

# Fundamentos de Ingeniería Informática

## Modulo IV - Unidad I Introducción a Desarrollo de software

Teachers: Héctor Molina García

# ¿Qué es el desarrollo de software?

# Introducción al desarrollo de software

**El desarrollo de software** se refiere al proceso de diseño, creación, prueba y mantenimiento de sistemas o aplicaciones de software.



Implica una combinación de programación, resolución de problemas y creatividad para construir programas informáticos que cumplan con requisitos específicos o resuelvan problemas particulares.

**La mayor parte del ciclo de desarrollo de software está relacionado con el código fuente.**

- Codificación e implementación
- Pruebas
- Despliegue
- Mantenimiento y actualizaciones

**La mayor parte del ciclo de desarrollo de software está relacionado con el código fuente.**

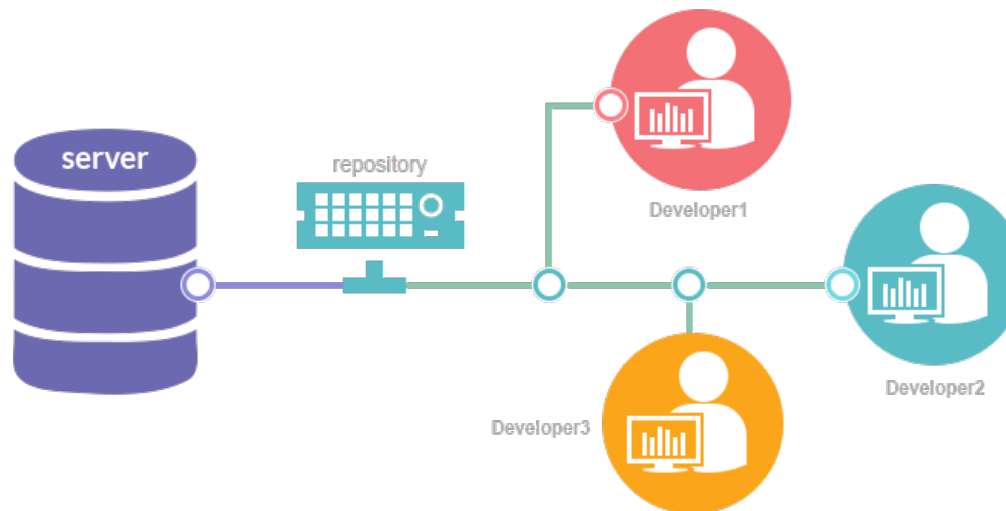
- Codificación e implementación
- Pruebas
- Despliegue
- Mantenimiento y actualizaciones

**El código fuente se maneja utilizando repositorios.**

# Introducción al desarrollo de software

**Un repositorio de código fuente**, también conocido como repositorio de control de versiones o repositorio de código, es una ubicación centralizada donde los desarrolladores de software almacenan y administran sus archivos de código fuente.

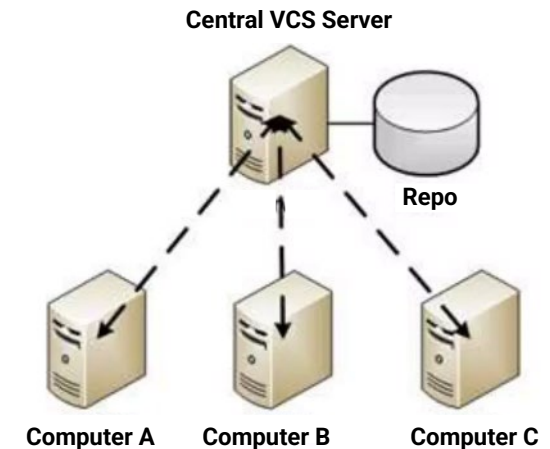
Es una herramienta fundamental utilizada en el desarrollo de software para realizar un seguimiento de los cambios, colaborar con otros desarrolladores y garantizar la integridad y disponibilidad del código fuente.



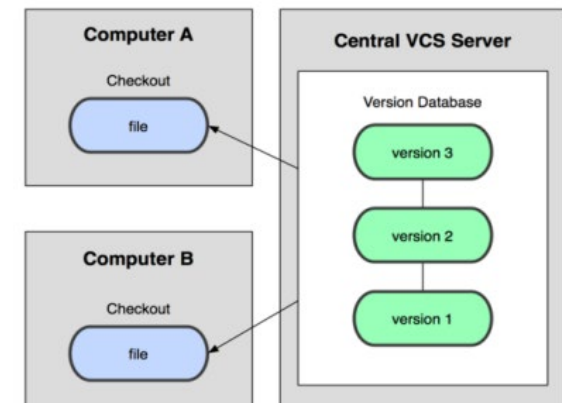
## Tipos de repositorios de versiones de control (repo): Centralizado

Un servidor VCS central (repo) contiene la "copia oficial" del código: el servidor mantiene el único historial de versiones del repositorio..

1. Las computadoras hacen "checkouts" de él a su copia local.
2. Los equipos realizan modificaciones locales no versionadas.
3. Las computadoras "registran" sus cambios en el servidor.



- Sistema de versiones concurrentes (CVS)
- Subversión (SVN)
- Team Foundation Server (TFS)



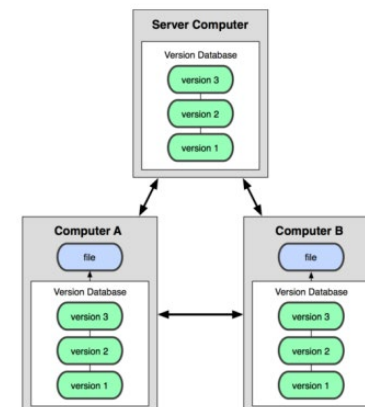
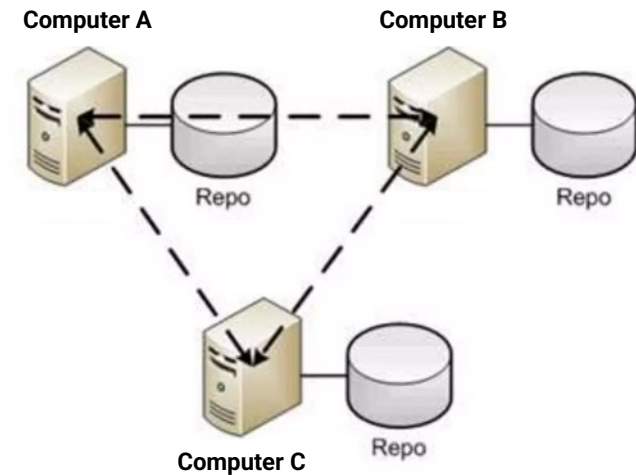
# Introducción al desarrollo de software

## Tipos de repositorios de versiones de control (repo): Distribuidos

No hay un servidor VCS central (repo). Los repositorios locales del equipo tienen una copia completa de todo lo que hay en el servidor remoto.

1. La mayoría de las operaciones se ejecutan localmente.
2. Las computadoras pueden "extraer" del servidor para actualizar sus repositorios locales.
3. Las computadoras pueden "enviar" los cambios de vuelta al servidor para actualizar sus cambios.

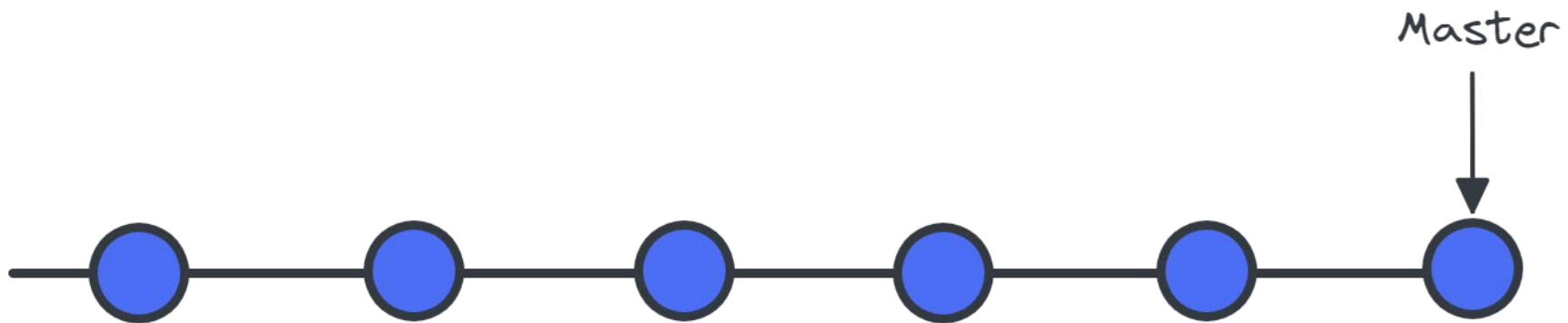
- Mercurial
- GIT





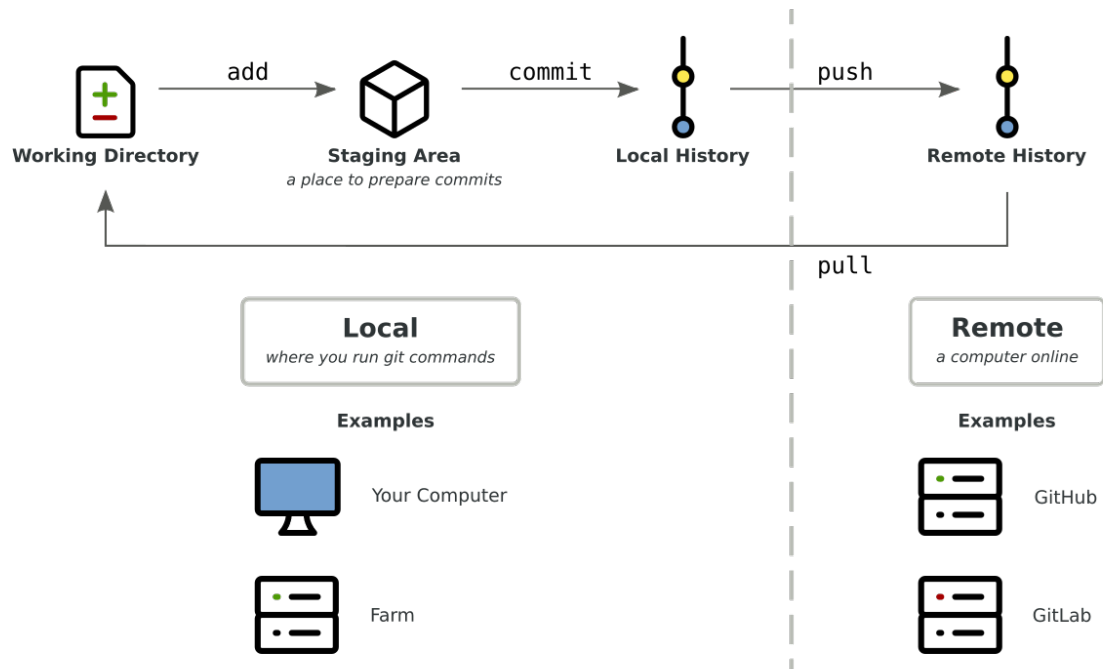
# GIT

**Git** es un sistema de control de versiones distribuido y de código abierto. Permite a los desarrolladores y científicos de datos realizar un seguimiento del código, combinar cambios y volver a versiones anteriores.



**Git se ha convertido en un estándar de la industria, ya que admite casi todos los entornos de desarrollo, herramientas de línea de comandos y sistemas operativos.**

## Entornos Git: locales y remotos

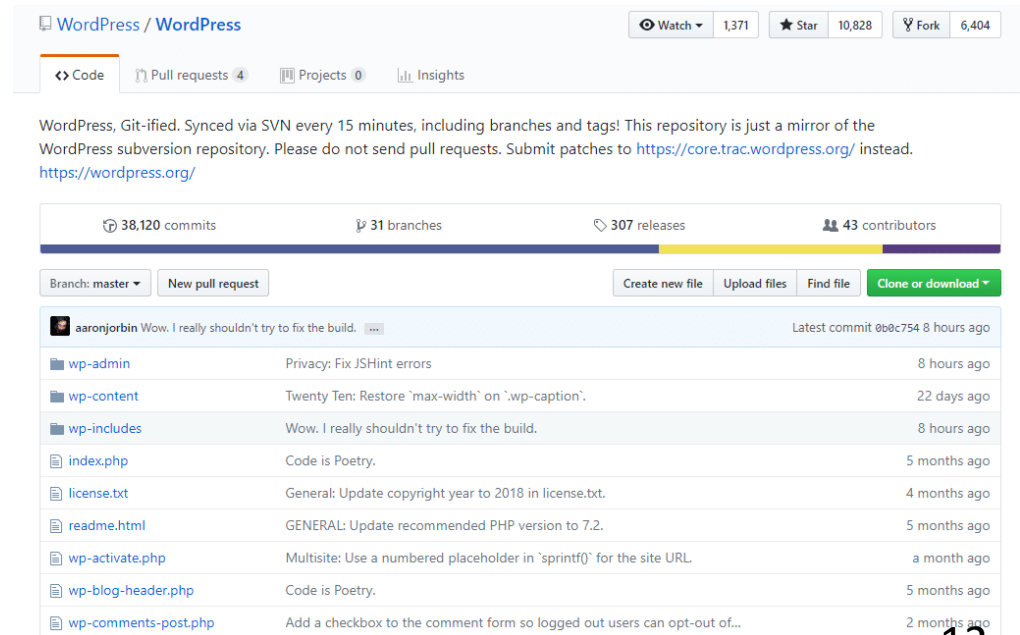


- Control de versiones distribuido.
- Los usuarios mantienen todo el código en las máquinas locales.
- Los usuarios pueden realizar cambios locales sin Internet.
- Pull and push sincroniza cambios locales y remotos.

## Repositorio de código

Un repositorio Git, a menudo denominado "repositorio", es una estructura de datos que contiene una colección de archivos, directorios y el historial de revisión de esos archivos. Sirve como una ubicación central para administrar y rastrear cambios en el código fuente de un proyecto o cualquier otro conjunto de archivos..

- Archivos y directorios
- Historial de confirmaciones
- Branches (Ramas)
- Conexión de repositorio remoto

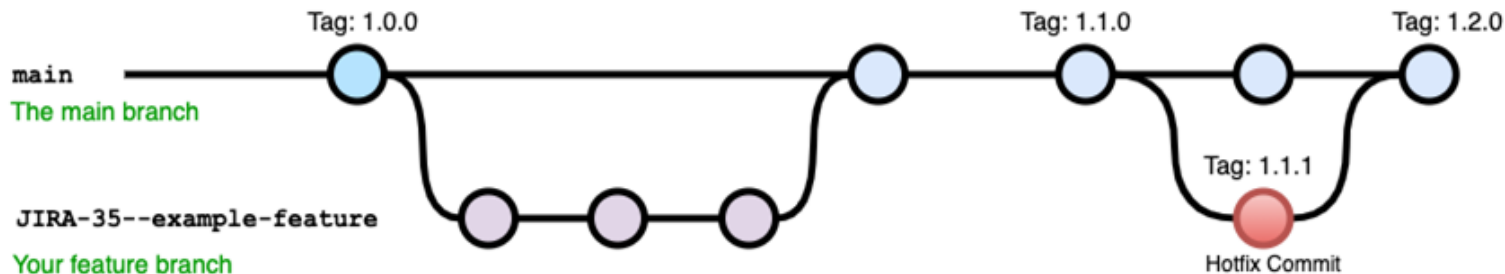


The screenshot shows the GitHub interface for the WordPress repository. At the top, it displays the repository name 'WordPress / WordPress' and statistics: 1,371 watches, 10,828 stars, and 6,404 forks. Below this, there are tabs for 'Code', 'Pull requests' (4), 'Projects' (0), and 'Insights'. A message states: 'WordPress, Git-ified. Synced via SVN every 15 minutes, including branches and tags! This repository is just a mirror of the WordPress subversion repository. Please do not send pull requests. Submit patches to <https://core.trac.wordpress.org/> instead. <https://wordpress.org/>'. Below the message, it shows 38,120 commits, 31 branches, 307 releases, and 43 contributors. A bar chart shows the commit history. At the bottom, there is a table of recent commits with columns for the commit message, the file changed, and the time since the commit.

Commit Message	File Changed	Time Since Commit
Wow. I really shouldn't try to fix the build.	wp-admin	8 hours ago
Privacy: Fix JSHint errors	wp-content	22 days ago
Twenty Ten: Restore "max-width" on ".wp-caption".	wp-includes	8 hours ago
Wow. I really shouldn't try to fix the build.	index.php	5 months ago
Code is Poetry.	license.txt	4 months ago
General: Update copyright year to 2018 in license.txt.	readme.html	5 months ago
GENERAL: Update recommended PHP version to 7.2.	wp-activate.php	a month ago
Multisite: Use a numbered placeholder in "sprintf()" for the site URL.	wp-blog-header.php	5 months ago
Code is Poetry.	wp-comments-post.php	2 months ago
Add a checkbox to the comment form so logged out users can opt-out of...		

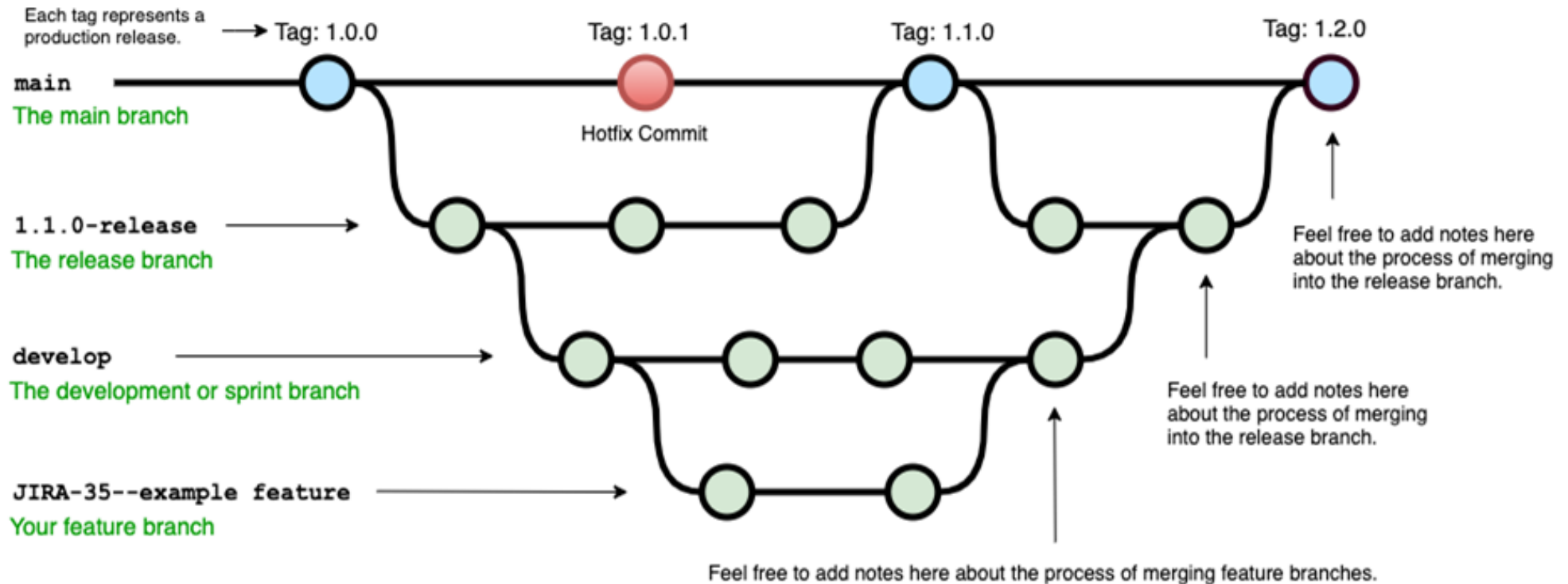
## Branch

En Git, un **branch** (rama) es un puntero móvil ligero que representa una línea de desarrollo independiente dentro de un repositorio.



Permite que varios desarrolladores trabajen en diferentes características, correcciones de errores o experimentos al mismo tiempo, manteniendo sus cambios aislados entre sí hasta que estén listos para fusionarse.

## Branch



La rama principal de un repositorio se denomina rama maestra o principal.

## Branch

En Git, una instantánea (snapshot) se refiere al estado de todos los archivos del proyecto en un momento determinado. Representa una imagen completa de todos los archivos, directorios y su contenido en un repositorio en una confirmación específica..

Cuando realiza una confirmación en Git, captura el estado actual del proyecto como una instantánea. Esta instantánea incluye todos los cambios realizados en los archivos, incluidas modificaciones, adiciones y eliminaciones.

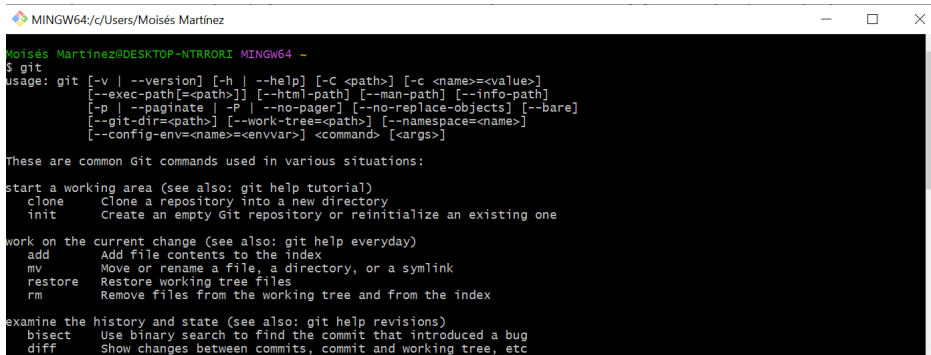
# GIT: Comandos básicos



# GIT: Comandos básicos

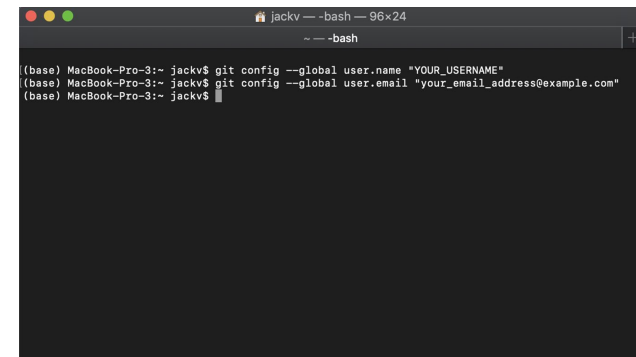
## Conceptos clave: Ejecutar git sobre el cmd

El método principal para usar Git es a través de la línea de comandos, que incluye Git Bash para Windows y la terminal para Mac y Linux.

A screenshot of a Git Bash terminal window in Windows. The title bar shows the path 'MINGW64: c:/Users/Moisés Martínez'. The terminal displays the 'git' command help text, including options like '-v', '--version', '-h', '--help', and various flags for cloning, working, and examining history. It lists common Git commands such as 'clone', 'init', 'add', 'mv', 'restore', 'rm', 'bisect', and 'diff' with brief descriptions of their functions.

```
Moisés Martínez@DESKTOP-NTRRORI MINGW64 ~  
$ git  
usage: git [-v | --version] [-h | --help] [-c <path>] [-c <name>=<value>]  
        [--exec-path<path>] [--html-path] [--man-path] [--info-path]  
        [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]  
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]  
        [--config-env=<name>=<envvar>] <command> [<args>]  
  
These are common Git commands used in various situations:  
  
start a working area (see also: git help tutorial)  
  clone Clone a repository into a new directory  
  init Create an empty Git repository or reinitialize an existing one  
  
work on the current change (see also: git help everyday)  
  add Add file contents to the index  
  mv Move or rename a file, a directory, or a symlink  
  restore Restore working tree files  
  rm Remove files from the working tree and from the index  
  
examine the history and state (see also: git help revisions)  
  bisect Use binary search to find the commit that introduced a bug  
  diff Show changes between commits, commit and working tree, etc
```

Git Bash en Windows

A screenshot of a Mac Terminal window. The title bar shows 'jackv -- -bash -- 96x24'. The terminal shows the prompt '(base) MacBook-Pro-3:~ jackv\$' followed by two 'git config' commands being executed to set global user.name and user.email.

```
jackv -- -bash -- 96x24  
~ -- -bash  
  
(base) MacBook-Pro-3:~ jackv$ git config --global user.name "YOUR_USERNAME"  
(base) MacBook-Pro-3:~ jackv$ git config --global user.email "your_email_address@example.com"  
(base) MacBook-Pro-3:~ jackv$
```

Terminal en Mac

# GIT: Comandos básicos

## Conceptos clave: Ejecutar git sobre el cmd

Hay algunos comandos básicos para usar la línea de comandos indispensables.

Command	Meaning
cd "folder name"	El comando cd, abreviatura de "cambiar directorio", permite a los usuarios cambiar de su directorio de trabajo actual a un directorio específico
ls -la	El comando ls permite a los usuarios listar archivos y directorios en el directorio de trabajo actual. Proporciona una vista concisa del contenido, mostrando los nombres de los archivos y carpetas. Agregar opciones como -l proporciona información detallada, mientras que -a muestra archivos ocultos.
mkdir "folder name"	El comando mkdir permite a los usuarios crear un nuevo directorio o carpeta. Al especificar un "nombre de carpeta" como argumento, genera instantáneamente el directorio especificado en el directorio de trabajo actual
rmdir "folder name"	El comando rmdir permite a los usuarios eliminar un directorio o carpeta vacíos. Elimina el directorio especificado si no contiene subdirectorios ni archivos.
cat "file name"	El comando cat permite a los usuarios concatenar y mostrar el contenido de uno o más archivos. Se puede emplear para ver todo el contenido de un archivo.

# GIT: Comandos básicos

## Conceptos clave: Clonar un repositorio (repo).

**git clone** es un comando de Git utilizado para crear una copia local de un repositorio remoto. Cuando clona un repositorio, recupera todos sus archivos, historial de confirmaciones y ramas en su máquina local

```
git clone https://github.com/your-username/your-repository.git
```

Una vez que haya creado un repositorio en GitHub, puede proceder a clonarlo, creando así una copia local en su máquina. Para iniciar este proceso, siga las [instrucciones](#) siguientes para crear su repositorio.

# GIT: Comandos básicos

## Conceptos clave: Crear un repositorio local (repo).

**git init** es un comando de Git que inicializa un nuevo repositorio de Git en un directorio del equipo local. Este comando configura la infraestructura Git necesaria, creando un repositorio vacío con una confirmación inicial.

```
git init
```

Después de ejecutar `git init`, verás un nuevo subdirectorio llamado `.git` creado dentro del directorio actual. Este directorio es donde Git almacena todos los archivos y metadatos necesarios para administrar el control de versiones para su proyecto.

# GIT: Comandos básicos

## Conceptos clave: añadir cambios/archivos al repositorio local.

**git add** es un comando de Git que se usa para agregar cambios o nuevos archivos al área de ensayo. El área de ensayo es una parte crítica del flujo de trabajo de Git, donde se seleccionan y preparan cambios o archivos específicos para incluirlos en la próxima confirmación.

```
git add <file_name_1> <file_name_2> ...
```

Después de ejecutar git add, los cambios o archivos especificados se agregan al área de ensayo, listos para ser confirmados.

**Git add se utiliza para agregar cambios a archivos existentes y agregar nuevos archivos.**

# GIT: Comandos básicos

## Conceptos clave: Confirmar cambios en el repositorio local.

**git commit** es un comando de Git utilizado para crear un nuevo commit en un repositorio de Git. Una confirmación representa una instantánea del estado actual de los archivos del proyecto y sirve como un registro permanente de los cambios realizados en el repositorio.

```
git commit -m "Added new features and fixed bugs"
```

Si bien es posible ejecutar git commit sin la opción -m, se considera una buena práctica incluir un mensaje de confirmación descriptivo para cada confirmación.

# GIT: Comandos básicos

## Conceptos clave: Push al repositorio remoto.

**git push** es un comando de Git utilizado para cargar confirmaciones locales en un repositorio remoto. Después de realizar confirmaciones locales en su repositorio Git, el comando git push le permite sincronizar esas confirmaciones con un repositorio remoto, como uno alojado en GitHub u otro servicio de alojamiento Git.

```
git push <remote> <branch>
```

- El <remote> se refiere al nombre del repositorio remoto donde desea insertar sus confirmaciones locales. Normalmente, este es el repositorio de origen, que se establece como el remoto predeterminado cuando se clona un repositorio.
- El <branch> especifica la rama que desea insertar. Git permite múltiples ramas en un repositorio, lo que permite el desarrollo paralelo o el aislamiento de características.

## Conceptos clave: Pull desde el repositorio remoto.

**git pull** es un comando de Git que se utiliza para recuperar y combinar cambios de un repositorio remoto en la rama actual. Cuando ejecuta `git pull`, Git recupera las confirmaciones y actualizaciones más recientes del repositorio remoto y las lleva a su repositorio local.

```
git pull <remote> <branch>
```

- El `<remote>` especifica el nombre del repositorio remoto del que desea extraer los cambios, normalmente denominado origen.
- El indica la rama `<branch>` del repositorio remoto que desea extraer los cambios en la rama actual.



## Conceptos clave: Crear una nueva rama

**git checkout -b** es un comando Git en Git que se utiliza para crear una nueva rama y cambiar a ella en un solo paso. Es un atajo conveniente que combina dos comandos Git separados: `git branch` y `git checkout`.

```
git checkout -b <new-branch-name>
```

El `<new-branch-name>` representa el nombre que desea dar a la nueva rama que está creando.

# GIT: Archivos

## Archivos especiales: README.md

Un archivo README.md es un archivo de uso común en proyectos de desarrollo de software, particularmente en proyectos de código abierto alojados en plataformas como GitHub. Está escrito en formato Markdown, que es un lenguaje de marcado ligero que permite un fácil formato y estructuración del texto.



El propósito de un archivo README.md es proporcionar información y documentación esencial sobre el proyecto.

## Archivos especiales: Licencia

En Git, un archivo de licencia es un archivo de texto que contiene los términos y condiciones legales bajo los cuales el proyecto de software se comparte con otros. Especifica los permisos, restricciones y obligaciones asociados con el uso, modificación, distribución y licencia del código fuente del proyecto. Las licencias más utilizadas son las licencias de código abierto, como:

- MIT License
- Apache License
- GNU General Public License (GPL)
- Creative Commons Licenses

El archivo de licencia es un componente importante de los proyectos de código abierto, ya que aclara cómo otros pueden usar, modificar y distribuir el código del proyecto al tiempo que garantiza el cumplimiento de los requisitos legales. Ayuda a definir el modelo de licencia del proyecto y proporciona protección legal tanto para el proyecto como para sus usuarios..

# GIT: Instalación

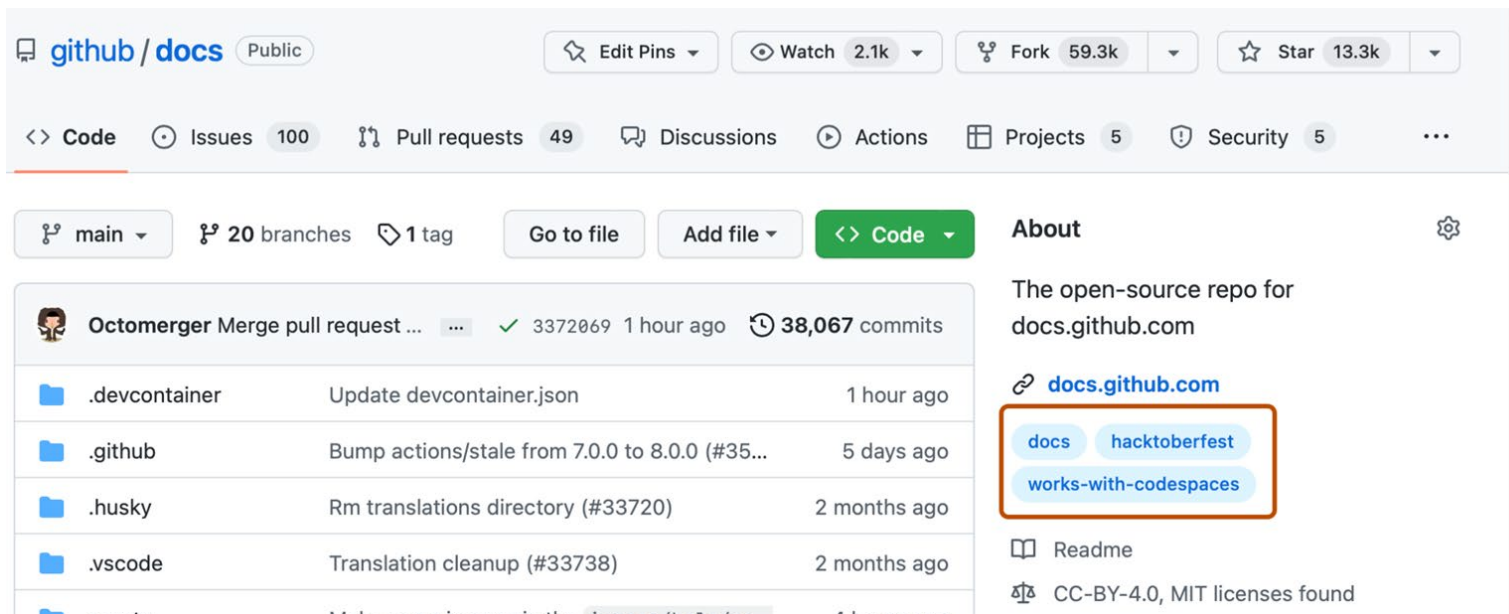
## Instala Git en tu portátil

- Linux (Ubuntu) → Command `sudo apt-get install git`
- Windows → <http://git-scm.com/download/win>
- Mac → <http://git-scm.com/download/mac>



# GIT: Instalación

## Crear una cuenta de github



The screenshot shows the GitHub repository page for 'github/docs'. The repository is public and has 2.1k watches, 59.3k forks, and 13.3k stars. The repository has 100 issues, 49 pull requests, 5 projects, and 5 security issues. The repository is currently on the 'main' branch, which has 20 branches and 1 tag. The repository is described as 'The open-source repo for docs.github.com'. The repository has 38,067 commits. The repository is currently being merged by 'Octomerger Merge pull request ...'. The repository has a README file. The repository is licensed under CC-BY-4.0 and MIT licenses. The repository is currently being merged by 'Octomerger Merge pull request ...'. The repository has a README file. The repository is licensed under CC-BY-4.0 and MIT licenses.

File	Commit Message	Time
.devcontainer	Update devcontainer.json	1 hour ago
.github	Bump actions/stale from 7.0.0 to 8.0.0 (#35...	5 days ago
.husky	Rm translations directory (#33720)	2 months ago
.vscode	Translation cleanup (#33738)	2 months ago
assets	Make some images in the /assets folder	4 hours ago



[Join to GH](#)

**GitHub es una plataforma basada en web que proporciona un servicio de alojamiento de repositorios para el control de versiones mediante Git.**

# GIT: Instalación. Conexión github-vscode

- Conocerlos conceptos básicos cmd.
  - `help//exit//cd "carpeta">//cd../dir//cls//mkdir carpeta//md carpeta//move archivo destino//mv`
- Instalar Git y crear cuenta en Github.
- Instalar VS Code. Conceptos básicos
  - `git versión//git help//`
  - `git init//`Iniciar un nuevo repositorio. Crea carpeta oculta.git, solo una vez por proyecto.
  - `git status -s//` archivos no registrados.
  - `git add ../`agregar archivos pendientes de cambio.
  - `git commit -m "comentario de los cambios"//`Etiquetar los cambios.



# GIT: Instalación. Conexión github-vscode

- Conexión en VS Code con Github (preparación)
  - `git config --global user.name "mi nombre"`// el de github
  - `git config --global user.email "micorreo@correo.es"`// el de github
  - `git config --global user.name`// comprobación
  - `git config --global user.email`// comprobación.
- Crear nuevo repositorio en Github.
  - `git remote add origin https://github.com/tuwebparafii.git`
  - `git Branch -m main`//rama principal
  - `git push -u origin main`//Incorporar credenciales en web.
- Para sincronizar:
  - `git add .`//añadir
  - `git commit -m "nuevo commit"`/etiquetar
  - `git push/`

# Fundamentos de Ingeniería Informática

## Módulo IV - Unidad II Herramientas de desarrollo

Teachers: Héctor Molina García

Version: 0.2



Universidad  
Francisco de Vitoria  
UFV Madrid

*Grado en Ingeniería Informática  
Escuela Politécnica Superior<sup>1</sup>*



# ¿Qué herramientas de desarrollo conoces?

# Herramientas de desarrollo

Una **herramienta de desarrollo**, también conocida como herramienta de desarrollo de software o herramienta de programación, se refiere a cualquier programa informático o aplicación que los desarrolladores utilizan para crear, depurar, mantener o dar soporte a otros programas y aplicaciones. Algunos tipos comunes de herramientas de desarrollo son:

- Entornos de Desarrollo Integrados (IDEs).
- Editores de texto.
- Sistemas de control de versiones (VCS) como Git.
- Depuradores.

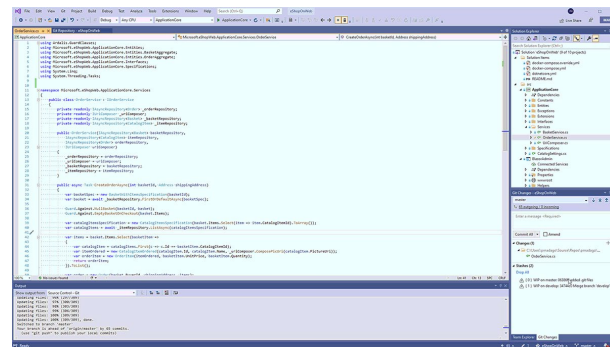
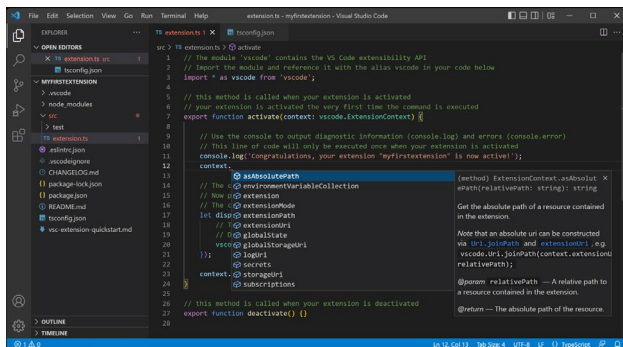




# ¿Qué es un IDE?

# Herramientas de desarrollo

Un entorno de desarrollo integrado (IDE) es un tipo de software de aplicación que proporciona un conjunto de herramientas y características para facilitar y agilizar el proceso de desarrollo de software.

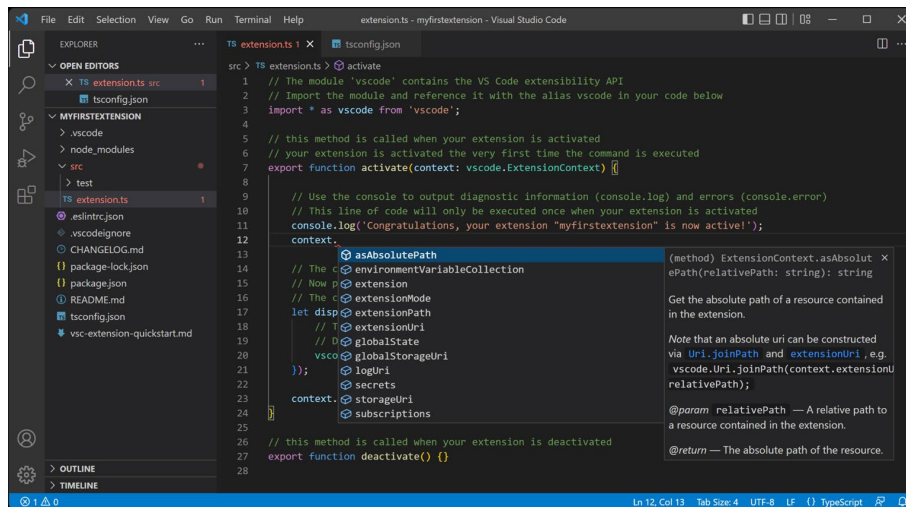


Los IDE están diseñados para ser un eje central donde los desarrolladores pueden escribir, editar, compilar, depurar y administrar su código en un entorno unificado.

# Herramientas de desarrollo

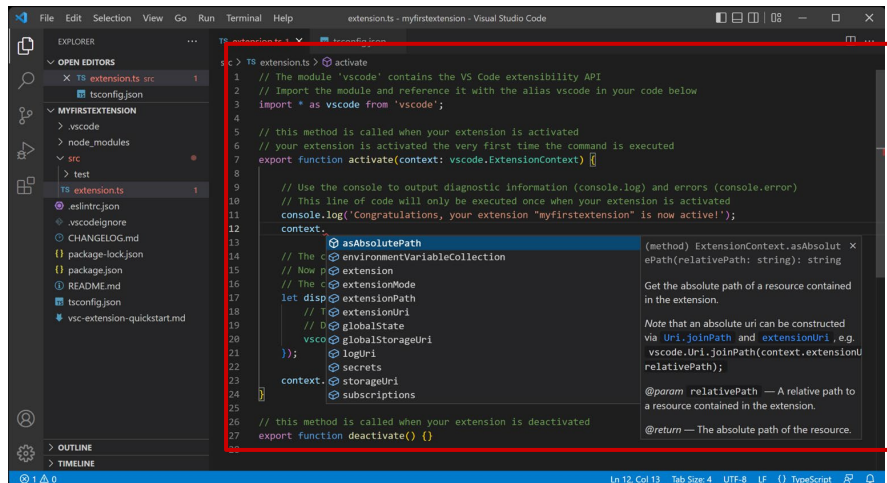
Las principales características que se encuentran comúnmente en un IDE son:

- **Editor de código fuente.**
- **Compilador/Intérprete.**
- **Depurador.**
- **Herramientas de automatización de compilaciones.**
- **Integración de control de versiones.**
- **Herramientas de gestión de proyectos.**



# Herramientas de desarrollo

El **editor de código fuente** es un editor de texto con funcionalidades avanzadas específicamente diseñadas para la programación. A menudo incluye características como el resaltado de sintaxis, la finalización de código y el plegado de código para mejorar la legibilidad y la facilidad de codificación.

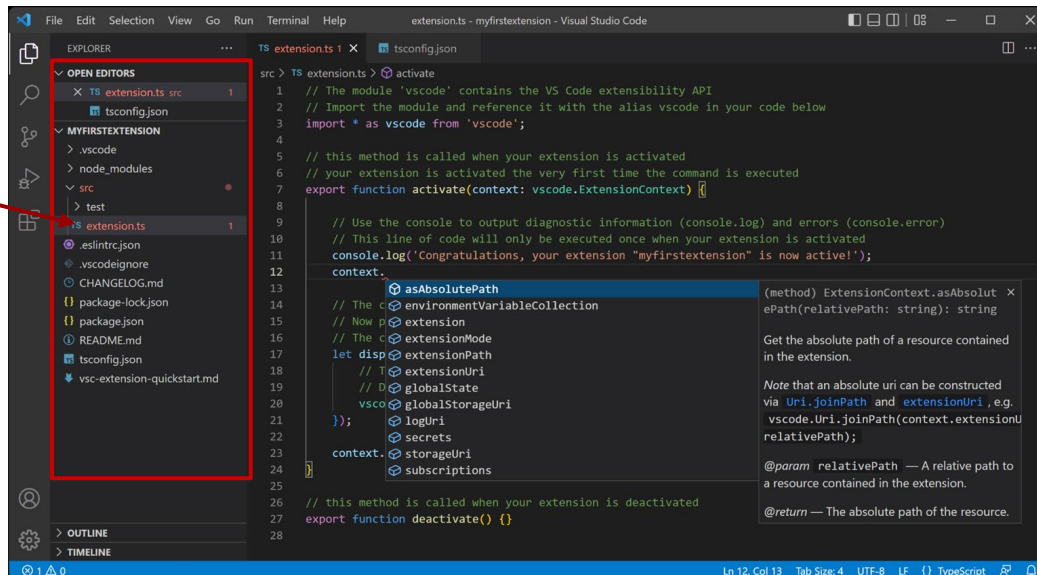




# Herramientas de desarrollo

Las herramientas de **gestión de proyectos** incluyen funciones para gestionar proyectos, organizar archivos y gestionar dependencias. Estas herramientas vinculan directorios almacenados en la memoria secundaria (como HDD) como proyectos, que abarcan varios tipos de archivos.

extension.ts is a  
typescript file





# Un proyecto de software

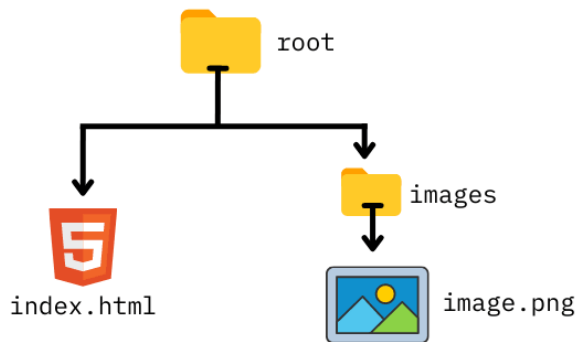
# Herramientas de desarrollo

Un proyecto de software consta de varios archivos, normalmente distribuidos en diferentes carpetas. En consecuencia, es crucial establecer rutas definidas para localizar estos diversos archivos:

- Una ruta de acceso absoluta denota la ubicación de un archivo o carpeta del directorio raíz del sistema de archivos.
- Una ruta relativa denota la ubicación de un archivo o carpeta en relación con el directorio actual (de trabajo). Por lo general, usamos rutas relativas en los proyectos por, y hay algunas reglas clave a seguir cuando las definimos:
  - / es la raíz de la unidad actual. Usamos c:/ en Windows OOS.
  - ./ es el directorio actual.
  - ../ es el padre del directorio actual.



# Herramientas de desarrollo



Este ejemplo ilustra una página web HTML fundamental (index.html) con una imagen (image.png).

La imagen se vincula mediante una ruta relativa.

```

```

**OR**

```

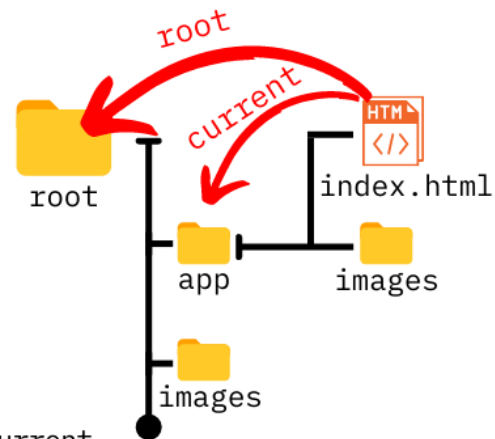
```

**Ruta relativa**

# Herramientas de desarrollo

En este ejemplo se describe la configuración de un disco duro con su directorio principal establecido como raíz.

- Una aplicación web se aloja dentro de la carpeta raíz.
- El disco duro contiene diferentes carpetas de imágenes: (1) una para imágenes generales; y (2) otro específicamente designado para aplicaciones web.



`./` - current.

`../` - one step back.

`/` - root.

`./images` - Find the folder in the current folder.

`../images` - Go one step back and find the folder.

`/images` - Find the folder at the root.

# Herramientas de desarrollo

El **Compilador/Intérprete** es un componente encargado de traducir el código fuente escrito por un programador a un formato que el ordenador pueda ejecutar. El rol específico depende del lenguaje de programación y de la naturaleza del código.

- Un compilador traduce todo el código fuente de un programa en código máquina o en código intermedio, todo a la vez. **La compilación suele ser un paso independiente antes de ejecutar el programa.**
- Un intérprete traduce el código fuente a código máquina o un código intermedio línea por línea a medida que se ejecuta el programa. **La interpretación se realiza en tiempo real durante la ejecución del programa y no hay ningún paso de compilación separado.**

En algunos casos, un idioma puede utilizar una combinación de compilación e interpretación. Por ejemplo, los programas Java se compilan inicialmente en un código de bytes intermedio y, a continuación, la máquina virtual Java (JVM) interpreta o compila este código de bytes en tiempo de ejecución.

# Herramientas de desarrollo

Algunos de los IDE de programación más populares:



# Herramientas de desarrollo

Algunos de los IDE de programación más populares:

- **Eclipse** es un IDE ampliamente utilizado para el desarrollo de Java, pero es compatible con muchos otros lenguajes a través de complementos.
- **Jupyter Notebook** es un IDE de código abierto que se usa para crear documentos de Jupyter que se pueden crear y compartir con códigos activos.
- **PyCharm** es un IDE diseñado específicamente para el desarrollo de Python, con características como finalización de código, depuración y soporte para marcos de desarrollo web.
- **Visual Studio Code**, a menudo abreviado como VS Code, es un IDE gratuito de código abierto desarrollado por Microsoft. Este es uno de los IDE más comunes utilizados por las aplicaciones web.



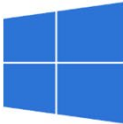



# Git y Visual Studio Code


# Herramientas de desarrollo

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

  
**↓ Windows**  
Windows 10, 11

  
**↓ .deb**  
Debian, Ubuntu  
**↓ .rpm**  
Red Hat, Fedora, SUSE

  
**↓ Mac**  
macOS 10.15 +

User Installer x64 Arm64

System Installer x64 Arm64

.zip x64 Arm64

CLI x64 Arm64

.deb x64 Arm32 Arm64

.rpm x64 Arm32 Arm64

.tar.gz x64 Arm32 Arm64

Snap Snap Store

CLI x64 Arm32 Arm64

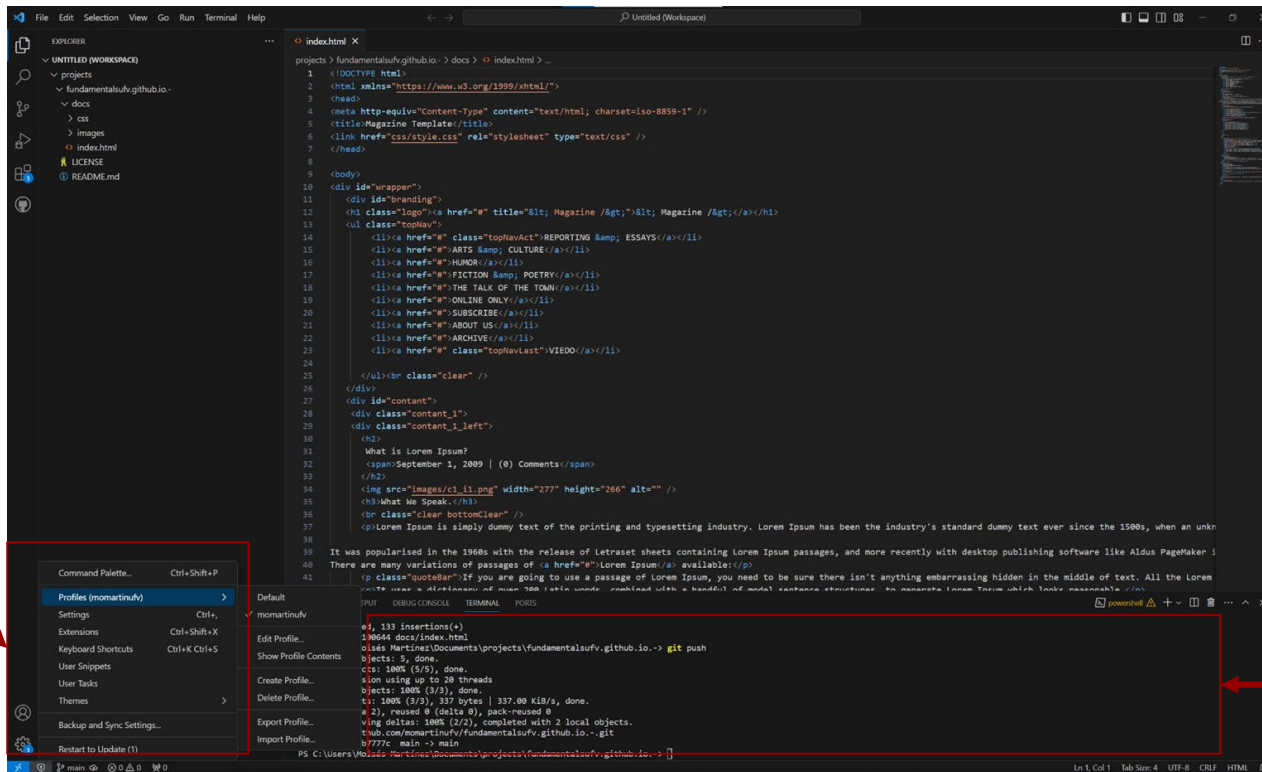
.zip Intel chip Apple silicon Universal

CLI Intel chip Apple silicon

<https://code.visualstudio.com/download>

# Herramientas de desarrollo

Ajuste para configurar  
el perfil



Terminal

# Herramientas de desarrollo

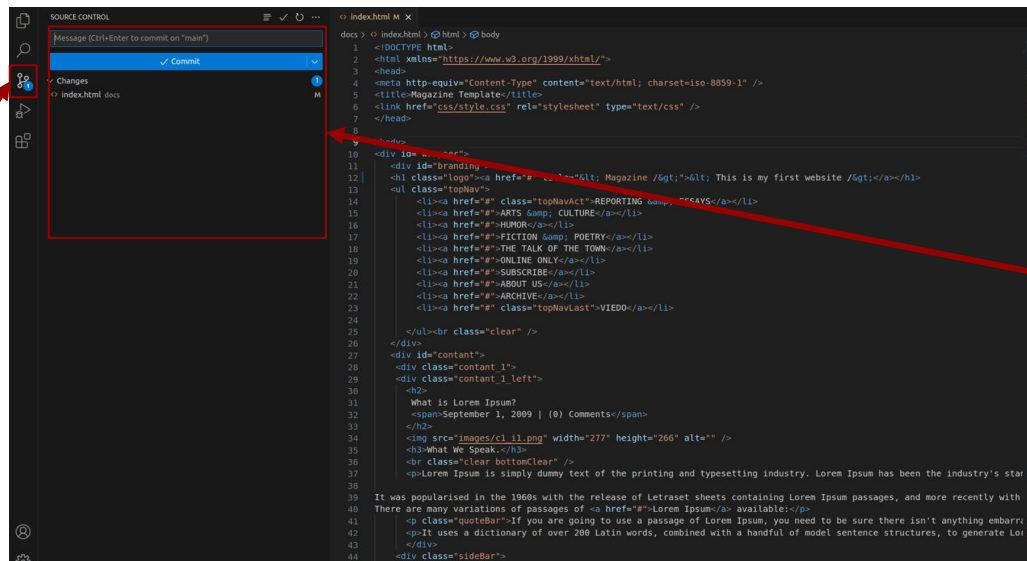
La manera más fácil de conectar tu cuenta de GitHub con Visual Studio Code es clonar un repositorio desde el terminal.

```
git clone https://github.com/your-username/your-repository.git
```

Esta acción abrirá una ventana en su navegador que sincroniza su cuenta con el IDE. A partir de este momento, podrá tirar y empujar los repositorios asociados con su usuario.

# Herramientas de desarrollo

Los IDE suelen ofrecer herramientas visuales para controlar el código fuente mediante git. El código de Visual Studio tiene una sección especial a la izquierda para controlar los cambios y ejecutar las inserciones.

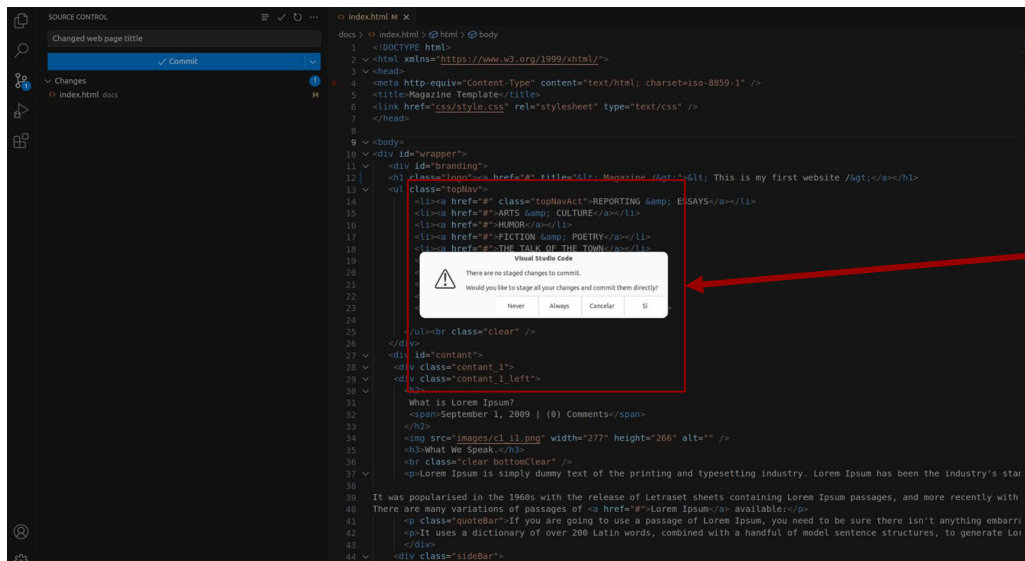


Esta sección le permite controlar los cambios.

Puede elegir los archivos que desea agregar a una confirmación y enviarlos al repositorio con unos pocos clics.

# Herramientas de desarrollo

Los IDE suelen ofrecer herramientas visuales para controlar el código fuente mediante git. El código de Visual Studio tiene una sección especial a la izquierda para controlar los cambios y ejecutar las inserciones.

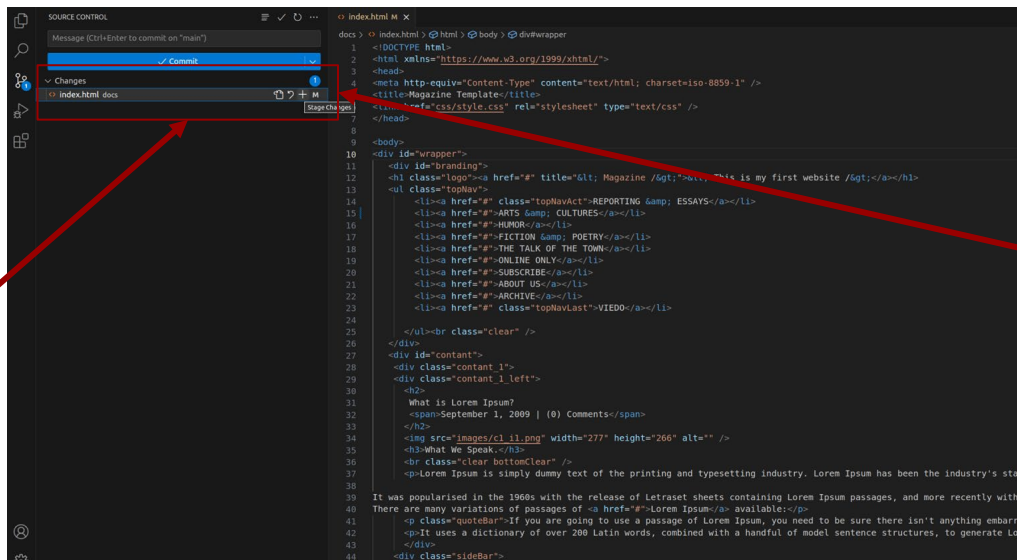


Este sistema le avisa de que no se han agregado cambios al área de preparación.

# Herramientas de desarrollo

Los IDE suelen ofrecer herramientas visuales para controlar el código fuente mediante git. El código de Visual Studio le permite agregar cambios en el área de almacenamiento provisional mediante la opción +.

Puedes elegir los archivos individualmente usando la sección git.



Puede agregar archivos individualmente usando la opción + a la derecha del archivo.

# Herramientas de desarrollo

Si quieres saber más sobre git y visual studio code.....



[https://code.visualstudio.com/docs/sourcecontrol/overview#\\_working-in-a-git-repository](https://code.visualstudio.com/docs/sourcecontrol/overview#_working-in-a-git-repository)