

GIT & DOCUMENTACIÓN – CHEATSHEET EXPANDIDO

== CONFIGURACIÓN ==

`git config --global user.name "Tu Nombre"`

Configura tu nombre global para los commits.

`git config --global user.email "tu.email@ejemplo.com"`

Configura tu correo global para los commits.

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

Define VS Code como editor por defecto.

`git config --global color.ui auto`

Activa colores en la terminal para Git.

== CREACIÓN Y CLONACIÓN ==

`git init`

Crea un repositorio Git en el directorio actual.

`git clone`

Descarga un repositorio remoto completo a tu máquina.

== ESTADO Y CAMBIOS ==

`git status`

Muestra los archivos modificados, nuevos y preparados.

`git add`

Añade un archivo modificado al área de preparación.

`git add .`

Añade todos los cambios detectados al área de preparación.

`git commit -m "mensaje"`

Guarda una instantánea de los cambios preparados.

`git commit --amend`

Modifica el último commit (mensaje o contenido).

`git diff`

Muestra diferencias NO preparadas.

`git diff --staged`

Muestra diferencias entre el área de preparación y el último commit.

== HISTORIAL Y REVERSIONES ==

`git log`

Muestra el historial completo de commits.

`git log --oneline`

Muestra el historial resumido en una línea por commit.

`git show`

Muestra detalles del commit indicado.

`git blame`

Indica qué commit modificó cada línea del archivo.

`git revert`

Crea un commit inverso que revierte los cambios del commit indicado.

`git reset --hard HEAD`

Elimina todos los cambios no confirmados (peligroso).

`git reset --soft`

Mueve HEAD al commit indicado sin borrar los cambios.

`git rm`

Elimina un archivo del repositorio.

`git rm --cached`

Deja de rastrear un archivo, pero no lo borra del sistema.

== RAMAS ==

`git branch`

Lista todas las ramas locales.

`git branch`

Crea una nueva rama.

`git checkout`

Cambia a otra rama.

`git checkout -b`

Crea una rama y cambia a ella.

`git merge`

Fusiona la rama indicada en la rama actual.

`git rebase`

Reescribe la historia colocando tus commits sobre otra rama.

`git rebase --continue`

Continúa el rebase tras resolver un conflicto.

== REMOTOS ==

`git remote add`

Añade un repositorio remoto con un alias.

`git remote -v`

Muestra las URLs del repositorio remoto.

`git push`

Sube tus commits al repositorio remoto.

`git push -u origin`

Establece la rama remota como predeterminada.

`git fetch`

Descarga commits SIN fusionarlos.

`git pull`

Descarga commits y los fusiona automáticamente.

== ETIQUETAS ==

`git tag`

Lista las etiquetas existentes.

`git tag -a v1.0 -m "mensaje"`

Crea una etiqueta anotada.

`git show`

Muestra detalles de una etiqueta.

`git push origin --tags`

Envía etiquetas al servidor remoto.

== .GITIGNORE ==

Patrones comunes:

- *.log — Ignora archivos de log
- node_modules/ — Ignora dependencias Node
- *.class — Ignora compilados Java
- *~ — Ignora archivos temporales de editores

== GITHUB ==

Pull Request: Flujo

- 1) git checkout -b feature-x
- 2) git push origin feature-x
- 3) Crear PR en GitHub

Fork: Copias un repo para colaborar sin permisos de escritura.

== MARKDOWN ==

Encabezados: #, ##, ### ...

Negrita: ****texto****

Cursiva: **texto**

Código: ``inline`, ```bloque````

Links: [texto](url)

Imágenes: ![alt](url)

== JAVADOC ==

Etiquetas:

- @author — Autor
- @version — Versión
- @param — Parámetro
- @return — Valor devuelto
- @throws — Excepciones
- @see — Referencias
- {@code } — Código inline
- {@link } — Enlaces internos

== XML DOC (C#) ==

Etiquetas:

- — Resumen
- — Detalles adicionales
- — Parámetros
- — Valor de retorno
- — Excepciones
- — Ejemplos
- — Código
- — Referencias
- — Ver también

