

SESIÓN 3: HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE UN PORTAFOLIO.

Objetivo: Implementar un portafolio de un producto digital utilizando técnicas y herramientas utilizadas en la industria para dicho propósito.

Temario:

- Utilización de GitHub para crear un portafolio: qué es GitHub, características y operaciones básicas.
- Buenas prácticas para tener un portafolio atractivo en GitHub.
- Buenas prácticas para la página principal de un repositorio.
- Buenas prácticas la página de perfil.
- Alojamiento de tu producto en un servidor: servicios gratuitos de hosting.
- Servicios cloud gratuitos.
- Utilización de YouTube: herramientas gratuitas para la confección de un video técnico.
- Cómo confeccionar un tutorial técnico en YouTube para demostrar tus capacidades.
- Cómo presentar tu proyecto en YouTube.


UTILIZACIÓN DE GITHUB PARA CREAR UN PORTAFOLIO: QUÉ ES GITHUB, CARACTERÍSTICAS Y OPERACIONES BÁSICAS

GitHub es una plataforma en línea basada en Git, un sistema de control de versiones distribuido, que permite a los desarrolladores almacenar, gestionar y colaborar en proyectos de software. Es ampliamente utilizado por profesionales y estudiantes para crear repositorios donde se almacena el código fuente de sus proyectos. Además, GitHub ha evolucionado más allá de ser solo una herramienta para el control de versiones; ahora también es una plataforma para mostrar habilidades técnicas mediante la creación de un portafolio profesional.

Qué es GitHub

GitHub es un servicio de alojamiento de repositorios de código que facilita la colaboración entre desarrolladores. Permite a los usuarios compartir su trabajo con otros, recibir comentarios, realizar mejoras y mantener un historial de cambios en el código. Además, ofrece funcionalidades como la gestión de problemas (issues), la revisión de código mediante pull requests y la integración con herramientas de desarrollo continuo (CI/CD).

Características principales de GitHub



Control de versiones : Permite rastrear cambios en el código, revertir versiones anteriores y colaborar sin sobrescribir el trabajo de otros.

Colaboración : Facilita la colaboración en equipo mediante ramas (branches), pull requests y revisiones de código.

Integración con herramientas : Se integra con servicios como Travis CI, Jenkins y muchas otras herramientas de desarrollo.

Portafolio público : Los repositorios públicos permiten mostrar tus habilidades a posibles empleadores o colaboradores.

Documentación : Soporta archivos README.md para documentar proyectos y proporcionar información clara sobre su propósito y uso.

Gists : Pequeños fragmentos de código que pueden ser compartidos rápidamente.

Operaciones básicas en GitHub

Crear un repositorio : Desde la página principal de GitHub, selecciona "New Repository" e introduce un nombre descriptivo.

Clonar un repositorio : Usa el comando `git clone <URL>` para descargar una copia local del repositorio.

Realizar commits : Guarda cambios en tu repositorio local con `git commit -m "mensaje"`.

Subir cambios : Usa `git push` para enviar tus cambios al repositorio remoto en GitHub.

Crear ramas : Usa `git branch <nombre>` para crear una nueva rama y `git checkout <nombre>` para cambiar a ella.

Resolver conflictos : Cuando dos personas modifican la misma parte del código, GitHub te guiará para resolver conflictos manualmente.


Pull requests : Proponer cambios en un repositorio colaborativo mediante una solicitud de fusión.

BUENAS PRÁCTICAS PARA TENER UN PORTAFOLIO ATRACTIVO EN GITHUB

Un portafolio en GitHub no solo debe demostrar tus habilidades técnicas, sino también ser visualmente atractivo y fácil de navegar. Aquí hay algunas recomendaciones:

Estructura clara : Organiza tus repositorios por categorías (proyectos personales, colaboraciones, tutoriales).

Nombres descriptivos : Usa nombres claros y específicos para tus repositorios (por ejemplo, "Proyecto-Web-React" en lugar de "Proyecto1").



README bien estructurado : Incluye una descripción detallada del proyecto, instrucciones de instalación y capturas de pantalla.

Uso de emojis : Añade emojis en el README para hacerlo más atractivo y fácil de leer.

Licencias : Agrega una licencia para indicar cómo pueden usarse tus proyectos.

Contribuciones destacadas : Si has contribuido a proyectos de código abierto, asegúrate de que estén visibles en tu perfil.

Actualizaciones frecuentes : Mantén tus repositorios actualizados para reflejar tu progreso continuo.

Enlaces externos : Incluye enlaces a demos en vivo, documentación adicional o artículos relacionados.

BUENAS PRÁCTICAS PARA LA PÁGINA PRINCIPAL DE UN REPOSITORIO

La página principal de un repositorio es la primera impresión que tendrán los visitantes. Por lo tanto, debe ser informativa y atractiva:

Título claro : El nombre del repositorio debe reflejar su propósito.

Descripción breve : Una frase que explique qué hace el proyecto.

Sección de instalación : Proporciona pasos claros para configurar el proyecto localmente.

Capturas de pantalla o GIFs : Muestra cómo funciona el proyecto.

Sección de uso : Explica cómo interactuar con el proyecto.

Contribuciones : Indica cómo otros pueden contribuir al proyecto.

Contacto : Incluye información de contacto o enlaces a redes sociales.

Buenas prácticas para la página de perfil

Tu página de perfil en GitHub es como tu currículum digital. Sigue estas recomendaciones:

Foto profesional : Usa una imagen clara y profesional.

Bio concisa : Describe tus habilidades y objetivos en unas pocas líneas.

Enlaces destacados : Añade enlaces a tu portafolio, blog o LinkedIn.

Pinned repositories : Fija tus mejores proyectos en la parte superior.

Contribución gráfica : Mantén un gráfico de contribuciones activo para mostrar tu compromiso.

Personalización : Usa GitHub Pages para crear un sitio web personalizado vinculado a tu perfil.

Alojamiento de tu producto en un servidor: servicios gratuitos de hosting

Si deseas mostrar tus proyectos en línea, necesitarás un servicio de hosting. Aquí hay algunas opciones gratuitas:

GitHub Pages : Ideal para sitios estáticos. Compatible con HTML, CSS y JavaScript.

Netlify : Ofrece hosting gratuito para proyectos estáticos con integración continua.

Vercel : Perfecto para aplicaciones frontend modernas, como React o Vue.js.

Heroku : Aloja aplicaciones backend y bases de datos con un plan gratuito limitado.

Glitch : Permite editar y alojar proyectos en tiempo real.

Servicios cloud gratuitos

Los servicios cloud gratuitos son útiles para almacenar datos, ejecutar aplicaciones y gestionar recursos. Algunas opciones incluyen:

Google Cloud Platform (GCP) : Ofrece créditos gratuitos para nuevos usuarios.

AWS Free Tier : Incluye servicios como EC2 y S3 con límites de uso.

Microsoft Azure : Proporciona créditos iniciales y acceso a servicios básicos.

Firebase : Ideal para aplicaciones móviles y web con autenticación y bases de datos en tiempo real.

Utilización de YouTube: herramientas gratuitas para la confección de un video técnico

YouTube es una plataforma poderosa para compartir conocimientos técnicos. Aquí hay herramientas gratuitas para crear videos:

OBS Studio : Software de grabación y transmisión en vivo.

DaVinci Resolve : Editor de video profesional gratuito.


Canva : Diseño de miniaturas atractivas.

Audacity : Edición de audio para mejorar la calidad del sonido.

Kapwing : Herramienta en línea para agregar subtítulos y efectos.

CÓMO CONFECCIONAR UN TUTORIAL TÉCNICO EN YOUTUBE PARA DEMOSTRAR TUS CAPACIDADES

Crear un tutorial técnico requiere planificación y claridad:



Define el objetivo : Decide qué quieres enseñar y quién es tu audiencia.

Estructura el contenido : Divide el tutorial en secciones claras (introducción, pasos, conclusión).

Graba en alta calidad : Usa un micrófono decente y asegúrate de que el video sea nítido.

Edita profesionalmente : Corta partes innecesarias y añade transiciones.

Incluye subtítulos : Facilita la comprensión para todos los espectadores.

Promueve interacción : Pide a los espectadores que comenten o sugieran temas futuros.

Cómo presentar tu proyecto en YouTube

PRESENTAR TU PROYECTO EN YOUTUBE IMPLICA DESTACAR SU VALOR Y FUNCIONALIDAD:

Introducción impactante : Captura la atención desde el principio.

Demostración práctica : Muestra cómo funciona el proyecto en tiempo real.

Explica el proceso : Habla sobre los desafíos y soluciones durante el desarrollo.

Incluye código relevante : Muestra fragmentos clave del código con explicaciones.

Conclusión clara : Resume los puntos principales y proporciona enlaces adicionales.

Al seguir estas pautas, podrás maximizar el impacto de tu presencia en GitHub y YouTube, destacando tus habilidades técnicas y atrayendo oportunidades profesionales.