versiones y colaboración.

Estas herramientas de versionamiento representan las soluciones más destacadas y ampliamente adoptadas en la industria del software. Al analizar sus características, ventajas y casos de uso, el informe brindará a los lectores una visión completa de las opciones disponibles para gestionar eficazmente los cambios en el código fuente de sus proyectos.

INFORME

En este informe se abordarán los principales elementos de un lenguaje de programación, con un enfoque en Java. Se cubrirán temas como clases, objetos, convenciones de nomenclatura, instanciación de objetos, atributos, métodos, constructores, modificadores de acceso, métodos accesorios y sobrecarga de métodos.

Herramientas y tecnologías de versionamiento:

Las principales herramientas y tecnologías de versionamiento utilizadas en el desarrollo de software son:

Git: Git es un sistema de control de versiones distribuido que permite a los desarrolladores llevar un registro de los cambios en el código fuente, colaborar en proyectos y gestionar las diferentes versiones del software.

GitHub: GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores almacenar, compartir y colaborar en proyectos de software.

GitLab: GitLab es otra plataforma de desarrollo colaborativo que también utiliza Git como sistema de control de versiones. Ofrece características similares a GitHub, como repositorios, control de versiones, herramientas de colaboración y despliegue.

Apache Subversion (SVN): Subversion es un sistema de control de versiones centralizado que permite a los desarrolladores llevar un registro de los cambios en el código fuente y colaborar en proyectos.

Mercurial: Mercurial es un sistema de control de versiones distribuido similar a Git, que también permite a los desarrolladores gestionar y colaborar en proyectos de software.

TortoiseSVN y TortoiseGit: Estas son herramientas de interfaz gráfica de usuario (GUI) que se integran con los sistemas de control de versiones Subversion y Git, respectivamente, facilitando su uso para los desarrolladores.

Estas herramientas y tecnologías de versionamiento son fundamentales en el desarrollo de software, ya que permiten a los equipos de desarrollo llevar un registro de los cambios en el código, colaborar en proyectos, gestionar diferentes versiones y facilitar el trabajo en equipo.