

Entornos de Desarrollo (ED) Unidad de Trabajo 10 Tarea 05

OBJETIVOS DE APRENDIZAJE

- Crear un proyecto de programación básico que utilice integración y despliegue continuo.

TA05

Tarea ED3i1005: Hola CI/CD – Proyecto con integración y despliegue continuos

Contexto y objetivo

En esta tarea vas a desarrollar una pequeña aplicación web que muestre un mensaje en pantalla, y configurarás un flujo de integración y despliegue continuos usando **GitHub Actions**. El resultado final debe estar accesible públicamente desde **GitHub Pages**.

El objetivo es que comprendas cómo se pueden automatizar las pruebas y el despliegue de una aplicación web real, con herramientas ampliamente utilizadas en el mundo profesional.

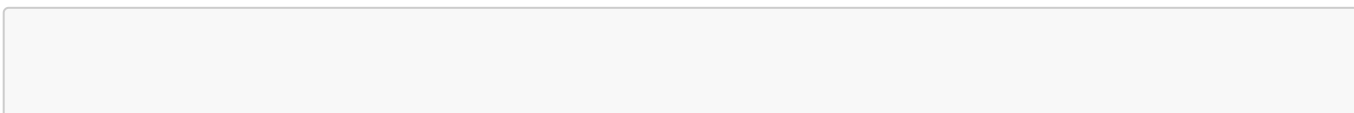
Tecnologías y herramientas utilizadas

- **HTML, CSS y JavaScript** (aplicación web básica)
 - **Git y GitHub** (control de versiones y repositorio remoto)
 - **GitHub Actions** (automatización del flujo CI/CD)
 - **GitHub Pages** (despliegue gratuito de sitios estáticos)
 - **Markdown (README.md)** para documentar el proyecto
-

Qué debe hacer tu aplicación

- Mostrar el mensaje **"Hola CI/CD"** en pantalla al abrir la página.
 - Tener una función en JavaScript (**getSaludo**) que devuelva ese mismo mensaje.
 - Tener una prueba que compruebe que **getSaludo()** devuelve **"Hola CI/CD"** correctamente.
 - Subir los archivos a GitHub en un repositorio propio.
 - Configurar un flujo de CI/CD que:
 - Ejecute los tests automáticamente cuando hagas **push**
 - Despliegue automáticamente la aplicación en GitHub Pages si los tests pasan
-

Estructura del proyecto sugerida



```
/hola-ci-cd/
├── index.html
├── style.css
├── script.js
├── test/
│   └── test.js
├── .github/
│   └── workflows/
│       └── ci.yml
└── README.md
```

Fases de trabajo sugeridas

Fase	Actividades
1. Diseño del mensaje	Crear <code>index.html</code> y <code>style.css</code> para mostrar el mensaje
2. Lógica en JS	Implementar <code>getSaludo()</code> en <code>script.js</code>
3. Prueba	Escribir una prueba que compruebe el valor retornado por <code>getSaludo()</code>
4. Git y GitHub	Crear repositorio, subir código, hacer commits
5. CI con GitHub Actions	Crear archivo <code>ci.yml</code> que ejecute la prueba en cada push
6. CD con GitHub Pages	Configurar despliegue automático si el test pasa
7. Documentación	Redactar un breve <code>README.md</code> con lo aprendido y el enlace al despliegue

Rúbrica

Nivel	Descripción del logro
0	No entrega o entrega sin funcionalidad relevante.
3	Entrega mínima: se muestra el mensaje en pantalla, pero sin uso de GitHub.
6	Uso de Git y GitHub correcto, mensaje en pantalla y función <code>getSaludo()</code> implementada. Faltan pruebas o no funciona el flujo CI/CD.
9	Todo el flujo funcional: prueba automatizada correcta, CI/CD configurado, despliegue automático activo y accesible públicamente. Documentación básica.
10	Aporta elementos personales añadidos