

git, sacar archivos del indice git (.gitignore)

Gemini dixit:

Esa es una excelente pregunta y es importante entender cómo funciona Git con `.gitignore` y los ficheros ya existentes.

⚠ Ficheros ya rastreados (Tracked Files)

Si el directorio `data/cache` **ya existía** en tu repositorio y contenía ficheros que Git ya estaba rastreando (*tracked*) cuando añadiste la línea `data/cache/*` al `.gitignore`, **estos ficheros seguirán siendo rastreados por Git, y las modificaciones que hagas en ellos aparecerán en `git status`.**

El fichero `.gitignore` solo afecta a los ficheros **no rastreados** (*untracked*). No tiene efecto sobre los ficheros que Git ya conoce y ha indexado.

📦 Solución: Dejar de rastrear los ficheros

Para que Git deje de rastrear los ficheros que están ahora ignorados por `.gitignore`, debes eliminarlos explícitamente del índice (el *staging area*), pero sin borrarlos de tu sistema de ficheros local.

Utiliza el siguiente comando:

```
git rm -r --cached data/cache
```

Explicación del comando

- **git rm**: El comando para eliminar ficheros de Git.
- **-r**: Indica recursividad, necesario para trabajar con un directorio completo.
- **--cached**: **Crucial**. Le dice a Git que **solo** elimine los ficheros del índice de Git (el caché) y del repositorio, pero que **deje la copia local** en el disco duro intacta.
- **data/cache**: El directorio objetivo.

📋 Pasos a seguir

Para asegurarte de que el directorio `data/cache` y su contenido son ignorados correctamente a partir de ahora, debes seguir este proceso:

1. Asegúrate de que el `.gitignore` está actualizado:

```
echo "data/cache/*" >> .gitignore
git add .gitignore
```

2. Elimina los ficheros del índice de Git (sin borrarlos del disco):

```
git rm -r --cached data/cache
```

3. Realiza un *commit* para confirmar la eliminación de los ficheros rastreados:

```
git commit -m "Stop tracking data/cache directory contents"
```

A partir de este *commit*, el contenido de `data/cache` estará ignorado, y Git ya no reportará cambios en esos

ficheros, incluso si existen localmente.

¿Te gustaría que te explicara alguna otra situación común con `.gitignore` o que te ayudara a añadir este proceso a tu *script* de *commit*?

From:

<https://matebcn.eu/> - **miguel angel torres egea**

Permanent link:

<https://matebcn.eu/tech:git:dotfiles:removefromindex>

Last update: **12/12/2025 13:44**

