

Informe Académico: Desarrollo y Compilación con Emscripten

José Luis Martínez Acevedo

24 de agosto de 2025

Resumen

En este informe se describe el proceso de creación de una aplicación sencilla en C integrada con WebAssembly (WASM), su compilación mediante Emscripten, la visualización en un entorno web y las herramientas adicionales empleadas en el flujo de trabajo, incluyendo Git, Git-Flow y plugins de Vim.

1. Introducción

El objetivo de este proyecto es comprender cómo se puede integrar código escrito en C con aplicaciones web utilizando WebAssembly, facilitando la ejecución de funciones matemáticas básicas directamente en el navegador.

2. Metodología

A continuación se resumen los pasos principales llevados a cabo:

2.1. Instalación de Emscripten

Se descargó e instaló el SDK de Emscripten:

```
git clone https://github.com/emscripten-core/emsdk.git
cd emsdk
./emsdk install latest
./emsdk activate latest
source ./emsdk_env.sh
```

2.2. Compilación del código en C

El código fuente (sum.c) define dos funciones: suma y multiplicación.

```
int sum(int a, int b){ return a+b; }
int multiply(int a, int b){ return a*b; }
```

Se compiló con:

```
emcc sum.c -o sum.js -s EXPORTED_FUNCTIONS="['_sum','_multiply']" \
-s EXPORTED_RUNTIME_METHODS="['ccall','cwrap']"
```

2.3. Activación de servidor local

Para probar la aplicación en el navegador se usó:

```
python3 -m http.server 8000
```

2.4. Uso de Git y Git-Flow

Se instaló Git-Flow:

```
sudo apt install git-flow
```

El flujo de ramas incluyó:

- **main**: versión estable.
- **develop**: integración de nuevas funciones.
- **feature/***: ramas de prueba características en este caso la prueba de poder multiplicar.

2.5. Prompts empleados

Durante el desarrollo se emplearon instrucciones en lenguaje natural para guiar el flujo de trabajo mediante chatgpt o5 basic. Algunos ejemplos son:

- “Crea un código en C que sume y multiplique dos números.”
- “Explicame cómo compilar con Emscripten y exportar funciones.”
- “Muéstrame cómo levantar un servidor local en Python para probar en el navegador.”
- “Explicame cómo usar Git-Flow para manejar ramas de features.”
- “Recomiéndame plugins de Vim para desarrollo en C y Git.”

2.6. Plugins de Vim

Se instalaron plugins para mejorar la productividad:

- **NERDTree**: explorador de archivos
- **vim-airline**: barra de estado mejorada
- **ale**: linting en tiempo real
- **vim-fugitive**: integración con Git

3. Resultados

La aplicación web permite ingresar dos números, realizar operaciones de suma y multiplicación, y visualizar los resultados directamente en la página. El estilo se mejoró con CSS para ofrecer un diseño más agradable en modo oscuro.

4. Conclusiones

Se logró comprender el flujo completo desde la creación del código en C hasta su ejecución en un navegador mediante WebAssembly. Asimismo, se exploraron buenas prácticas en el uso de Git-Flow y Vim como entornos de desarrollo.

5. Referencias

- Documentación oficial de Emscripten: <https://emscripten.org>
- Pandoc para conversión de documentos: <https://pandoc.org>
- Git-Flow cheatsheet: <https://danielkummer.github.io/git-flow-cheatsheet/>