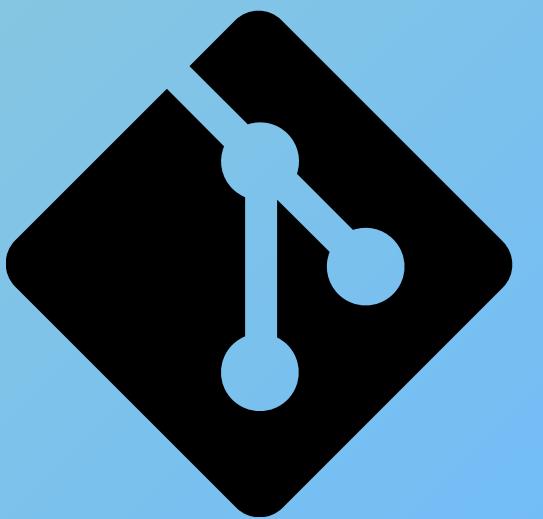


# Git

## Clase 03



# **<índice>**

## **Git: Instalación y configuración**

**¿Qué es Git?**

---

**¿Qué es Github?**

---

**Mi primer contacto con Git**

---

**Chuleta PDF**

---

**Instalación de Git**

---

**Configuración de Git....**

---

# ¿Qué es Git?

# ¿Qué es Git?

- Sistema de control de versiones
  - Centralizados => Subversion
  - Distribuidos => Git
- Es open source
- Gratis
- Escalable
- Rápido

# ¿Qué es Git?

- Base de datos local
- ¿Qué, Quién y Cuándo se modificó un archivo?
- Movernos en la línea de tiempo del proyecto
- Colaboración entre varios desarrolladores

# ¿Qué es Git?

Git es un **must** en desarrollo

# ¿Qué es Git?

## **Control de Versiones Eficiente**

Git facilita el seguimiento de cada cambio hecho en el código fuente, permitiendo a los desarrolladores revisar la historia del proyecto y revertir a versiones anteriores si es necesario.

# ¿Qué es Git?

## Trabajo en Paralelo

Permite a múltiples desarrolladores trabajar simultáneamente en diferentes ramas del mismo proyecto, sin interferir con el trabajo de los demás, lo que aumenta la eficiencia y reduce los conflictos en el código.

# ¿Qué es Git?

## Descentralización

Cada desarrollador trabaja con una copia local del repositorio, lo que significa que pueden seguir trabajando incluso sin conexión a internet. Los cambios pueden ser sincronizados con el repositorio remoto una vez que se restablezca la conexión.

# ¿Qué es Git?

## Rápido y ligero

Git es conocido por su rendimiento. Las operaciones son rápidas y eficientes, incluso en proyectos grandes, ya que Git maneja bien grandes volúmenes de archivos y cambios.

# ¿Qué es Git?

## Seguridad

Utiliza criptografía de hash SHA-1 para asegurar la integridad del código fuente.

Cada archivo y commit es verificado y seguro, lo que reduce las posibilidades de corrupción del código o interferencias externas.

# ¿Qué es Git?

## Ramas

Git facilita la gestión de ramas (branches), permitiendo a los desarrolladores experimentar con nuevas funciones o cambios sin afectar el proyecto principal. La fusión (merging) de estos cambios una vez completados es también sencilla y eficiente.

# ¿Qué es Git?

## **Independencia de la plataforma**

Git es compatible con la mayoría de los sistemas operativos y plataformas, lo que permite a los desarrolladores usarlo en cualquier entorno de desarrollo.

# ¿Qué es Git?

## Interfaz con Sistemas de Integración

### Continua (CI)

Git se integra bien con sistemas de CI/CD, lo que permite automatizar las pruebas y el despliegue del software, asegurando que solo el código probado y aprobado sea desplegado.

# ¿Qué es Github?

# ¿Qué es Github?

La “red social” y el CV de los Devs

# ¿Qué es Github?

La “red social” y el CV de los Devs

# ¿Qué es Github?

## Colaboración facilitada

GitHub hace más fácil para los equipos colaborar en proyectos, permitiendo a los usuarios contribuir al mismo código, revisarlo y discutir cambios antes de integrarlos al proyecto principal.

# ¿Qué es Github?

## Control de versiones

Permite a los desarrolladores realizar un seguimiento de los cambios en el código a lo largo del tiempo, revertir a versiones anteriores si es necesario y gestionar diferentes versiones de un proyecto simultáneamente.

# ¿Qué es Github?

## Integración Continua (CI/CD)

GitHub puede integrarse con varias herramientas de CI/CD, automatizando el proceso de prueba y despliegue de software, lo que mejora la eficiencia y la calidad del desarrollo.

# ¿Qué es Github?

## Documentación

Facilita la creación de documentación junto al código, lo que mejora la comprensión del proyecto y facilita la integración de nuevos colaboradores.

# ¿Qué es Github?

## Gestión de proyectos

Con herramientas como tableros de proyectos, listas de tareas e issues, los equipos pueden organizar el trabajo, priorizar tareas y seguir el progreso del desarrollo dentro de la misma plataforma.

# ¿Qué es Github?

## Comunidad y visibilidad

Ser parte de GitHub puede aumentar la visibilidad de un proyecto y del desarrollador, permitiendo la colaboración no solo dentro de equipos cerrados sino con una amplia comunidad de desarrolladores, lo que puede conducir a mejoras en el código y nuevas oportunidades.

# ¿Qué es Github?

## Seguridad

Ofrece herramientas para identificar y corregir vulnerabilidades de seguridad dentro del código, además de permitir la gestión de permisos para controlar quién puede acceder y modificar el proyecto.

# ¿Qué es Github?

## Aprendizaje y recursos

GitHub alberga una enorme cantidad de proyectos de código abierto, haciéndolo un recurso invaluable para aprender nuevas tecnologías y prácticas de codificación observando y colaborando en proyectos reales.

# **Mi primer contacto con Git**

# Mi primer contacto con Git

Primer proyecto con Git

**Chuleta PDF**

# Chuleta PDF

Enlace

# Instalación de Git

# Instalación de Git

- Windows => Git Bash
- Mac
- Linux
- Terminal vs GUI

# Configuración de Git

# Configuración de Git

```
git config --global user.name "tunombre"
```

```
git config --global user.email "tuemail"
```

```
git config --global core.editor "code --wait"
```

Comprobamos con

```
git config --global -e
```

# Configuración de Git

```
git config --global user.name "tunombre"
```

```
git config --global user.email "tuemail"
```

```
git config --global core.editor "code --wait"
```

# **Ficheros de texto plano**

# Ficheros de texto plano

Principales diferencias

# Ficheros de texto plano

Abramos un fichero de texto y un fichero rtf  
y un fichero de word

# Ficheros de texto plano

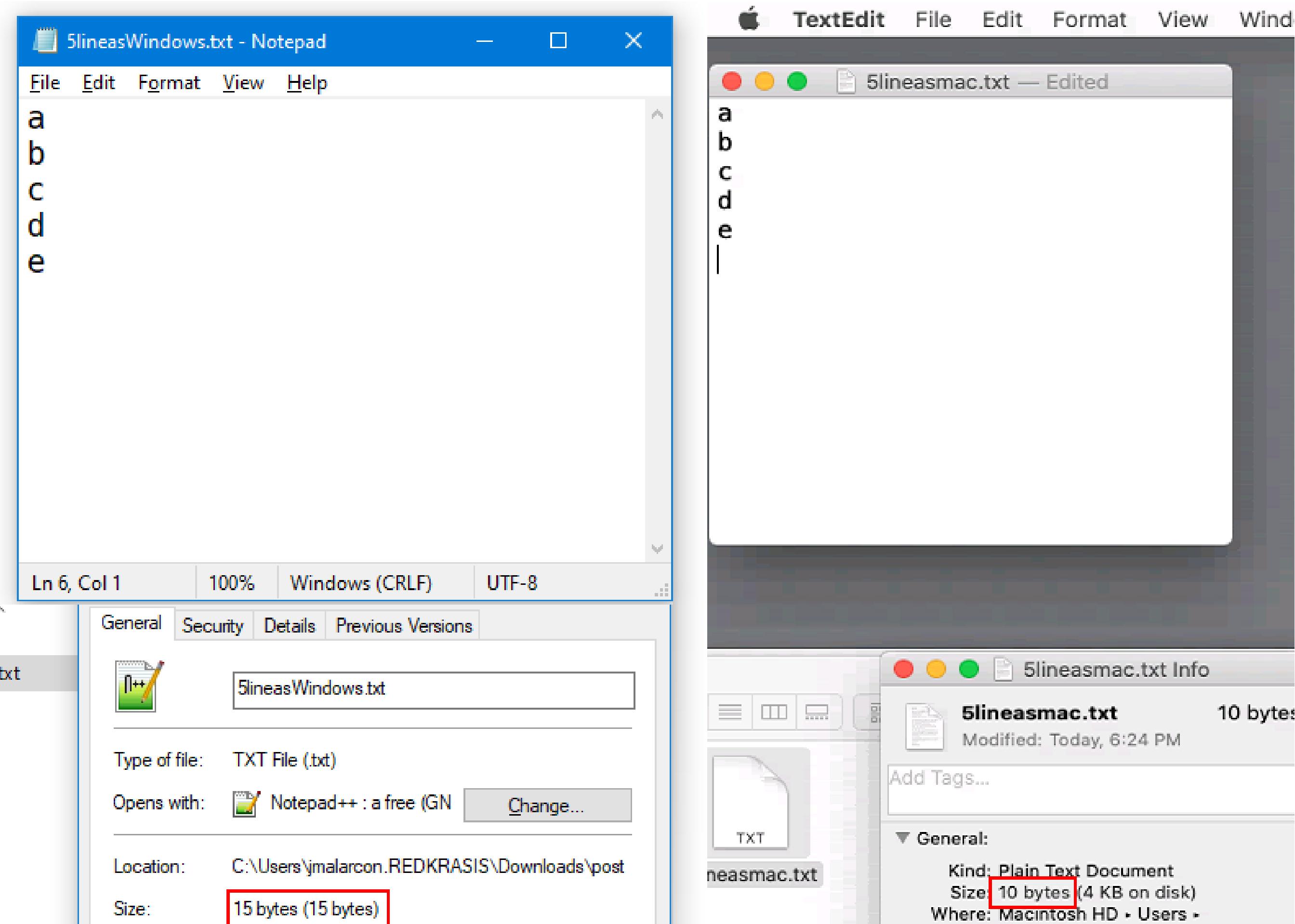
Git permite guardar ficheros binarios pero no controlará los cambios ni historia dentro del mismo, es importante entender esto

# **Saltos de línea Win vs Linux vs Mac**

# Saltos de línea

Principales diferencias

# Saltos de línea



# Saltos de línea

En macOS y Linux se usa el carácter LF para cambiar de línea, mientras que Windows utiliza dos caracteres, LF y CR, conocidos como CRLF, ocupando así 1 byte más por línea, lo que significa un 50% más en casos extremos.

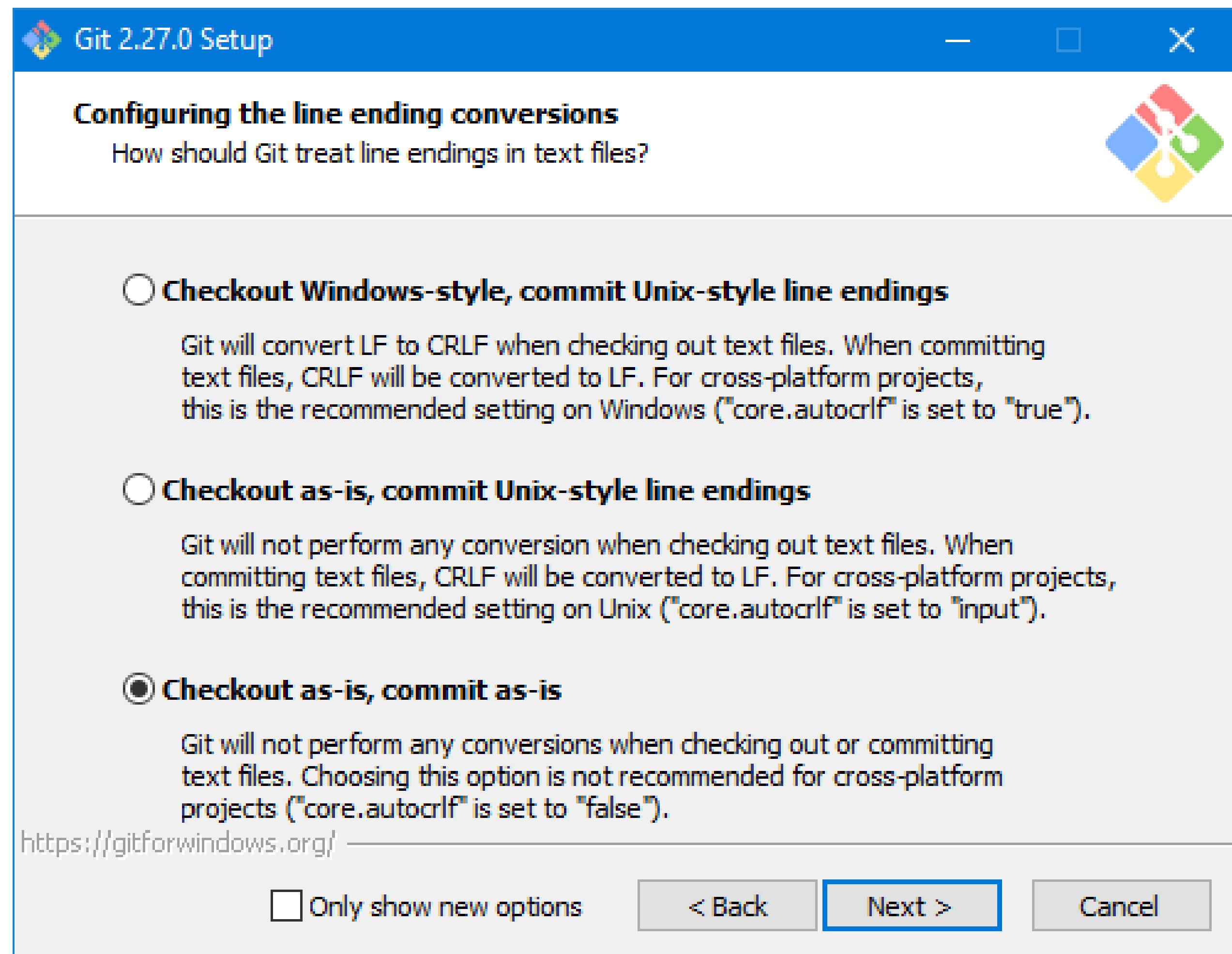
# Saltos de línea

The screenshot shows a text editor with three tabs open, each displaying a different line separator:

- 5lineasWindows.txt:** Contains lines 1 through 6, each ending with a **CRLF** (Carriage Return Line Feed) sequence.
- 5lineasmac.txt:** Contains lines 1 through 6, each ending with a **CR** (Carriage Return) character.
- 5lineasLinux.txt:** Contains lines 1 through 6, each ending with a **LF** (Line Feed) character.

The editor interface includes a menu bar with File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, TextFX, Plugins, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Find.

# Saltos de línea



# Saltos de línea

Al usar "**Checkout Windows-Style, commit Unix-style**",  
Git convierte los finales de línea a CRLF para trabajar  
cómodamente en Windows. Luego, al subir cambios, los  
vuelve a LF para mantener el estilo UNIX en el  
repositorio. ¡Es como tener lo mejor de ambos mundos!

# Saltos de línea

Con "**Checkout as is, commit Unix-style**", los archivos se descargan tal como están del repositorio, sin cambios. Pero cuando los enviamos de vuelta, Git cambia todos los finales de línea a LF, manteniendo el estilo UNIX en el repositorio. ¡Es sencillo y directo!

# Saltos de línea

**Checkout as-is, commit as-is:** con esta opción, Git no hace cambio alguno sobre los cambios de línea.

# Saltos de línea

La última opción es la peor de todas y la que no deberíamos elegir casi nunca. Si en el proyecto trabajan personas con diferentes sistemas o, si nosotros mismos trabajamos con más de un sistema operativo, acabaremos con el problema que describía antes, con varios tipos de cambios de línea mezclados.

# Saltos de línea

Lo que deberíamos hacer es, **marcar la opción 1 si trabajamos en Windows y la opción 2 si trabajamos en Linux o Mac.** De este modo tendríamos siempre el tipo de cambio de línea apropiado para nuestro sistema, y al mismo tiempo se guardaría de manera homogénea en los repositorios remotos.

# Saltos de línea

Documentación oficial

**Estamos listos...**

**<Despedida>**

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**