

Repositorio de artefactos

TicTacToe Didáctico (Triqui Online)

Angie Giseth Juez Anzola

Fecha: 12/06/2025



Tabla de contenido

Introducción	3
1. Refactorización del código JavaScript	4
2. Optimización de la hoja de estilos CSS.....	4
3. Mejora de accesibilidad web (A11y).....	4
Proceso Repositorio Git- GitHub.....	5
Conclusión.....	9

Introducción

El presente plan de mejora continua se formula con el fin de fortalecer la calidad del proyecto **TicTacToe** desde una perspectiva técnica, funcional y de accesibilidad. A pesar de que el sistema ha cumplido exitosamente con los criterios establecidos en el plan de pruebas y no se han identificado incidencias críticas, se reconoce la importancia de mantener una cultura de mejora constante que permita escalar el sistema, optimizar su mantenimiento y ampliar su alcance a usuarios con diferentes capacidades.

Las acciones propuestas a continuación están orientadas a refactorizar el código, mejorar la experiencia de usuario en diversos dispositivos y asegurar el cumplimiento de estándares de accesibilidad web.

1. Refactorización del código JavaScript

Objetivo: Mejorar la mantenibilidad y claridad del código, facilitando futuras extensiones o corrección de errores.

- Modularizar funciones repetitivas o de alta responsabilidad (como la validación de victoria y empate) en funciones puras reutilizables.
- Reemplazar variables globales con estructuras más organizadas (como objetos o módulos).
- Utilizar const y let adecuadamente, eliminando el uso de var.

2. Optimización de la hoja de estilos CSS

Objetivo: Incrementar la eficiencia del diseño, mejorar la adaptabilidad y facilitar futuras modificaciones estéticas.

- Incorporar variables CSS para colores, tipografías y tamaños frecuentes, facilitando el mantenimiento de la identidad visual.
- Agrupar y simplificar selectores, eliminando clases no utilizadas o estilos redundantes.
- Evaluar el uso de una arquitectura CSS (como BEM) si se escala el proyecto.

3. Mejora de accesibilidad web (A11y)

Objetivo: Garantizar la inclusión digital de personas con discapacidades y cumplimiento con estándares de accesibilidad.

- Añadir atributos ARIA (como aria-label, role, aria-live) para describir elementos dinámicos como el tablero de juego o el turno actual.
- Aumentar el contraste de colores si es necesario, para cumplir con los estándares WCAG (mínimo AA).
- Asegurar que el foco visual (focus outline) sea visible en todos los elementos interactivos

Proceso Repositorio Git- GitHub

Para poder realizar la corrección del código usé Virtual Studio Code lo que da bastantes ventajas en este proceso, ya que es un editor de código que nos permite hacer conexión directa con Git, lo cual permite el control de versiones y la sincronización de proyectos, realicé los siguientes pasos:

1. Instalación de Git

Se descargue e instale Git desde su sitio oficial en mi caso uso Windows por lo que lo descargue desde [Git - Downloading Package](#), esta herramienta es necesaria para gestionar los repositorios de manera local. Además, se utilicé Git Bash como terminal para la configuración inicial.

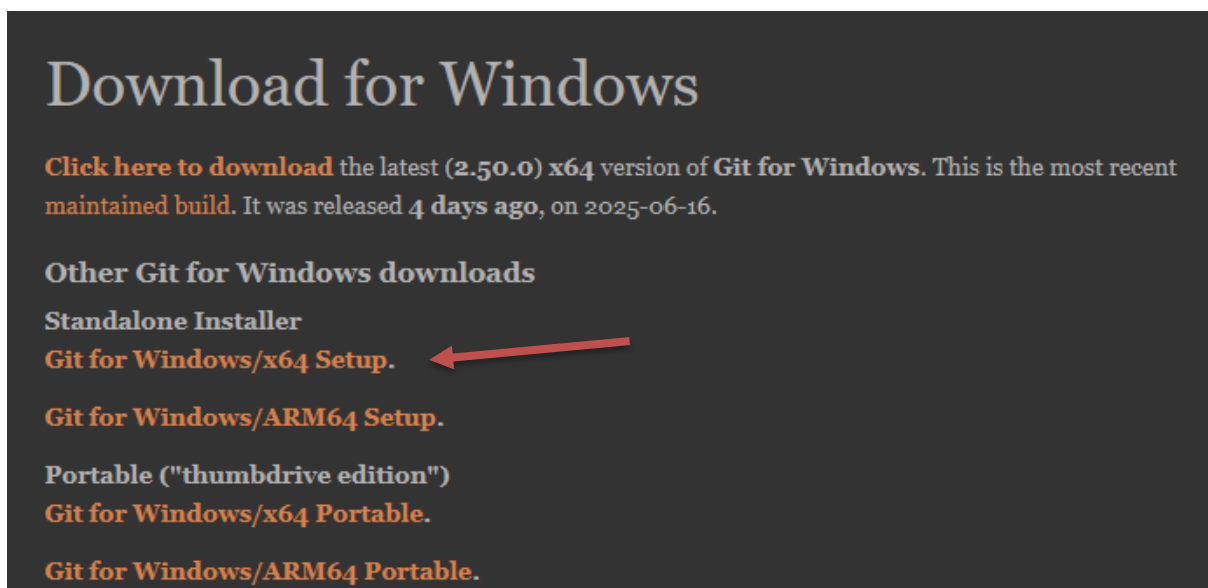


Imagen 1 – Descargar Git

Fuente: <https://git-scm.com>

2. Configuración de Git en el sistema

A través de Git Bash, configuraré mi identidad con los siguientes comandos:

- `git config --global user.name "Angie Juez"`
- `git config --global user.email lauradisanta@gmail.com`

Esto permite que los commits que realice se asocien correctamente al usuario de GitHub.



```
MINGW64/c/Users/Victor
Victor@Sveltedust$ MINGW64 ~
$ git config --global user.name "Angie Juez"
Victor@Sveltedust$ MINGW64 ~
$ git config --global user.email "lauradisanta@gmail.com"
Victor@Sveltedust$ MINGW64 ~
$
```

Imagen 2 – Configuración GitBash

Fuente: *Elaboración propia*

3. Inicio de sesión en GitHub

Se accedió a la cuenta de GitHub para autorizar la conexión entre Visual Studio Code y la plataforma.

4. Creación del repositorio desde Visual Studio Code

Desde Visual Studio Code, se creó un nuevo repositorio Git utilizando la opción "Initialize Repository". Luego, se añadió el proyecto al control de versiones con Git.

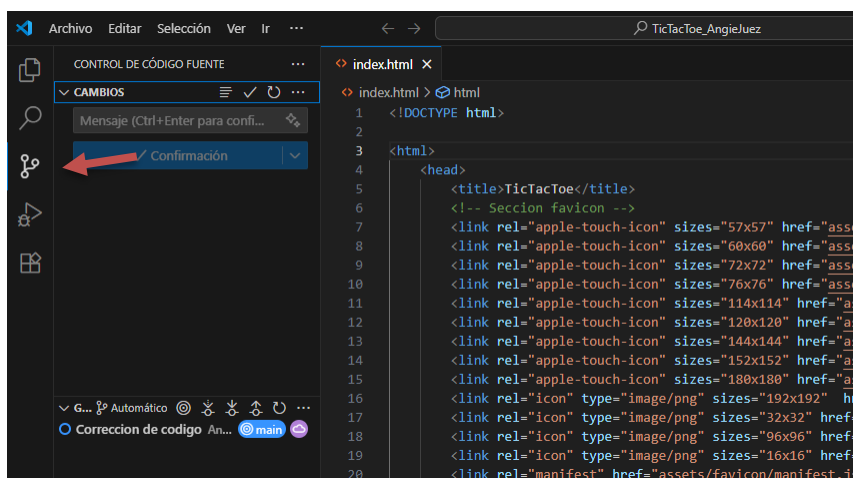


Imagen 3 – Inicializar repositorio

Fuente: <https://git-scm.com>

5. Vinculación del repositorio local con GitHub

En Visual Studio Code se usó la funcionalidad de "Publish to GitHub" para conectar el repositorio local con un repositorio remoto en GitHub. Esto permitió subir los archivos del proyecto directamente desde el editor al repositorio remoto, también se escoge la configuración, en este punto se elije a que rama se quiere subir el proyecto (main, master, etc) en mi caso, lo deje en AUTO esto por defecto deja el repositorio en la rama main , añadí un comentario en index.html para ver los cambios reflejados en el commit de GitHub.

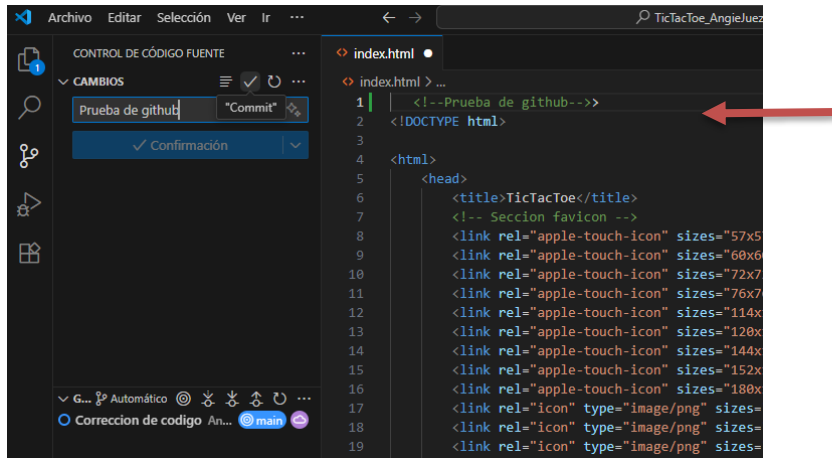


Imagen 4 – Creación de commit

Fuente: Elaboración propia

6. Sincronización de cambios

A partir de este punto, se pueden realizar commits y sincronizar los cambios entre el repositorio local y GitHub directamente desde Visual Studio Code, utilizando los botones integrados en la interfaz de control de versiones.

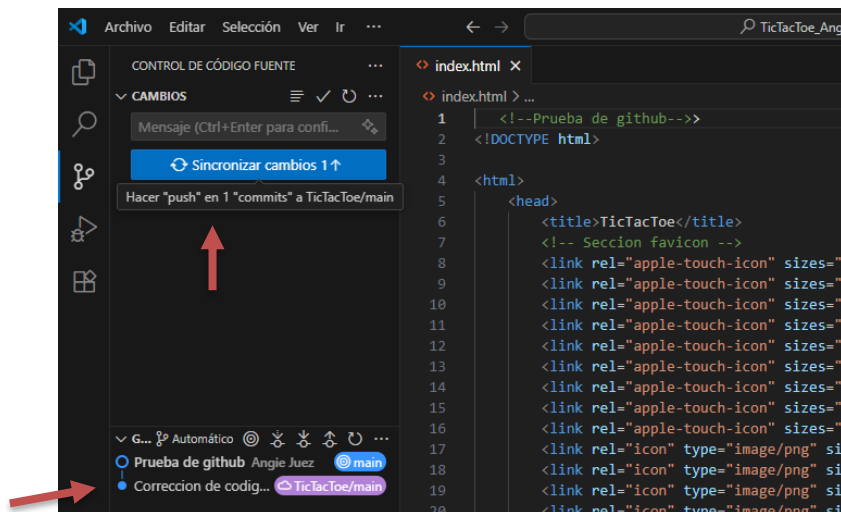


Imagen 5 – Sincronización GitHub

Fuente: Elaboración Propia

Repositorio de artefactos

Técnico en procesamiento de pruebas de software - SENA

Al hacer esto sincronizar los cambios ahora podemos verlos en nuestro repositorio de GitHub en el cual hice el comentario de prueba señalado en la imagen

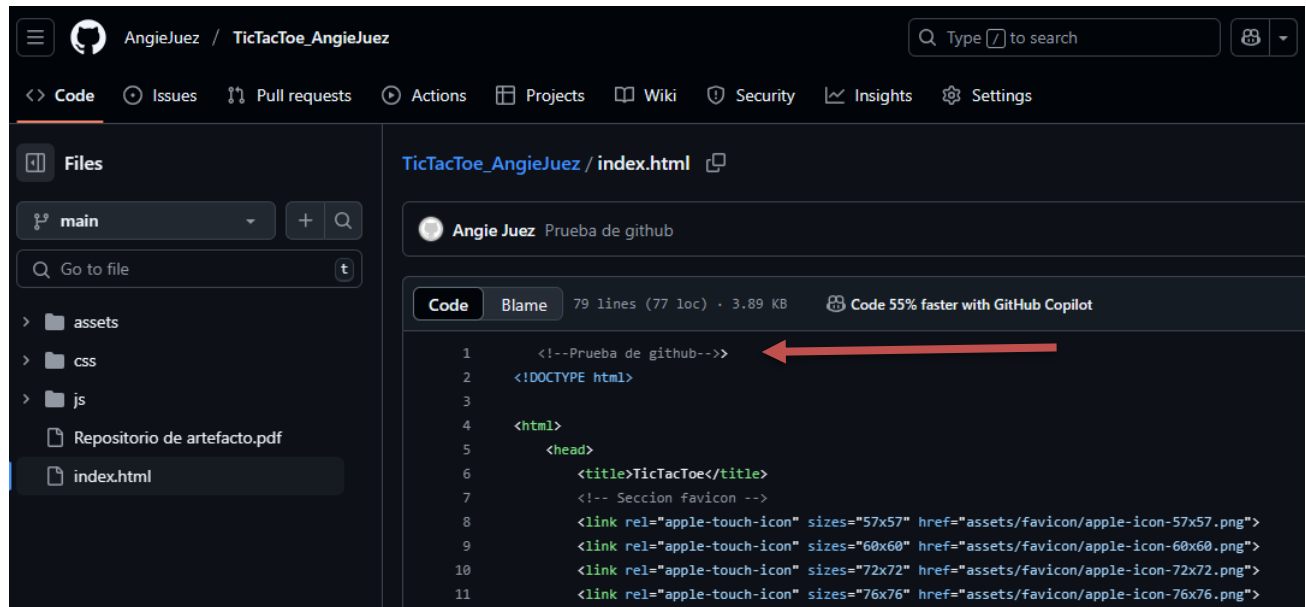


Imagen 6 – Cambios GitHub

Fuente: https://github.com/AngieJuez/TicTacToe_AngieJuez

Conclusión

La implementación de estas mejoras contribuirá a que el proyecto TicTacToe evolucione desde una solución funcional básica hacia una aplicación más robusta, escalable y socialmente inclusiva. La modularización del código facilita el mantenimiento, la optimización del CSS mejora el rendimiento visual en múltiples dispositivos, y la accesibilidad garantiza que el sistema pueda ser utilizado por un público más amplio. Estas acciones no solo anticipan futuros requerimientos, sino que también elevan el estándar de calidad del desarrollo, alineándolo con buenas prácticas del desarrollo frontend moderno.