



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



Enigma Engineers

Solicitante: I.T.S. – Instituto Tecnológico Superior Arias - Balparda

Nombre de Fantasía del Proyecto: Enigma Engineers

Grupo de Clase: 3°BR

Turno: Nocturno

Materia: Programación Web

Nombre de los Integrantes del Grupo:

-Maximiliano Dominguez

-Lucio Schiavoni

-Mathias Labora

-Leandro Hernandez

Fecha de entrega: 24/07/2023



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



Objetivo:

Alcance:

Gestión de versiones:

¿Qué es una versión?

Una versión se refiere a una iteración o una variante específica de un programa, aplicación, sistema operativo o cualquier otro tipo de software.

Estas suelen estar identificadas por un número o conjunto de números, letras u otros identificadores que se diferencian según su ubicación en la secuencia.

Cada vez que se realiza una modificación en el software, ya sea para corregir errores, agregar nuevas características o mejorar su rendimiento, se puede lanzar una nueva versión del mismo. Tal como mencionamos estas versiones suelen estar mencionadas de manera secuencial, como la versión 1.0, 2.0, 3.0 y así sucesivamente, también podemos ver en algunos casos combinaciones alfanuméricas por ejemplo 1.2b, 2.1a, 3.5 rc (candidata a ser la versión final).

Es importante gestionar correctamente las versiones ya que permiten a los desarrolladores y usuarios distinguir entre los diferentes cambios que hayan influido en el programa, ya que cada versión puede tener cambios significativos en términos de características, funcionalidades, interfaz de usuario, compatibilidad con otros sistemas, rendimiento, seguridad y estabilidad.

¿Cómo se gestionan las versiones?

La mejor manera para gestionar versiones es utilizando herramientas y prácticas específicas para controlar y administrar las diferentes versiones de un programa.

Algunas formas de gestionar versiones son:

- 1) Control de versiones centralizado
- 2) Control de versiones distribuido (En nuestro caso nos enfocaremos en este punto)
- 3) Ramas (branches)
- 4) Etiquetas (tags)
- 5) Herramientas de gestión de versiones

2) En un control de versiones distribuido cada desarrollador tiene una copia completa del repositorio, esto permite más autonomía y flexibilidad. Los sistemas de control de versiones distribuidos como Git o Mercurial, permiten a los desarrolladores realizar



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



cambios y crear ramas (branches) locales para testear y luego mergearlos (fusionarlos) los cambios en el repositorio principal.

¿En qué momento se cambian los valores de las versiones?

Utilizaremos como ejemplo la versión “v2.1.4” donde el “2” representa la versión principal, lo que indica que hubo cambios significativos en comparación con la versión anterior (entiéndase como versión anterior “1”).

El “1” define la versión secundaria, lo que indica que se agregaron nuevas características o mejoras desde la versión anterior, y por último el número “4” hace referencia a las revisiones o parches que se hayan implementado para solucionar errores o problemas específicos.

Es importante tener en cuenta que el significado exacto de los números de versión puede variar según la convención utilizada por el desarrollador o la empresa responsable del software. Algunas organizaciones pueden tener sus propias reglas y pautas para asignar números de versión, por lo que es recomendable consultar la documentación o las políticas específicas del proyecto en cuestión para comprender completamente su sistema de versiones.