

Integración de github y git en un IDE

- Integración de github y git en un IDE
 - **Autenticación en Git con Tokens Personales (PAT)**
 - [Crear un Token de Acceso Personal en GitHub](#)
 - [Métodos según tu sistema operativo:](#)
 - [Para Windows:](#)
 - [Para macOS:](#)
 - [Para sistemas basados en Linux:](#)
 - [Integración con diferentes IDEs](#)
 - [Para IDEs de JetBrains \(IntelliJ, PhpStorm, WebStorm, etc.\):](#)
 - [Integrar Git y GitHub en VSCode](#)
 - [Crear un repositorio Git](#)
 - [Clonar un repositorio de GitHub](#)
 - [Push y pull](#)
 - [Otros comandos útiles de Git](#)
 - [Integrar IntelliJ IDEA y PyCharm con Git y GitHub](#)
 - [Configurar Git en IntelliJ IDEA o PyCharm](#)
 - [Crear un repositorio Git en IntelliJ IDEA o PyCharm](#)
 - [Inicializar un repositorio Git](#)
 - [Conectar con GitHub](#)
 - [Configurar el cuenta de GitHub](#)
 - [Crear un repositorio GitHub desde IntelliJ IDEA o PyCharm](#)
 - [Clonar un repositorio GitHub en IntelliJ IDEA o PyCharm](#)
 - [Push y pull en GitHub](#)
 - [Otras funcionalidades de Git dentro de IntelliJ IDEA y PyCharm](#)

Autenticación en Git con Tokens Personales (PAT)

A partir del 13 de agosto de 2021, GitHub ya no acepta contraseñas de cuenta para la autenticación de operaciones Git. Es necesario utilizar un **PAT (Token de Acceso Personal)**. Puedes seguir el siguiente método para añadir un PAT a tu sistema.

Crear un Token de Acceso Personal en GitHub

1. Desde tu cuenta de GitHub, ve a **Settings** → **Developer Settings** → **Personal Access Token** → **Tokens (classic)** → **Generate New Token**.
2. Introduce tu contraseña, rellena el formulario y haz clic en **Generate Token**.
3. Copia el Token generado, que tendrá un formato similar a:
`ghp_sFhFsSHhTzMDreGRLjmks4Tzuzgthdvfsrta` .

Métodos según tu sistema operativo:

Para Windows:

1. Ve al **Administrador de credenciales** desde el **Panel de Control** → **Credenciales de Windows** → busca `git:https://github.com` → **Editar**.
2. Sustituye la contraseña por tu **GitHub Personal Access Token**.
3. Si no encuentras `git:https://github.com` , haz clic en **Agregar una credencial genérica**.
 - La dirección de Internet será `git:https://github.com` .
 - Introduce tu nombre de usuario, y para la contraseña, utiliza tu **Personal Access Token**.
4. Haz clic en **Aceptar** y ya estará configurado.

Para macOS:

1. Haz clic en el icono de la lupa (Spotlight) en la parte derecha de la barra de menú.
2. Escribe **Acceso a Llaveros** y pulsa la tecla Intro para abrir la aplicación.
3. Busca `github.com` dentro de **Acceso a Llaveros**.
4. Encuentra la entrada de la contraseña de Internet para `github.com` y edita o elimina la entrada según sea necesario.
Ya deberías haber terminado.

Para sistemas basados en Linux:

1. Necesitas configurar el cliente local de GIT con un nombre de usuario y dirección de correo electrónico:

```
$ git config --global user.name "tu_nombre_de_usuario_github"  
$ git config --global user.email "tu_correo_github"  
$ git config -l
```

2. Una vez GIT esté configurado, puedes empezar a utilizarlo para acceder a GitHub. Ejemplo:

```
$ git clone https://github.com/TU-NOMBRE-DE-USUARIO/TU-REPOSITORIO
> Cloning into `TU-REPOSITORIO`...
Username: <introduce tu nombre de usuario>
Password: <introduce tu contraseña o token de acceso personal de GitHub>
```

3. Ahora, puedes guardar (cachear) el token en tu ordenador para recordarlo:

```
$ git config --global credential.helper cache
```

4. Si necesitas borrar la memoria caché en algún momento:

```
$ git config --global --unset credential.helper
$ git config --system --unset credential.helper
```

5. Puedes verificar con la opción `-v` cuando hagas un pull:

```
$ git pull -v
```

6. Para clonar en Debian u otras distribuciones de Linux:

```
git clone https://<tu_token_aquí>@github.com/<usuario>/<repositorio>.git
```

Integración con diferentes IDEs

Para IDEs de JetBrains (IntelliJ, PhpStorm, WebStorm, etc.):

1. Consulta la página de ayuda del IDE que utilices para más información sobre cómo iniciar sesión con un Token.
2. A continuación, tienes un resumen rápido de cómo hacerlo en **IntelliJ**:
 - Abre los **ajustes** pulsando `⌘Cmd0/CtrlAlt0` (las teclas pueden variar).
 - Selecciona **Version Control | GitHub**.
 - Haz clic en el botón **Añadir**.
 - Elige la opción **Iniciar Sesión con Token**.
 - Introduce el token generado en el campo de texto correspondiente.
 - Haz clic en **Añadir Cuenta**.

Información extraída de [stackoverflow](#).

Integrar Git y GitHub en VSCode

Una de las herramientas más potentes de VSCode es la **integración con Git**, que te permite gestionar repositorios locales y remotos (como GitHub) directamente desde el editor.

Crear un repositorio Git

Para empezar a trabajar con Git, primero necesitas crear un repositorio:

1. Inicializar un repositorio

Abre la carpeta de tu proyecto en VSCode y haz clic en el **panel de control de Git** (icono de ramas). Allí encontrarás la opción "Initialize Repository". Esto creará un repositorio Git en tu carpeta de trabajo.

2. Añadir ficheros al repositorio

Una vez creado, todos los ficheros no seguidos aparecerán en la sección "Changes". Para añadirlos a tu commit, selecciona los ficheros y haz clic en el icono "+".

3. Hacer un commit

Después de añadir ficheros, puedes crear un **commit**, que es un punto de restauración dentro de tu proyecto. Escribe un mensaje descriptivo en el campo de texto y haz clic en el icono de confirmación.

Clonar un repositorio de GitHub

Si ya tienes un repositorio en GitHub y quieres trabajar en él desde VSCode, sigue estos pasos:

1. Clonar el repositorio

Ve a `Ctrl+Shift+P` o `Cmd+Shift+P` y escribe "Git: Clone". Introduce la URL del repositorio GitHub y selecciona la carpeta donde quieres guardarlo.

2. Hacer cambios y hacer un commit

Una vez clonado el repositorio, puedes hacer cambios en los ficheros como de costumbre. Después añade los ficheros cambiados a Git, crea un commit y finalmente envíalo a GitHub.

Push y pull

1. Hacer push

Para enviar los cambios a tu repositorio remoto (GitHub), haz clic en el icono de sincronización (arriba en la barra de Git) o escribe "Git: Push" en la paleta de comandos.

2. Hacer pull

Si otros colaboradores han hecho cambios en el repositorio remoto, puedes descargar esos cambios haciendo un `pull`. Escribe "Git: Pull" en la paleta de comandos o haz clic en el icono de sincronización.

Otros comandos útiles de Git

- **Git: Checkout to...** permite cambiar de rama.
- **Git: Merge Branch...** permite fusionar ramas.
- **Git: Create Branch...** crea una nueva rama para trabajar.

Integrar IntelliJ IDEA y PyCharm con Git y GitHub

Configurar Git en IntelliJ IDEA o PyCharm

1. Abrir las preferencias del IDE

Ve a `File > Settings` (o `Preferences` en macOS) y luego busca `Version Control > Git`.

2. Comprobar la ruta de Git

El IDE intentará detectar automáticamente la ubicación del binario de Git en tu sistema. Si no lo hace correctamente, especifica la ruta manualmente. La ruta típica para Git es:

- Windows: `C:\Program Files\Git\bin\git.exe`
- macOS y Linux: `/usr/bin/git` o `/usr/local/bin/git`

3. Comprobar la configuración

Haz clic en el botón `Test` para verificar que Git está configurado correctamente y funcionando. Si todo está bien, verás el mensaje "Git executable is found".

Crear un repositorio Git en IntelliJ IDEA o PyCharm

Inicializar un repositorio Git

1. Abrir un proyecto o crear uno nuevo

Abre el proyecto existente o crea uno nuevo en IntelliJ o PyCharm.

2. Inicializar un repositorio Git

Ve a `VCS` (Version Control System) en el menú superior y selecciona

`Enable Version Control Integration`. Aparecerá una ventana emergente donde debes seleccionar `Git` como sistema de control de versiones.

3. Añadir ficheros al repositorio

Una vez inicializado el repositorio, podrás ver los **ficheros no seguidos** en el panel de "Version Control". Para añadirlos al seguimiento de Git, selecciónalos, haz clic con el botón derecho y selecciona `Git > Add`.

4. Hacer un commit

Después de añadir los ficheros, podrás hacer un **commit**, que es un punto de restauración del proyecto. Ve a `VCS > Commit` o haz clic en el icono de commit en la barra de control de versiones. Escribe un mensaje descriptivo y confirma el commit.

Conectar con GitHub

Para trabajar con GitHub, primero tienes que **conectar tu cuenta al IDE**.

Configurar el cuenta de GitHub

1. Acceder a GitHub desde el IDE

Ve a `File > Settings` (o `Preferences` en macOS) y busca `Version Control > GitHub`.

2. Iniciar sesión en GitHub

Haz clic en `Add Account`. Esto abrirá una ventana donde podrás iniciar sesión con tu cuenta de GitHub. Si lo prefieres, también puedes utilizar un **token de acceso personal (PAT)** que puedes generar en tu cuenta de GitHub.

Crear un repositorio GitHub desde IntelliJ IDEA o PyCharm

1. Crear el repositorio remoto

Una vez tienes Git integrado en tu proyecto, puedes subir tu proyecto a GitHub directamente desde el IDE. Ve a `VCS > Import into Version Control > Share Project on GitHub`.

2. Añadir una descripción

Especifica el nombre del repositorio y su descripción (opcional). El IDE creará automáticamente el repositorio remoto en GitHub y hará un `push` de los ficheros al repositorio.

Clonar un repositorio GitHub en IntelliJ IDEA o PyCharm

Si ya tienes un repositorio en GitHub y quieres clonarlo para trabajar en él localmente:

1. Clonar el repositorio

Ve a `File > New > Project from Version Control`. Selecciona `Git` e introduce la URL del repositorio GitHub que quieres clonar.

2. Seleccionar la ubicación

Indica dónde quieres guardar el proyecto clonado en tu ordenador. Haz clic en `Clone` y el IDE descargará el repositorio en la ubicación seleccionada.

Push y pull en GitHub

1. Hacer push de los cambios a GitHub

Una vez has hecho cambios y creado commits, puedes enviar esos cambios a tu repositorio remoto en GitHub haciendo clic en el icono `Push` en la barra de control de versiones o yendo a `VCS > Git > Push`. Esto enviará tus commits al repositorio remoto.

2. Hacer pull de los cambios

Si alguien más ha hecho cambios en el repositorio remoto, puedes hacer un `pull` para actualizar tu repositorio local. Ve a `VCS > Git > Pull` para descargar los cambios.

Otras funcionalidades de Git dentro de IntelliJ IDEA y PyCharm

- **Ramas (branches):** Puedes crear nuevas ramas para trabajar en funcionalidades o correcciones específicas sin afectar la rama principal (normalmente `main` o `master`). Ve a `VCS > Git > Branches > New Branch` para crear una nueva.
- **Merge:** Cuando terminas de trabajar en una rama, puedes fusionarla con la rama principal. Selecciona la rama con la que quieres trabajar en `VCS > Git > Branches` y luego selecciona `Merge` para fusionar las ramas.
- **Resolución de conflictos:** Si hay conflictos entre diferentes cambios (por ejemplo, si tú y otro colaborador habéis hecho cambios en el mismo fichero), el IDE te mostrará una interfaz para resolver los conflictos de manera visual.

Actividad

Haz una memoria en la que guardes los pasos exactos que haces para integrar github en un IDE a tu elección.