

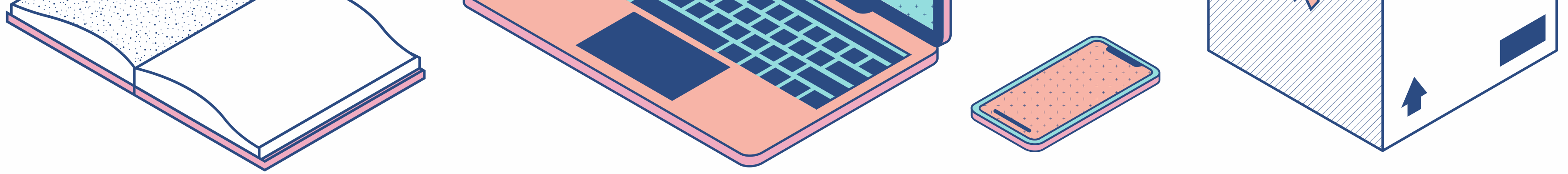


Hooks, Alias y Trucos de Git

¿Qué es un Hook?

- Un hook, o punto de enganche, es la posibilidad de ejecutar una acción o script cada vez que ocurre un evento determinado de Git.
- Hooks del lado del cliente
- Hooks del lado del servidor





Hooks del lado del cliente

Sólo afectan al repositorio local que los contiene.

- **pre-commit**
 - Podrías comprobar si se está intentando hacer un commit de demasiados archivos.
 - Puede ser un buen sitio para ejecutar el linter sobre los archivos que han sido modificados.
- **prepare-commit-msg**
 - Para modificar el mensaje del commit o añadir cualquier información extra.
- **commit-msg**
 - Es el sitio perfecto para hacer todas las comprobaciones pertinentes al mensaje.
- **post-commit**
 - Su uso principal es la de notificar por Slack.
- **pre-push**
 - Para ejecutar una batería de tests.
- **post-checkout y post-merge**
 - Permite limpiar el directorio de trabajo, tras realizar un checkout, o el de limpiar las ramas que ya no se usan tras realizar un merge.



Hooks del lado del servidor

Es interesante conocerlos ya que páginas como GitHub o GitLab los usan intensivamente a la hora de construir.

- **pre-receive**
 - Para comprobar que los commits que se quieren guardar están bien formados.
 - Verificar que el usuario que intenta grabar los commits tiene los permisos necesarios para hacerlo.
- **update**
 - Puedes evitar de una forma granular cada actualización.
- **post-receive**
 - Enviar un correo a todos los usuarios del repositorio que se han grabado nuevos cambios en el repositorio remoto
 - Reflejar en una UI las nuevas referencias, ramas o commits disponibles.

Creando mi primer hook

Para crear un propio hook sólo tienes que crear un archivo nombre-del-hook en la carpeta `.git/hooks` y en él poner el código que quieras que se ejecute.

Puedes usar todo tipo de intérpretes de lenguaje de programación como *bash*, *node*, *python*, *perl*, etc.



¿Qué es un Alias?

- Los alias permiten definir una serie de comandos que pueden ser usados en lugar de los nombres completos.

git co → **git commit**

git st → **git status**

Los alias pueden ser sencillos atajos para comandos muy usados a complejas combinaciones de comandos con parámetros...



Creando mi primer alias

Git te permite crear tus propios alias fácilmente para comandos que usas habitualmente en tu proyecto con este sistema de control de versiones.

Para crear tu propio comando debes usar el comando `git config` y ponerlo de la siguiente manera:

```
$ git config --global alias.[nombre-del-alias] "comando a ejecutar"
```

También puedes añadir alias directamente editando el archivo de configuración de git. Para ello, ejecuta `git config --list --show-origin` y edita el fichero según el ámbito donde quieres que se pueda usar tu alias (normalmente `global`).



Trucos en Git



Guarda tus cambios temporalmente

```
git stash  
git stash -u  
git stash pop
```

Aplicar cambios de commits en específico

```
git cherry-pick <SHA>
```

Detectar qué commit es el que ha introducido un bug

```
git bisect  
git bisect start  
git bisect bad  
git bisect good  
git bisect reset
```


Trucos en Git



Cambiar el nombre de un commit

```
git commit --amend -m <descripcion commit>
```

Recupera un archivo en concreto de otra rama o commit

```
git checkout <SHA> <archivo>
```
