

SESIÓN 3: HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE UN PORTAFOLIO

GitHub es una plataforma basada en la web que permite almacenar código, colaborar con otros desarrolladores y gestionar versiones de software utilizando Git. Es ampliamente usada en el mundo del desarrollo profesional, y un excelente lugar para mostrar tus proyectos a potenciales empleadores o clientes.

 **por Kibernetum Capacitación S.A. Kibernetum Capacitación S.A.**

¿Qué es GitHub?

GitHub es una plataforma basada en la web que permite almacenar código, colaborar con otros desarrolladores y gestionar versiones de software utilizando Git. Es ampliamente usada en el mundo del desarrollo profesional, y un excelente lugar para mostrar tus proyectos a potenciales empleadores o clientes.

Características principales de GitHub:

- Repositorios públicos y privados.
- Control de versiones con Git.
- Interfaz web para explorar código y documentación.
- Integración con GitHub Pages para alojar sitios estáticos.
- Comunidad activa para compartir y colaborar.
- Issues y Pull Requests para seguimiento de cambios y feedback.

Crear una cuenta y configurar tu perfil

Para crear una cuenta en GitHub es necesario seguir los siguientes pasos:

1. Ve a <https://github.com>.
2. Haz clic en Sign up.
3. Ingresa un nombre de usuario único (este será parte de la URL de tu perfil, por ejemplo: <https://github.com/tu-usuario>).
4. Ingresa tu correo electrónico y una contraseña segura.
5. Completa la verificación de seguridad y haz clic en Create account.
6. GitHub te guiará por algunos pasos opcionales, como elegir tu nivel de experiencia o tipos de proyectos de interés.

Configurar el perfil de GitHub

Una vez que tienes tu cuenta, tu perfil es tu carta de presentación ante la comunidad. Aquí van algunas buenas prácticas para optimizar tu perfil:

Foto de perfil

- Usa una imagen profesional o al menos clara.
- Es tu primer punto de contacto visual con quien visita tu perfil.

Biografía corta (bio)

- Escribe una frase que describa tu enfoque como desarrollador o tus intereses.
- Ejemplo: "Desarrollador web enfocado en accesibilidad y rendimiento | Explorando proyectos con Vue y Firebase."

Ubicación y redes

- Agrega tu país/ciudad.
- Vincula tu portafolio, LinkedIn, sitio web o blog si tienes.

Repositorios destacados y README del perfil

Repositorios destacados

Puedes "pinear" (fijar) los repositorios más importantes que quieras mostrar en la parte superior de tu perfil.

Ejemplo: proyectos personales, trabajos en equipo, clonaciones de sitios famosos, prácticas técnicas bien resueltas.

README del perfil

GitHub permite tener un repositorio especial con un archivo README.md que se muestra directamente en tu perfil.

Este README puede incluir:

- Tu presentación personal.
- Tecnologías que dominas (con íconos si quieres).
- Proyectos destacados.
- Cómo contactarte.

¿Cómo crear un README de perfil?

1. Crea un nuevo repositorio con exactamente el mismo nombre que tu usuario. Ejemplo: Si tu usuario es ana-dev, crea un repositorio llamado ana-dev.
2. Marca la opción "Add a README file".
3. GitHub detectará que coincide con tu nombre de usuario y mostrará el contenido del README.md directamente en tu perfil.

Crear un nuevo repositorio

Un repositorio en GitHub permite almacenar el proyecto en la nube de forma segura y accesible desde cualquier lugar. Compartir el trabajo con otras personas y de esta forma, colaborar en equipo.



Acceder a GitHub

Ingresa a <https://github.com> y accede con tu cuenta.



Iniciar creación

Haz clic en el botón "New" o ve a: <https://github.com/new>.



Completar información

- Repository name: Usa un nombre claro y corto. Ejemplo: portafolio-web.
- Description (opcional): "Sitio web personal con mis proyectos destacados".
- Visibility: Puedes seleccionar Public (visible para todos) o Private.
- Initialize this repository with: No marques esta opción si ya tienes el proyecto en tu computador. Si estás empezando desde cero, puedes marcar "Add a README file".



Finalizar

Haz clic en "Create repository".

Subir archivos o hacer push desde Git local

Si ya tienes tu proyecto creado en tu computador (es decir, un conjunto de archivos y carpetas con código), puedes subirlo a GitHub para compartirlo, mostrarlo en tu portafolio o colaborar con otros. Para eso, debes inicializar un repositorio local y conectarlo con un repositorio remoto en GitHub.

1. Los pasos a seguir son:
2. Navegar a la carpeta del proyecto.
3. Inicializar un repositorio Git.
4. Agregar todos los archivos al área de staging.
5. Realizar el commit.
6. Conectar el proyecto local con el repositorio remoto de GitHub.
7. Subirlo realizando el push.

A continuación, podrás ver los comandos necesarios para realizar estos pasos.

Comandos Git para subir un proyecto

```
# Navegar a la carpeta del proyecto
cd ruta/a/tu-proyecto

# Inicializar un repositorio Git
git init

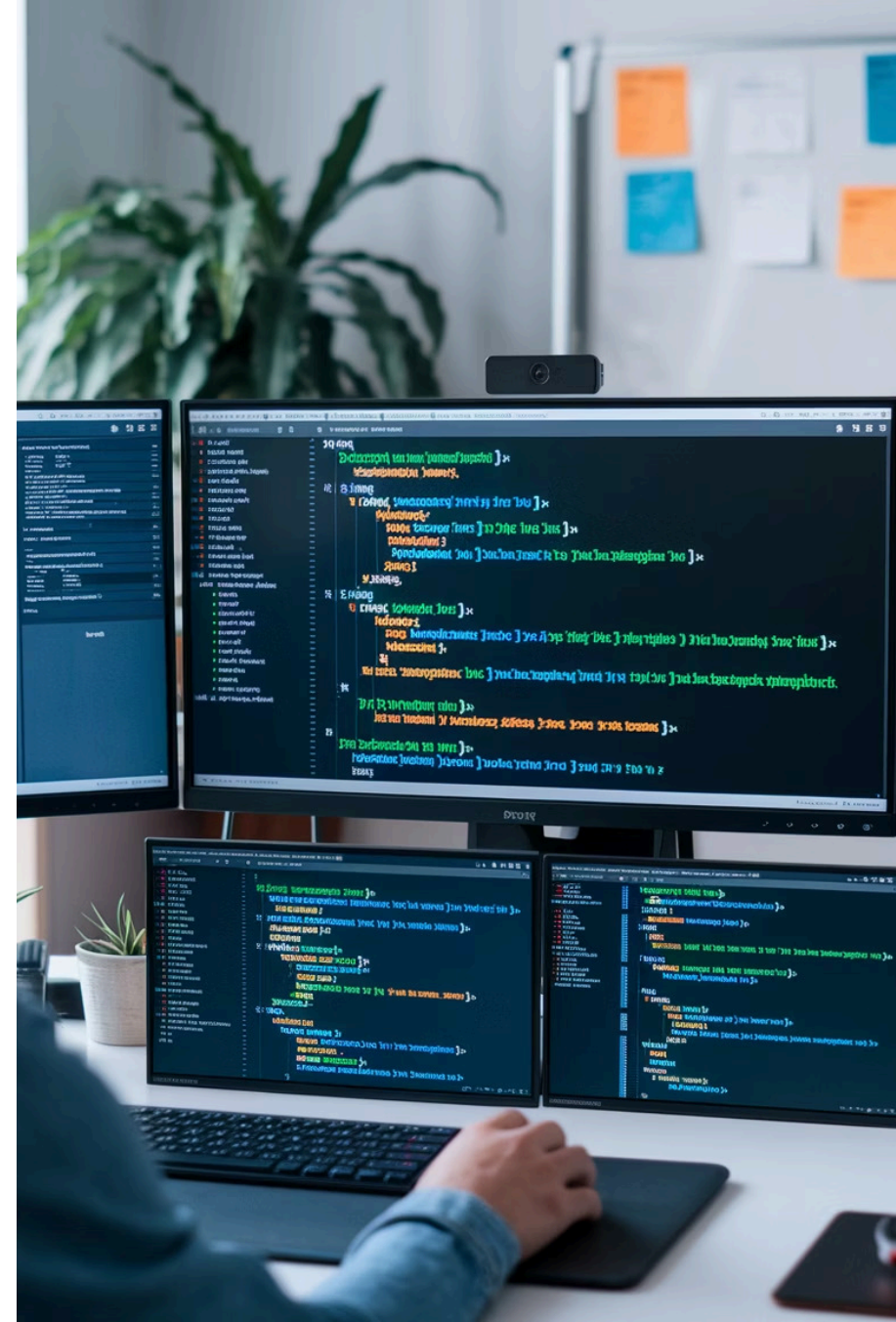
# Agregar todos los archivos
git add .

# Realizar el primer commit
git commit -m "Primer commit: Versión inicial del proyecto"

# Conectar con el repositorio remoto
git remote add origin https://github.com/tu-usuario/nombre-repositorio.git

# Subir el código
git push -u origin main
```

Estos comandos te permitirán inicializar un repositorio local, preparar tus archivos, crear un commit con un mensaje descriptivo, conectar con tu repositorio remoto en GitHub y finalmente subir todo el código a la nube.



Usar README.md para documentar tu proyecto

El archivo README.md es lo primero que las personas ven al entrar a tu repositorio en GitHub. Funciona como la portada de tu proyecto y cumple un rol clave en explicar de qué se trata, cómo se usa y por qué es valioso.



Explica el propósito del proyecto

¿Qué problema resuelve? ¿Para qué sirve?



Indica cómo instalar o ejecutar el proyecto

Especialmente útil para otros desarrolladores.



Muestra ejemplos de uso

Capturas de pantalla o GIFs para hacerlo visual y atractivo.



Detalla las tecnologías utilizadas

Dependencias y estructura del código.



Facilita la colaboración

Explica cómo contribuir al proyecto si es público.

Crear ramas (branches) y fusionarlas (merge)

Las ramas en Git te permiten trabajar en nuevas funcionalidades o correcciones sin afectar el código principal (normalmente en main). Son ideales para experimentar sin romper nada.

Por otro lado, el merge o fusión de ramas debe ocurrir una vez que terminas los cambios y todo funciona bien, pudiendo unir la rama en la que has trabajado con la rama principal.

A continuación, puedes ver los comandos necesarios:

Crear y cambiar a una nueva rama

```
# Crear y cambiar a una nueva rama
git checkout -b nueva-funcionalidad

# Trabajar en los archivos...

# Agregar y hacer commit de los cambios
git add .
git commit -m "Implementa nueva funcionalidad"
```

Fusionar ramas

```
# Volver a la rama principal
git checkout main

# Traer los cambios de la otra rama
git merge nueva-funcionalidad

# Subir los cambios a GitHub
git push origin main
```

Usar GitHub Pages para alojar tu proyecto

GitHub te permite almacenar tus proyectos en la nube de forma gratuita, organizarlos, compartirlos y trabajar colaborativamente. Es mucho más que un lugar para guardar código: es una herramienta profesional que muestra tu trabajo al mundo.

¿Por qué usar GitHub para alojar proyectos?



Acceso desde cualquier lugar

Tus proyectos están siempre disponibles en línea.



Visibilidad

Puedes compartir enlaces con potenciales empleadores, clientes o colaboradores.



Colaboración

Permite trabajar en equipo con control de versiones y seguimiento de cambios.



Documentación integrada

Puedes incluir un README.md, issues, wikis y más.



Historial y control

Git registra cada cambio, quién lo hizo y cuándo.

Además, puedes usar GitHub Pages para alojar sitios web estáticos, como tu portafolio, directamente desde un repositorio.

Buenas prácticas para tener un portafolio atractivo en GitHub

Tu portafolio en GitHub es mucho más que una colección de proyectos: es una vitrina profesional que muestra tu forma de trabajar, tus habilidades técnicas y tu capacidad de organización. Un buen portafolio puede marcar la diferencia al postular a empleos, presentar tus trabajos o colaborar con otros desarrolladores.

Organiza tus repositorios

Usa nombres claros, descriptivos y consistentes. Elimina o archiva proyectos obsoletos o sin contenido útil.

Cuida la documentación

Cada proyecto debe tener al menos un README.md que explique de forma clara de qué trata, cómo instalarlo, cómo usarlo y qué tecnologías utiliza.

Agrega una licencia

Si quieres compartir tu código, incluye una licencia de código abierto para proteger tu trabajo y permitir su reutilización responsable.

Incluye capturas de pantalla o GIFs

Una imagen o demo animada ayuda a comprender rápidamente la funcionalidad y el diseño de tu proyecto.

Usa etiquetas y topics

Añadir etiquetas como JavaScript, Vue, API-REST mejora la visibilidad, el filtrado y la organización de tus repositorios.

Buenas prácticas para la página principal de un repositorio

La página principal de cada repositorio muestra el contenido del archivo README.md. Es la introducción visual y técnica a tu proyecto, por lo que debe ser clara, completa y bien estructurada.

¿Qué debe tener un buen README.md?

Descripción clara del proyecto

Explica brevemente qué hace el proyecto, su propósito y su contexto.

Tabla de contenido (opcional)

Si el archivo es largo, una tabla de contenido ayuda a navegar rápidamente por las secciones.

Instrucciones de instalación y ejecución

Expón los pasos necesarios para ejecutar el proyecto de forma local, incluyendo dependencias y comandos.

Ejemplos de uso

Incluye ejemplos prácticos (con código o imágenes) para mostrar cómo funciona tu proyecto.

Listado de tecnologías

Menciona lenguajes, frameworks, librerías, APIs u otras herramientas que utilizaste.

Enlaces a demos o versiones en línea

Si tienes una demo publicada (ej. GitHub Pages, Vercel, Netlify), incluye el enlace para que puedan probarlo al instante.

Buenas prácticas para la página de perfil de GitHub

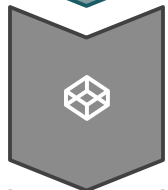
La página de perfil es tu carta de presentación personal dentro de GitHub. Ahí se ve tu foto, tu nombre, tu biografía, enlaces externos y tus repositorios destacados. Puedes personalizar aún más esta sección usando un README.md especial.

Cómo optimizar tu perfil:



Agrega una biografía corta

Resume quién eres y qué haces en una o dos frases. Ejemplo: "Desarrolladora Fullstack apasionada por la educación y el software libre."



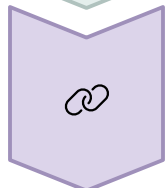
Menciona tus intereses tecnológicos

Puedes hablar brevemente sobre tecnologías, lenguajes o áreas que te entusiasman: frontend, backend, IA, accesibilidad, etc.



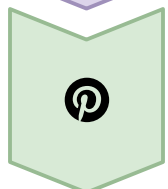
Personaliza con un README especial

Crea un repositorio con el mismo nombre que tu usuario e incluye un README.md. Su contenido se mostrará directamente en tu perfil. Úsalo para presentarte, compartir proyectos destacados, tecnologías y formas de contacto.



Incluye enlaces relevantes

Agrega enlaces a tu LinkedIn, sitio web, portafolio personal o redes profesionales.



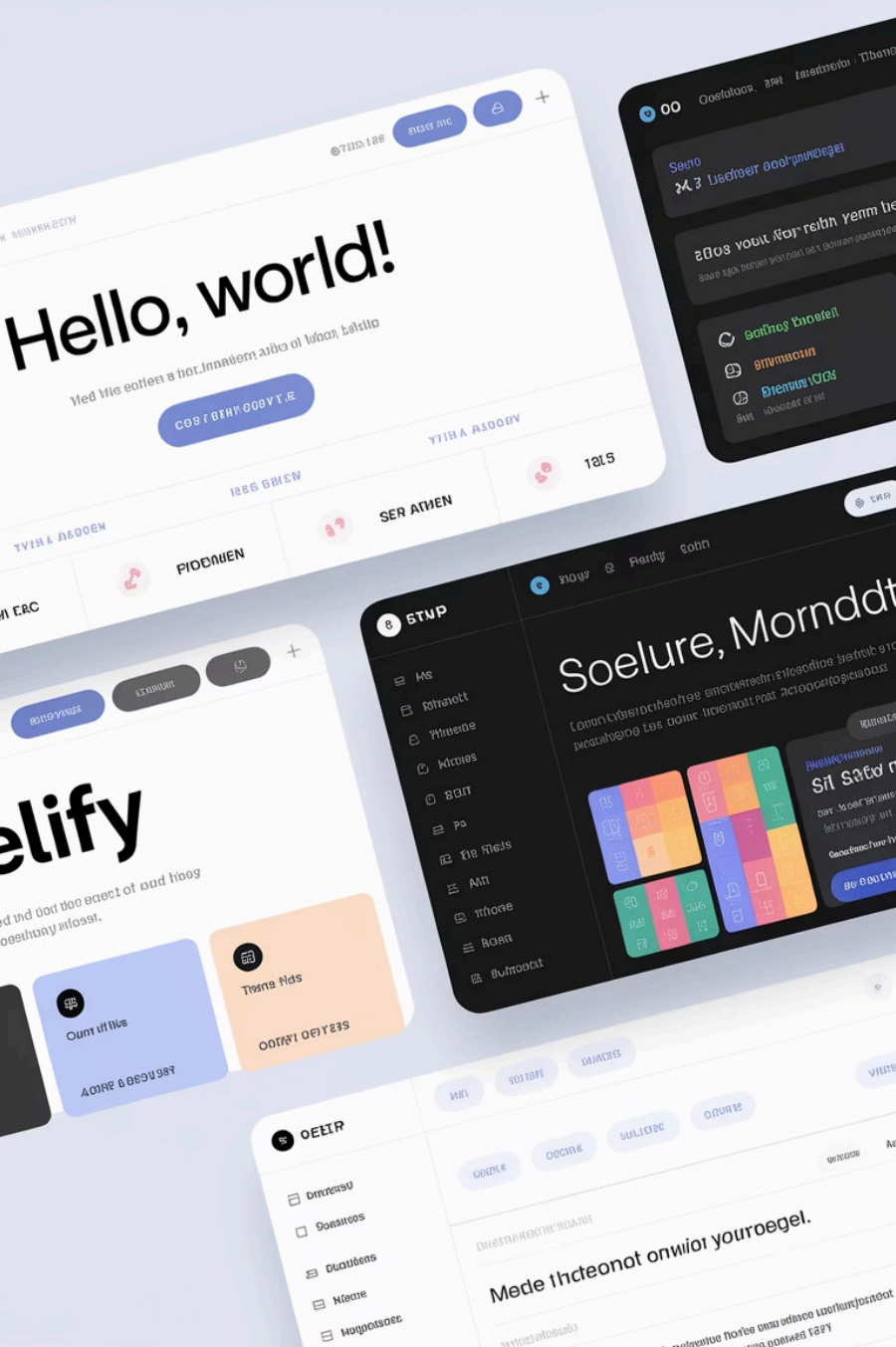
Fija tus proyectos más importantes

Usa la función "Pin repositories" para destacar aquellos que mejor representan tu trabajo y habilidades.



Cuida tu imagen y nombre

Usa una foto profesional (o al menos clara y amigable) y asegúrate de que tu nombre de usuario sea fácil de reconocer o vincular contigo.



ALOJAMIENTO DE TU PRODUCTO EN UN SERVIDOR

Servicios gratuitos de hosting:



GitHub Pages

Ideal para páginas estáticas.



Netlify

Hosting estático con CI/CD y soporte para frameworks modernos.



Vercel

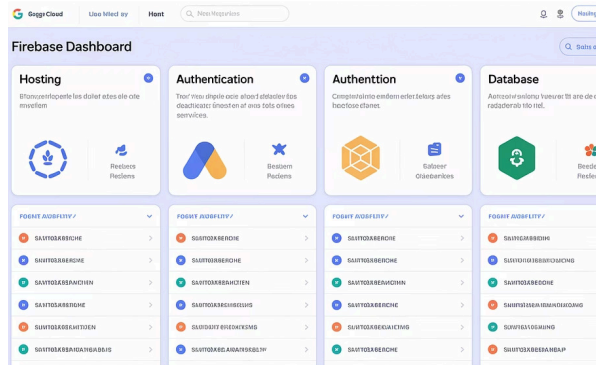
Ideal para proyectos con Next.js, pero sirve para otros también.



Render

Permite desplegar backend y frontend con cuenta gratuita.

Servicios cloud gratuitos

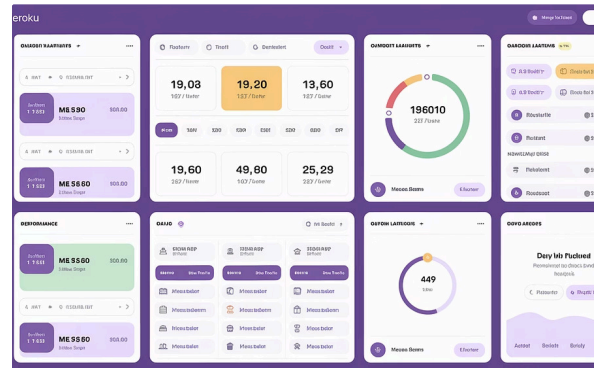


Firestore (Google)

Hosting, autenticación, base de datos y más.

Plataforma completa para desarrollo web y móvil con múltiples servicios integrados.

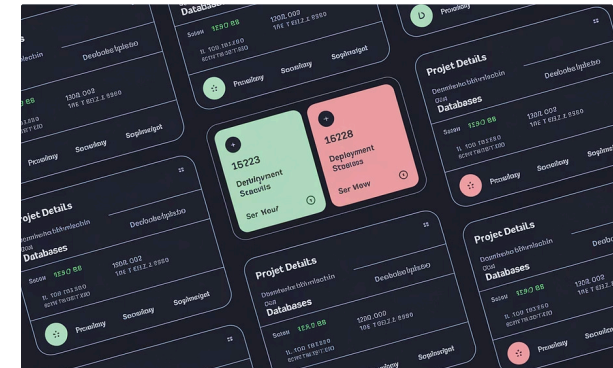
También está disponible [Replit](#): Editor y despliegue en línea para varios lenguajes.



Heroku (Free Tier)

Hosting para aplicaciones backend.

Plataforma que facilita el despliegue de aplicaciones en varios lenguajes de programación.



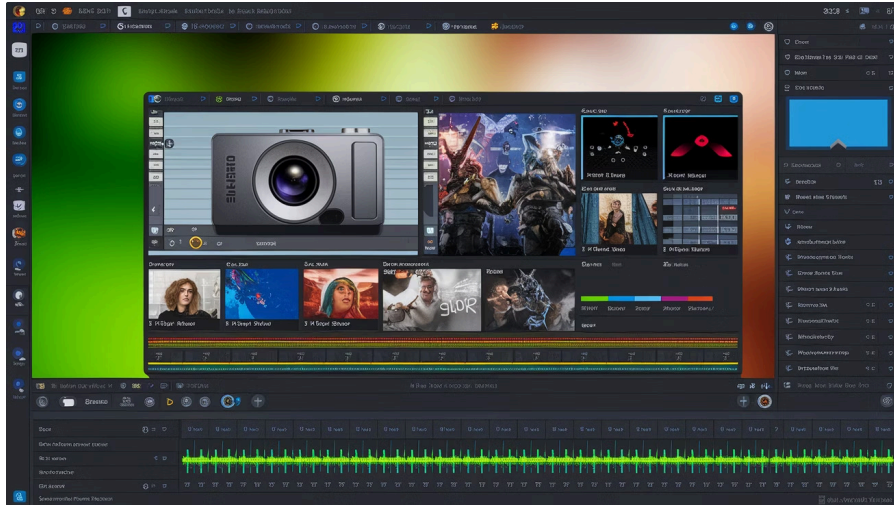
Railway

Plataforma sencilla para subir proyectos con base de datos.

Solución moderna para desplegar aplicaciones completas con infraestructura incluida.

UTILIZACIÓN DE YOUTUBE

Herramientas gratuitas para confeccionar un video técnico:



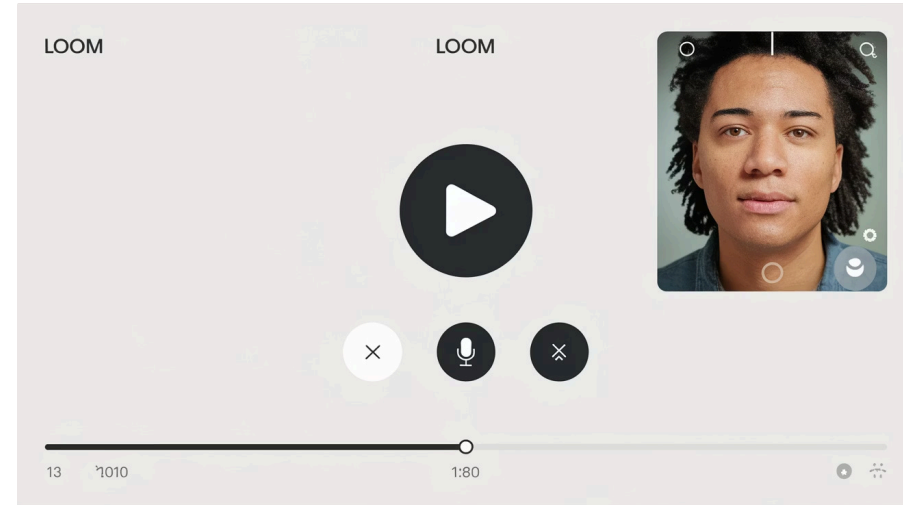
OBS Studio

Graba tu pantalla y voz de forma profesional.

Software completo para grabación y streaming con múltiples fuentes y escenas.

Otras opciones incluyen:

- **ScreenCast-O-Matic**: Ideal para grabaciones rápidas y edición básica.
- **Shotcut / DaVinci Resolve**: Editores de video gratuitos y potentes.



Loom

Graba rápidamente y comparte videos desde el navegador.

Herramienta sencilla para crear tutoriales y presentaciones con facilidad de compartir.

Cómo confeccionar un tutorial técnico en YouTube

Define el objetivo del video

¿Qué quieres mostrar? ¿Qué aprenderá quien lo vea?

Planifica tu guion

Haz una introducción breve, explica el contexto, muestra el código o funcionalidad y cierra con una reflexión o próximos pasos.

Sé claro y conciso

Evita explicaciones excesivamente largas.

Usa buena calidad de audio y video

Graba en un lugar tranquilo, con voz clara.

Agrega subtítulos si puedes

Ayuda a la accesibilidad.

Cómo presentar tu proyecto en YouTube



Título atractivo y descriptivo

Ejemplo: "Sistema de gestión de tareas con Vue y Firebase – Portafolio Técnico"



Descripción completa del video

Agrega enlaces al código en GitHub, una demo si existe, y tus redes.



Miniatura personalizada

Aumenta las visitas y visibilidad.



Capítulos o marcas de tiempo

Divide el video en secciones temáticas.



Incluye una llamada a la acción

"Puedes ver el código en GitHub en el enlace de la descripción", o "Sígueme para ver más proyectos".

Recursos adicionales para tu portafolio

Plataformas complementarias

Además de GitHub y YouTube, considera utilizar estas plataformas para ampliar tu presencia profesional:

- LinkedIn para networking profesional
- Medium o Dev.to para escribir artículos técnicos
- Twitter/X para compartir actualizaciones y conectar con la comunidad
- Behance o Dribbble si tu trabajo incluye diseño

Consejos finales

Para maximizar el impacto de tu portafolio:

- Mantén tus repositorios actualizados
- Contribuye a proyectos open source
- Documenta tu proceso de aprendizaje
- Solicita feedback de otros desarrolladores
- Revisa periódicamente tu perfil para mantenerlo relevante