



TECNÓLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE FICHA
2627040 -2627039

Evidencia de producto:

GA7-220501096-AA1-EV01, EV02, EV03 - Informe técnico de plan de trabajo para
construcción de software

GA7-220501096-AA1-EV02 definir estándares de codificación de acuerdo a
plataforma de desarrollo elegida

GA7-220501096-AA1-EV03 identificar herramientas de versionamiento

Aprendiz

Darwin Estivenson Martínez Muñoz

SENA
Colombia
2024

Contenido

Objetivos	3
Herramientas de versionamiento	4
Entorno de desarrollo integrado (IDE)	7
Lenguajes de programación para desarrollo web	8
Comunicación y Colaboración	8
Bases de Datos	9
Flujo de trabajo recomendado	9
Conclusión	11
Bibliografía	12

Introducción

El presente texto tiene como objetivo brindar una corta descripción del proyecto de software a desarrollar, mediante la gestión y colaboración del equipo de trabajo, del mismo modo se podrán observar las características básicas y mínimas del sistema de información a desarrollar.

En el desarrollo de software, la gestión eficaz de las versiones es fundamental para mantener la integridad, la colaboración y la evolución de los proyectos. Esto se logra mediante el uso de sistemas de control de versiones (VCS), que permiten a los equipos registrar, rastrear y gestionar los cambios en el código fuente y otros archivos. Dentro del ámbito del control de versiones, existen diferentes enfoques y conceptos importantes que los desarrolladores deben entender para optimizar su flujo de trabajo. Dos de estos conceptos clave son el versionamiento local y el versionamiento remoto, que abordan cómo se gestionan y almacenan las versiones de un proyecto. En esta breve exploración, examinaremos en detalle tanto el versionamiento local como el remoto, destacando sus diferencias, ventajas y aplicaciones en el contexto del desarrollo de software.

Tratare de proporcionar una descripción detallada de las actividades que se realizarán para poner en marcha la aplicación, igualmente los recursos económicos necesarios y los plazos para desarrollar el proyecto.

Objetivos

Objetivo General:

Desarrollar un sistema de información mediante la utilización de herramientas de versionado.

Objetivo Específico:

Elegir una herramienta y/o tecnología de versionado a utilizar en el desarrollo del sistema de información.

Garantizar un diseño sencillo y amigable con el usuario final.

Integrar adecuadamente eficiencia y versiones de software.

Herramientas de versionamiento

Una herramienta de versionamiento, también conocida como sistema de control de versiones (VCS por sus siglas en inglés), es un software diseñado para gestionar y controlar cambios en el código fuente y otros archivos de un proyecto de desarrollo de software. Su función principal es mantener un registro histórico de todas las modificaciones realizadas a lo largo del tiempo, permitiendo a los desarrolladores rastrear, revertir y fusionar cambios de manera efectiva.

Estas herramientas son fundamentales para facilitar la colaboración entre miembros de un equipo de desarrollo, ya que permiten trabajar en paralelo en diferentes partes del código sin interferir unos con otros. Además, proporcionan mecanismos para resolver conflictos que puedan surgir cuando dos o más personas modifican el mismo archivo simultáneamente.

Las herramientas de versionamiento se utilizan en una amplia variedad de entornos de desarrollo de software, desde proyectos pequeños con un solo desarrollador hasta grandes equipos distribuidos en todo el mundo. Permiten mantener la integridad y la coherencia del código base a lo largo del tiempo, lo que facilita su mantenimiento y evolución.

Entre las características comunes de las herramientas de versionamiento se incluyen la capacidad de crear ramas (branching) para trabajar en nuevas características o correcciones de errores sin afectar el código principal, la fusión (merging) de ramas para incorporar cambios de vuelta al tronco principal, y el etiquetado (tagging) de versiones para marcar hitos importantes en el desarrollo del proyecto.

Mi proyecto se basa en la creación de una página o recurso web, donde el propietario o usuario del sistema de información, podrá manejar todo lo relacionado con reporte de inventarios, pedidos, reporte de pagos, solicitudes de personalización de prendas de vestir, además el público en general podrá realizar la solicitud de compra o personalización de una prenda de vestir sin salir de la comodidad de su hogar.

Gestión de Usuarios:

Permite a los usuarios registrarse, iniciar sesión y administrar su perfil.

Gestión de Reportes:

Permite al usuario administrador generar reportes de ventas, inventario, solicitud de pedidos, envío de pedidos, etc

Interfaz de Usuario Intuitiva: El software cuenta con una interfaz de usuario amigable que facilita la navegación y la realización de operaciones, según las necesidades del cliente o usuario.

Beneficios de usar un sistema de control de versiones:

El uso de un sistema de control de versiones (VCS) ofrece una serie de beneficios significativos para los equipos de desarrollo de software y otros entornos donde se gestionan archivos y colaboran en proyectos. Aquí están algunos de los principales beneficios:

Registro Histórico: Un VCS registra todos los cambios realizados en el código fuente y otros archivos a lo largo del tiempo. Esto proporciona un historial completo de cómo ha evolucionado el proyecto, lo que facilita la comprensión de los cambios y la identificación de cuándo y por qué se realizaron modificaciones específicas.

Control de Versiones: Los VCS permiten mantener múltiples versiones del código base y otros archivos. Esto significa que los equipos pueden revertir a versiones anteriores si es necesario, lo que proporciona una forma de deshacer cambios no deseados o corregir errores introducidos en el código.

Colaboración Efectiva: Los VCS facilitan la colaboración entre miembros del equipo al permitir que múltiples personas trabajen en el mismo proyecto de forma simultánea. Los cambios realizados por diferentes personas se pueden fusionar de manera segura, lo que reduce los conflictos y garantiza que todos estén trabajando con la última versión del código.

Gestión de Ramificaciones (Branching): Los VCS permiten crear ramas independientes del código principal para trabajar en nuevas características o experimentos sin afectar el código base. Esto facilita la implementación de nuevas funcionalidades de forma aislada y la gestión de versiones paralelas del proyecto.

Seguimiento de Cambios: Los VCS proporcionan visibilidad sobre quién realizó qué cambios y cuándo. Esto facilita la revisión de los cambios realizados por diferentes miembros del equipo, la identificación de responsabilidades y la colaboración en la solución de problemas.

Gestión de Conflictos: Cuando dos o más personas modifican el mismo archivo simultáneamente, los VCS ofrecen herramientas para resolver conflictos de manera efectiva. Esto ayuda a garantizar que los cambios se incorporen de manera adecuada y se eviten pérdidas de datos o inconsistencias en el código.

Experimentación Segura: Los VCS permiten a los desarrolladores experimentar con nuevas ideas sin temor a dañar el código base. Al trabajar en ramas separadas, los desarrolladores pueden probar cambios sin afectar el funcionamiento estable del proyecto principal.

- **Versionamiento Remoto:**

En el versionamiento local, todas las operaciones de control de versiones se realizan en el sistema de archivos local de un desarrollador, es decir, en su propia máquina o computadora. Cuando un desarrollador trabaja en un proyecto utilizando un VCS local, todas las modificaciones, confirmaciones (commits), creación de ramas (branching), fusiones (merging) y otras operaciones de control de versiones se realizan sin conexión a un servidor remoto.

Los VCS locales, como Git o Mercurial, almacenan un repositorio completo en el sistema de archivos del desarrollador, lo que significa que tienen acceso a todo el historial de versiones y pueden realizar operaciones de control de versiones incluso cuando no están conectados a Internet. Sin embargo, los cambios no se comparten automáticamente con otros miembros del equipo hasta que se sincronizan con un repositorio remoto.

- **Versionamiento Remoto:**

En el versionamiento remoto, el repositorio de un proyecto se almacena en un servidor remoto, al que pueden acceder múltiples desarrolladores desde diferentes ubicaciones. Las operaciones de control de versiones, como confirmaciones, creación de ramas, fusiones y actualizaciones, se realizan en el servidor remoto en lugar del sistema de archivos local de un desarrollador.

Los VCS remotos, como GitHub, GitLab o Bitbucket, actúan como el punto central de colaboración para el equipo de desarrollo. Los desarrolladores pueden clonar un repositorio remoto en sus máquinas locales para trabajar en el proyecto y luego enviar sus cambios al servidor remoto para que otros miembros del equipo los revisen y fusionen.

El versionamiento remoto facilita la colaboración entre desarrolladores distribuidos geográficamente, ya que proporciona un único lugar centralizado donde se almacena y gestiona el código fuente del proyecto. Además, permite un seguimiento transparente de quién hizo qué cambios y cuándo se realizaron, lo que facilita la colaboración y la resolución de conflictos.

Software a desarrollar

Para el sistema de información a desarrollar se utilizarán las siguientes herramientas:

Entorno de desarrollo integrado (IDE)

- **Sublime Text 4**

Plataforma: Windows

Lenguajes: Soporte para múltiples lenguajes a través de extensiones

Características: Editor ligero, extensible y con gran cantidad de extensiones disponibles.

Control de Versiones

- **Git**

Plataforma: Multiplataforma

Características: Sistema de control de versiones distribuido

- **GitHub / GitLab / Bitbucket**

Plataforma: Basado en la web

Características: Plataformas de alojamiento de código que ofrecen control de versiones, seguimiento de problemas, integración continua, entre otros.

Lenguajes de programación para desarrollo web

- **HTML/CSS**

HTML (HyperText Markup Language)

CSS (Cascading Style Sheets)

- **JavaScript**

- **Laravel (PHP)**

Gestión de Proyectos y Tareas

- **Trello**

Plataforma: Basado en la web

Herramienta de gestión de proyectos basada en tarjetas y listas.

Comunicación y Colaboración

- **Microsoft Teams**

- **Zoom**

Bases de Datos

- **MySQL / PostgreSQL**

Conclusión

La integración del sistema de control de versiones junto con plataformas de alojamiento de código como GitHub o GitLab, herramientas de integración continua como GitLab, y plataformas de contenedores, proporciona una base sólida y eficiente para el desarrollo de software. Estas herramientas permiten una gestión de versiones efectiva, una colaboración y revisión de código en tiempo real, una automatización del proceso de desarrollo y despliegue, y una consistencia y reproducibilidad de los entornos de desarrollo y producción, lo que resulta en un proceso de desarrollo ágil, colaborativo y de alta calidad. La importancia del trabajo colaborativo y el desarrollo de ideas en conjunto es clave para el éxito de cualquier desarrollo de software. De aquí la importancia de utilizar las herramientas de versionamiento, control y gestión.