

# Desarrollo de un Compilador Genérico de Lenguaje Ensamblador para el Simulador CREATOR

Álvaro Guerrero Espinosa

Leganés, Julio 2025

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

- Importancia de conocer el funcionamiento general de un computador y el lenguaje ensamblador
  - Los lenguajes de alto nivel se construyen sobre los de bajo nivel
  - Sus capacidades y restricciones determinan el coste de las diferentes abstracciones
- Para su enseñanza se utilizan simuladores
  - Necesitan un ensamblador

# CREATOR

CREATOR 1.1 RISC-V (RV32IMFD)

disAsCto and generateAsmEntry progAsmSimulator

Feedback

Suggestions

Community

About us

Architecture

Assembly

Reset

Inst.

Run

Flash

Stop

Examples

Calculator

Configuration

Info

Break	Address	Label	User Instruction	Loaded Instructions
	0x0	<b>main</b>	addi sp sp -4	addi sp -4
	0x4		sw ra 0 (sp)	sw ra 0 (sp)
	0x8		li a0 5	addi a0 x0 5
	0xc		jal x1 factorial	jal x1 0x24
	0x10		li a7 1	addi a7 x0 1
	0x14		ecall	ecall
	0x18		lw ra 0 (sp)	lw ra 0 (sp)
	0x1c		addi sp sp 4	addi sp sp 4
	0x20		jr ra	jalr x0 0 (ra)
	0x24	<b>factorial</b>	addi sp sp -12	addi sp sp -12
	0x28		sw ra 8 (sp)	sw ra 8 (sp)
	0x2c		sw fp 4 (sp)	sw fp 4 (sp)
	0x30		addi fp sp 4	addi fp sp 4
	0x34		li x5 2	addi x5 x0 2
	0x38		bge a0 t0 b_else	bge a0 t0 2
	0x3c		li a0 1	addi a0 x0 1

INT/Ctrl Registers

FP Registers

Memory

Stats

CLK Cycles

Register value representation

Signed

Unsigned

Hex

Register name representation

Name

Alias

All

PC 44

x0 | zero

x1 | ra 16

x2 | sp 26043549...

x3 | gp

x4 | tp

x5 | t0

x6 | t1

x7 | t2

x8 | fp | s0

x9 | s1

x10 | a0 5

x11 | a1

x12 | a2

x13 | a3

x14 | a4

x15 | a5

x16 | a6

x17 | a7

x18 | s2

x19 | s3

x20 | s4

x21 | s5

x22 | s6

x23 | s7

x24 | s8

x25 | s9

x26 | s10

x27 | s11

x28 | t3

x29 | t4

x30 | t5

x31 | t6

Clear

Enter

<https://creatorsim.github.io/>

Desarrollar un ensamblador capaz de sustituir al actualmente utilizado por CREATOR corrigiendo sus problemas.

- Permitir funcionalidades avanzadas
- Mejorar los mensajes de error
- Organizar el código para facilitar su modificación

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

# Ensambladores

- Conversión de código ensamblador a una representación ejecutable
- Ejemplos: GNU Assembler (GAS), Tiny C Compiler Assembler (TCCASM), Netwide Assembler (NASM).

Ensamblador	GAS	TCCASM	NASM	CREATOR	Propuesta
Ejecución web				✓	✓
Biblioteca				✓	✓
Expresiones	✓*	✓*	✓*		✓
Etiquetas como valores	✓	✓	✓		✓
Bigints	✓				✓
UTF-8	✓	✓	✓		✓
Definición de constantes	✓		✓		✓**
Macros	✓		✓		✓**
Compilación condicional	✓		✓		✓**
Recuperación de errores	✓		✓		✓**

\* Solo posible con números enteros de tamaño fijo

\*\* Trabajo futuro



# Mensajes de error – Ensambladores

```
$ riscv64-unknown-elf-as -march=rv32i -o test.o test.s
test.s: Assembler messages:
test.s:3: Error: illegal operands `addi t0,t0,t1'
test.s:5: Error: unrecognized opcode `unknown t0,t0,10'
```

## GNU Assembler

```
$ tcc test.s
test.s:5: error: bad operand with opcode 'addl'
```

## Tiny C Compiler Assembler

```
$ nasm test.s
test.s:5: error: invalid combination of opcode and operands
test.s:7: error: invalid combination of opcode and operands
```

## Netwide Assembler

# Mensajes de error – Compiladores

```
Compiling tfg_test v0.1.0 (/home/alvaro/Documentos/temp/tfg_test)
error[E0373]: closure may outlive the current function, but it borrows `name`, which is owned by the current function
--> src/main.rs:13:30

13 |         self.0.iter().filter(|&inst| inst.name == name)
    |                               ~~~~~
    |                               |
    |                               ---- `name` is borrowed here
    |                               |
    |                               may outlive borrowed value `name`

note: closure is returned here
--> src/main.rs:13:9

13 |         self.0.iter().filter(|&inst| inst.name == name)
    |                               ~~~~~
help: to force the closure to take ownership of `name` (and any other referenced variables), use the `move` keyword

13 |         self.0.iter().filter(move |&inst| inst.name == name)
    |                               ++++
```

For more information about this error, try ``rustc --explain E0373``.

error: could not compile `tfg\_test` (bin "tfg\_test") due to 1 previous error

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis**
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

- 19 Requisitos de usuario
  - 14 de capacidad
  - 5 de restricción
- 40 Requisitos de software
  - 28 funcionales
  - 12 no funcionales

---

## RS-FN-08

---

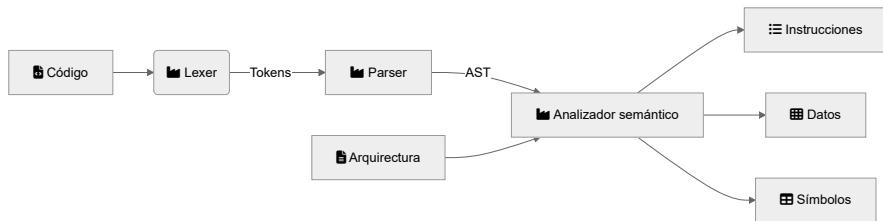
<b>Descripción</b>	El sistema debe soportar directivas de datos de números decimales con el formato IEEE 745 [48] de precisión simple (binary32) y doble (binary64).
<b>Necesidad</b>	Esencial
<b>Prioridad</b>	Alta
<b>Estabilidad</b>	No cambia
<b>Verificabilidad</b>	Alta
<b>Origen</b>	RU-CA-01

---

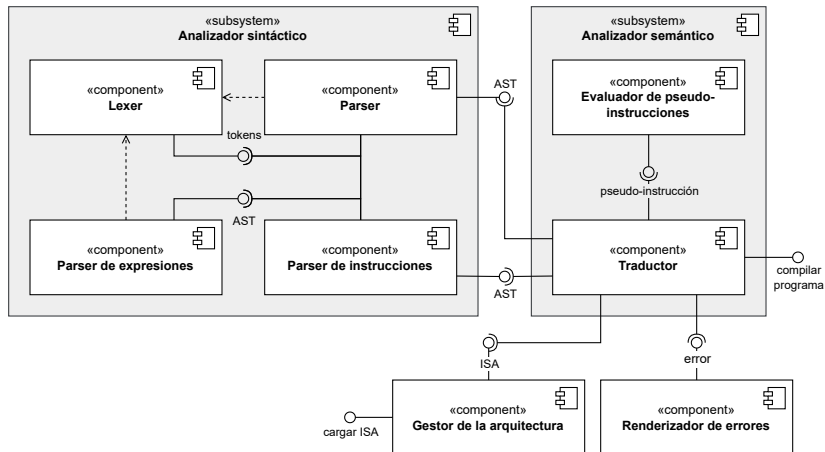
# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño**
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

# Funcionamiento



# Arquitectura



- Diseño modular
- Analizador sintáctico y semántico independientes
  - AST como representación intermedia
- Compilación de múltiples pasadas
- Rust y WebAssembly con una interfaz para JS



# Mensajes de error

```
$ ./creator.sh -a RISC_V_RV32IMFD.json -s 1.s -o min --color
[E11] Error: Label repeated is already defined
[ assembly:10:1 ]
3  repeated: li t0, 0
   └── Note: Label also defined here
10 repeated: add t2, t0, t1
   └── Duplicate label
Help: Consider renaming either of the labels

Not executed
keyboard[0x0]:''; display[0x0]:'';
```

```
$ ./creator.sh -a RISC_V_RV32IMFD.json -s 3.s -o min --color
[E22] Error: Value 65536 is outside of the valid range of the field
[ assembly:2:7 ]
2  .half 0x10000
   └── This expression has value 65536
Note: Allowed range is [-32768, 65535]

Not executed
keyboard[0x0]:''; display[0x0]:'';
```

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación**
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros

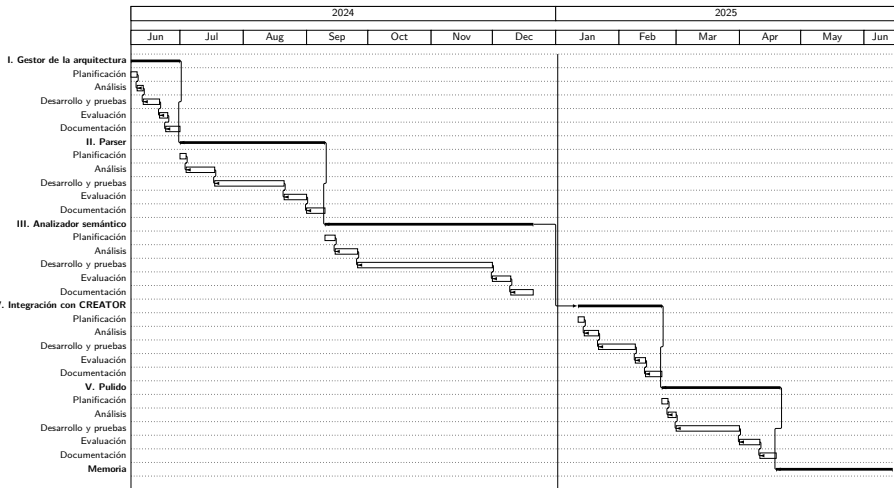
# Estructura de ficheros

```
/
├── js_example/ ..... Ejemplos de uso
├── src/
│   ├── architecture/ ..... Deserialización de la arquitectura
│   ├── compiler/ ..... Subsistema analizador semántico
│   ├── parser/ ..... Subsistema analizador sintáctico
│   ├── architecture.rs ..... Gestor de la arquitectura
│   ├── compiler.rs ..... Traductor
│   ├── error_rendering.rs ..... Renderizador de errores
│   ├── js.rs ..... API JS
│   ├── lib.rs ..... Biblioteca del compilador
│   ├── parser.rs ..... Parser
│   └── span.rs ..... Región de código
├── tests/ ..... Arquitecturas de prueba
├── Cargo.lock ..... Versiones de dependencias
├── Cargo.toml ..... Metadatos de la biblioteca
├── LICENSE
├── README.md
└── build.sh ..... Compilación biblioteca
```

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto**
- 7 Conclusiones y trabajos futuros

# Tiempo estimado



Personal	28.350,00 €
Equipamiento	169,32 €
Costes indirectos	306,28 €
<b>Total</b>	<b>28.825,60 €</b>

- Grandes empresas están empezando a diseñar procesadores propios
  - Necesitan ingenieros con conocimiento de hardware y ensamblador
- Usado por múltiples universidades españolas e internacionales

# Tabla de Contenidos

- 1 Introducción
- 2 Estado del arte
- 3 Análisis
- 4 Diseño
- 5 Implementación
- 6 Plan del proyecto
- 7 Conclusiones y trabajos futuros**



- Ventajas del nuevo ensamblador
  - Mejores mensajes de error
  - Nuevas funcionalidades
  - Código estructurado y flexible
- Conocimientos adquiridos
  - Diseño e implementación de compiladores y ensambladores
  - Creación de interfaces entre lenguajes
  - Programación en Rust

- Más funcionalidades
  - Macros
  - Definición de constantes
  - Compilación condicional
- Estrategias de recuperación de errores
- Mejoras en la implementación
  - Mejoras en la evaluación de expresiones
  - Mejoras del *parser*

- Tabla de punteros