



# Unidad 1

## Instalaciones

### Contenidos

- Parte 1. Instalaciones

# Parte 1. Instalaciones

En esta asignatura, utilizaremos herramientas fundamentales para trabajar con tecnologías modernas y desarrollar un entorno eficiente de trabajo. Las herramientas principales son:

1. **Node.js:** Para ejecutar JavaScript fuera del navegador y gestionar paquetes.
2. **Git:** Para el control de versiones.
3. **Visual Studio Code (VSCode):** Como entorno de desarrollo integrado (IDE).

## 1. Node.js

### ¿Qué es Node.js?

Documentación Oficial | Node.js

**Node.js** es un entorno de tiempo de ejecución para **JavaScript** que permite ejecutar código fuera del navegador. Algunas de sus funcionalidades clave son:

- **Gestión de datos:** Leer, escribir y procesar información.
- **Ejecución de scripts:** Ejecuta aplicaciones JavaScript en servidores.
- **Gestión de paquetes:** Soporte para librerías y frameworks con **npm** (Node Package Manager).

### Instalación de Node.js

1. Visita la sección de descargas: **Node.js Descargas**.
2. Selecciona tu sistema operativo y descarga la versión **LTS** (Long Term Support).
3. Sigue los pasos del instalador aceptando las configuraciones predeterminadas.

### Verificación de la Instalación

Después de instalar Node.js, abre una terminal y escribe:

```
node -v  
npm -v
```

Esto mostrará las versiones instaladas de **Node.js** y **npm**.

## 2. Git

### ¿Qué es Git?

Documentación Oficial | Git

**Git** es un sistema de control de versiones diseñado para gestionar cambios en proyectos de software. Sus características principales incluyen:

- **Versionado:** Compara y restaura diferentes versiones del código.
- **Colaboración:** Facilita el trabajo en equipo mediante repositorios remotos.
- **Gestión de ramas:** Permite trabajar en varias versiones de un proyecto simultáneamente.

## Instalación de Git

1. **Accede a la página de descargas:** [Git Descargas](#).

2. **Selecciona tu sistema operativo.**

- **Windows:** Durante la instalación, asegúrate de incluir **Git Bash**.
- **MacOS/Linux:** Usa gestores de paquetes como Homebrew (Mac) o apt-get (Linux).

## Verificación de la Instalación

Para comprobar que Git está instalado, abre una terminal y escribe:

```
git --version
```

Esto mostrará la versión instalada de Git.

## 3. Visual Studio Code (VSCode)

### ¿Qué es VSCode?

Descargar VSCode

**VSCode** es un editor de código fuente desarrollado por Microsoft, ideal para lenguajes como JavaScript, Python y otros. Ofrece:

- **Extensiones:** Personaliza y mejora la funcionalidad del editor.
- **Depuración:** Identifica y corrige errores en el código.
- **Snippets:** Inserta fragmentos de código predefinidos.

## Instalación de VSCode

1. **Descarga el instalador desde** [Visual Studio Code Descargas](#).

2. **Selecciona la versión para tu sistema operativo.**

- **Linux:** Usa instaladores **.deb** o **.rpm** para facilitar el proceso.

## Extensiones Recomendadas

Para optimizar el uso de VSCode, instala las siguientes extensiones:

- **Prettier**: Formateador de código.
- **GitLens**: Integración avanzada con Git.
- **Live Server**: Servidor local para proyectos web.

Para instalar extensiones:

1. Haz clic en el ícono de extensiones en la barra lateral.
2. Busca la extensión deseada y haz clic en **Instalar**.

## 4. GitHub

### ¿Qué es GitHub?

GitHub es una plataforma que permite almacenar proyectos de manera remota utilizando **Git**. Sus funciones principales incluyen:

- **Alojamiento de repositorios**: Almacena código en servidores remotos.
- **Colaboración**: Permite trabajar en equipo en proyectos compartidos.
- **Gestión de versiones**: Historial de cambios accesible desde cualquier lugar.

### Conexión SSH con GitHub

#### 1. Genera una clave SSH:

- **Windows**: Usa Git Bash.
- **Mac/Linux**: Usa la terminal.

```
bash
Copiar código
ssh-keygen -t ed25519 -C "your_email@example.com"
```

#### 1. Copia la clave pública generada:

- **Windows**: `clip < ~/.ssh/id_ed25519.pub`
- **Mac**: `pbcopy < ~/.ssh/id_ed25519.pub`
- **Linux**: `cat ~/.ssh/id_ed25519.pub`

#### 2. Añade la clave en GitHub > Settings > SSH and GPG Keys.

#### 3. Verifica la conexión:

```
ssh -T git@github.com
```

## 5. Comandos Básicos de Git

### Configuración Inicial

- Configurar nombre de usuario y correo electrónico:

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu_email@example.com"
```

- Activar colores en los comandos:

```
git config --global color.ui auto
```

### Gestión de Repositorios

#### 1. Iniciar un repositorio local:

```
git init
```

#### 1. Clonar un repositorio remoto:

```
git clone URL_DEL_REPOSITORIO
```

#### 1. Añadir y guardar cambios:

```
git add .  
git commit -m "Descripción del cambio"
```

#### 1. Subir cambios al repositorio remoto:

```
git push origin rama
```

### Gestión de Ramas

- Crear una nueva rama:

```
git branch nombre-rama
```

- Cambiar a una rama existente:

```
git checkout nombre-rama
```

- Fusionar ramas:

```
git merge rama-origen
```

## 6. Conclusión

Con estas herramientas configuradas, estás listo para comenzar a trabajar en tus proyectos de programación. Practica su uso para familiarizarte con su funcionamiento y optimizar tu flujo de trabajo.