

## WebAssembly

Durante muchos años se ha ido mejorando el rendimiento en la web, utilizando lenguajes como JavaScript y estándares como ECMAScript, sin embargo, todavía no se había encontrado algo que verdaderamente se viera la diferencia, hasta que llegó WebAssembly.

WebAssembly (WASM) es un estándar abierto que permite la ejecución de código binario en la web, proporcionando un nivel de rendimiento superior al rendimiento ofrecido por un lenguaje interpretado como JavaScript. Prácticamente es una mejora total a Javascript actual, ya que cuesta bastante tener módulos Javascript que no afecten mucho el rendimiento, pero ya teniendo algo que el navegador entienda como binario es muy potente ya que podemos compilar bundles que se comprimen más pequeños que el que hacemos con Javascript y correrlos aún más rápido.

Wasm es un lenguaje de bajo nivel, similar al lenguaje ensamblador, con un formato binario compacto que se ejecuta con rendimiento casi nativo y provee un objetivo de compilación para lenguajes como C/C++ y Rust que les permite correr en la web. Funciona perfectamente en solitario como a la par de JavaScript, permitiendo que ambos trabajen juntos.

WebAssembly dispone de 2 formatos: uno, binario compilado a un nivel cercano al de ensamblador y que puede ser generado desde otros lenguajes de programación y otro que es WebAssembly Text Format o WAT y que es la representación textual del código binario actuando como estado intermedio más amigable con los humanos.

Personalmente me llama bastante la atención el tema, ya que siempre me ha gustado bastante todo lo de web y siempre ha existido ese problema de los bundles de JavaScript, y con wasm de verdad que se ve el potencial y el futuro de la web. Wasm al ser un lenguaje bastante similar a ensamblador, de muy bajo nivel, me motiva a poder aprenderlo, ya que, en la materia se aprendió ensamblador y sería un buen punto de partida porque este maneja instrucciones y memoria similar a ensamblador. Al implementarlo junto con JavaScript ofrece una interacción y comunicación totalmente bidireccional donde desde JavaScript podemos llamar a módulos Wasm y beneficiarnos de su alto rendimiento y desde módulos Wasm podemos tener acceso a las APIs del navegador.