

¿Qué es GitHub? 🖨

GitHub es una plataforma basada en la nube para alojar y gestionar proyectos de código, siendo la herramienta más popular del mundo para esto. Está construida sobre Git, un sistema de control de versiones distribuido.

¿Cuál es su utilidad?

Su principal utilidad es facilitar la colaboración y el control de versiones en proyectos de desarrollo de software. Permite a múltiples desarrolladores trabajar en el mismo código de forma segura y organizada, llevando un registro detallado de todos los cambios. Además, sirve como un portafolio para mostrar tu trabajo a la comunidad o a posibles empleadores.

¿Qué es un repositorio en GitHub?

Un repositorio (repository o repo) es esencialmente la carpeta central de un proyecto. Contiene todos los archivos del proyecto (código, documentación, imágenes, etc.) y la historia completa de todas las revisiones y cambios realizados en esos archivos. Es el elemento básico para empezar a usar GitHub.

¿Cuál es la diferencia entre un repositorio público y uno privado?

La diferencia fundamental reside en la visibilidad y el acceso:

Un repositorio público puede ser visto por cualquier persona en Internet. Cualquiera puede clonarlo, hacer una copia (fork) y proponer cambios, lo que lo hace ideal para proyectos de código abierto (open source) y para tu portafolio.

Un repositorio privado solo es visible y accesible para ti y los colaboradores específicos que tú invites. Esto lo convierte en la opción ideal para proyectos internos, código sensible de empresas o información que debe mantenerse confidencial.

¿Qué es un "commit"? ¿Por qué es importante hacer commits?

¿Qué es un "commit"?

Un commit es un "punto de guardado" o una instantánea de tu proyecto en un momento específico. Representa un conjunto de cambios lógicos (por ejemplo, "Se añade la función de validación de correo"). Cada commit incluye un mensaje descriptivo que explica el qué y el porqué de la modificación, junto con el autor y la fecha.

¿Por qué es importante hacer commits?

Es vital hacer commits de forma frecuente porque:

Seguimiento: Crea un historial claro y detallado de la evolución del proyecto.

Seguridad: Te permite revertir el proyecto a cualquier estado anterior si se introduce un error o si un cambio no funciona como esperabas.

Organización: Obliga a los desarrolladores a organizar su trabajo en unidades de cambio pequeñas y coherentes, facilitando la revisión del código y la depuración de problemas.