Tecnológico de Monterrey

Proyecto 4. Generador de Código

TC3002B: Desarrollo de Aplicaciones Avanzadas de Ciencias Computacionales

Grupo: 501

Alumnos:

Alan Anthony Hernandez Perez | A01783347

Alejandro Fernandez | A01024998

Campus:

Santa Fe

Profesores:

Victor Manuel de la Cueva Hernandez

21 de mayo de 2025

Introducción de MIPS

En este proyecto se genera código intermedio en lenguaje ensamblador **MIPS (Microprocessor without Interlocked Pipeline Stages)**. MIPS es una arquitectura de tipo RISC (Reduced Instruction Set Computer), lo que significa que utiliza un conjunto reducido de instrucciones simples y eficientes. Cada instrucción suele ejecutarse en un solo ciclo de reloj, lo que permite un procesamiento rápido y predecible.

El uso de MIPS en este contexto se debe a varias razones: su simplicidad facilita el aprendizaje del proceso de generación de código, su estructura clara permite una implementación más ordenada del traductor, y existen herramientas como el simulador **QtSPIM** que permiten ejecutar y depurar programas de forma sencilla y actualizada. Aunque inicialmente se consideró el uso de MARS, este presentaba problemas de compatibilidad con versiones recientes del sistema, por lo que se optó por **QtSPIM**, que reconoce correctamente el formato del archivo generado y ofrece una ejecución estable del código MIPS.

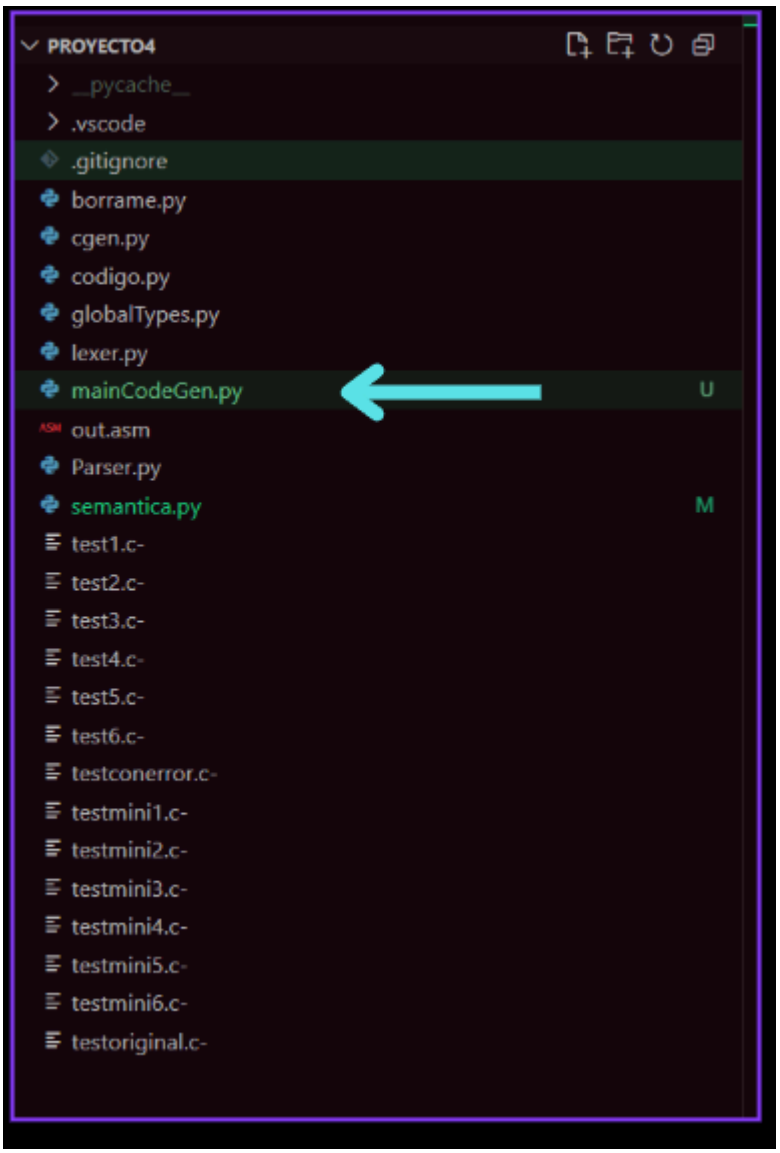
Por estas razones, MIPS resulta una opción adecuada para implementar traductores de lenguajes de alto nivel en entornos académicos.

Manual de usuario

Descarga y descompresión
Descargue el archivo .zip proporcionado y descomprimirlo en una ubicación de su preferencia.

Ubicación del archivo principal
Dentro de la carpeta descomprimida, localice el archivo llamado **mainCodeGen.py**. Este archivo es el encargado de ejecutar el proceso completo de generación de código.

Para editar o ejecutar este archivo, se recomienda utilizar un entorno de desarrollo como Visual Studio Code (**VSCode**), PyCharm o cualquier otro editor de texto compatible con Python. Asegúrese de tener Python instalado correctamente en su sistema para poder ejecutar el archivo sin problemas.



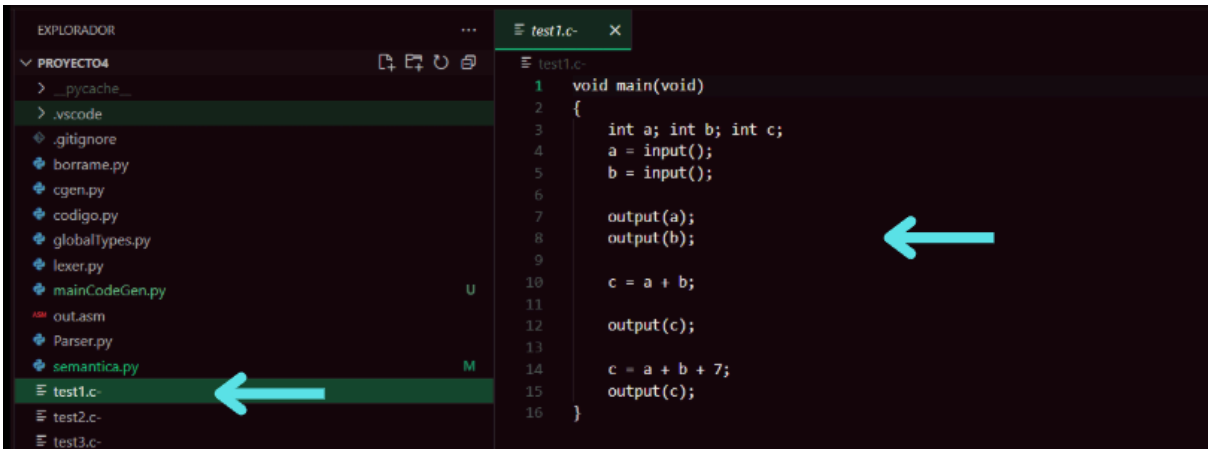
Funcionamiento general

Al ejecutar mainCodeGen.py, el programa realiza los siguientes pasos:

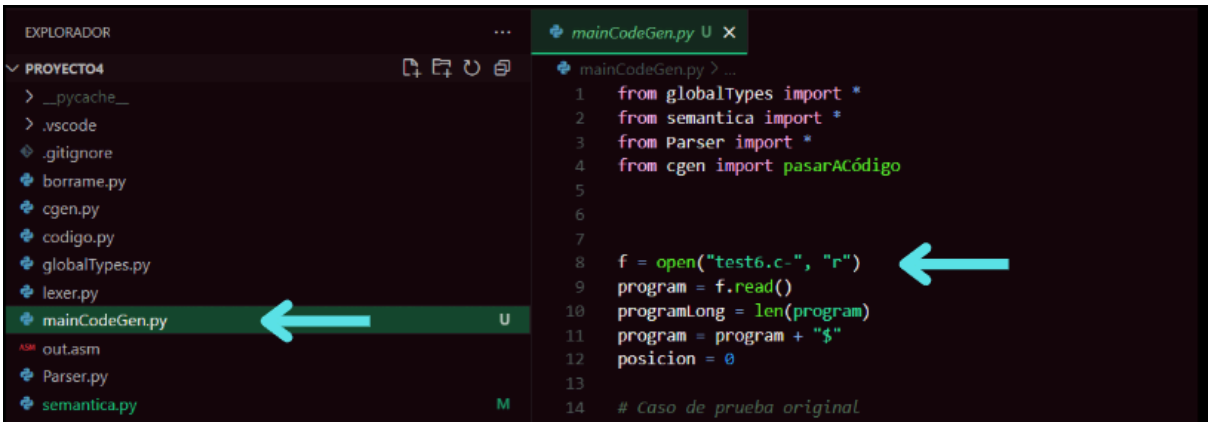
- Revisión de léxico.
- Genera el Árbol de Sintaxis Abstracto (AST).
- Revisa la semántica sobre el código fuente.
- Si se detecta algún error de léxico, sintaxis o semántica, el sistema indicará el tipo de error y la línea correspondiente en el código fuente en C-, y **no se generará el código MIPS** hasta que se corrijan dichos errores.

Uso con un archivo personalizado
Por defecto, el programa ya tiene cargado un archivo de prueba. Si desea probar con su propio código, siga estos pasos:

- Cree un archivo de texto con extensión .c-, que contenga el código en lenguaje C-.



- Coloque ese archivo dentro de la misma carpeta donde se encuentra mainCodeGen.py.
- Abra el archivo mainCodeGen.py y diríjase a la línea 8. Reemplace el nombre del archivo actual (primer parámetro de la función open) por el nombre de su propio archivo.

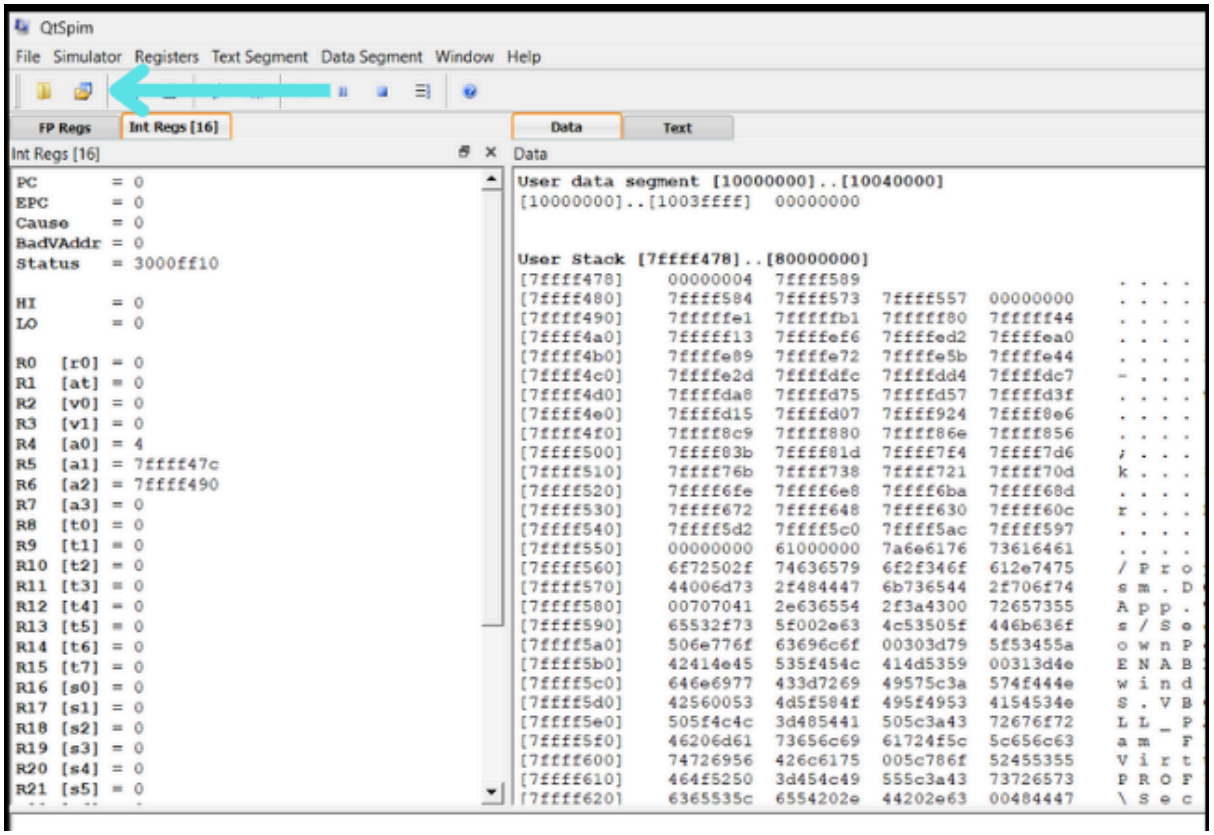


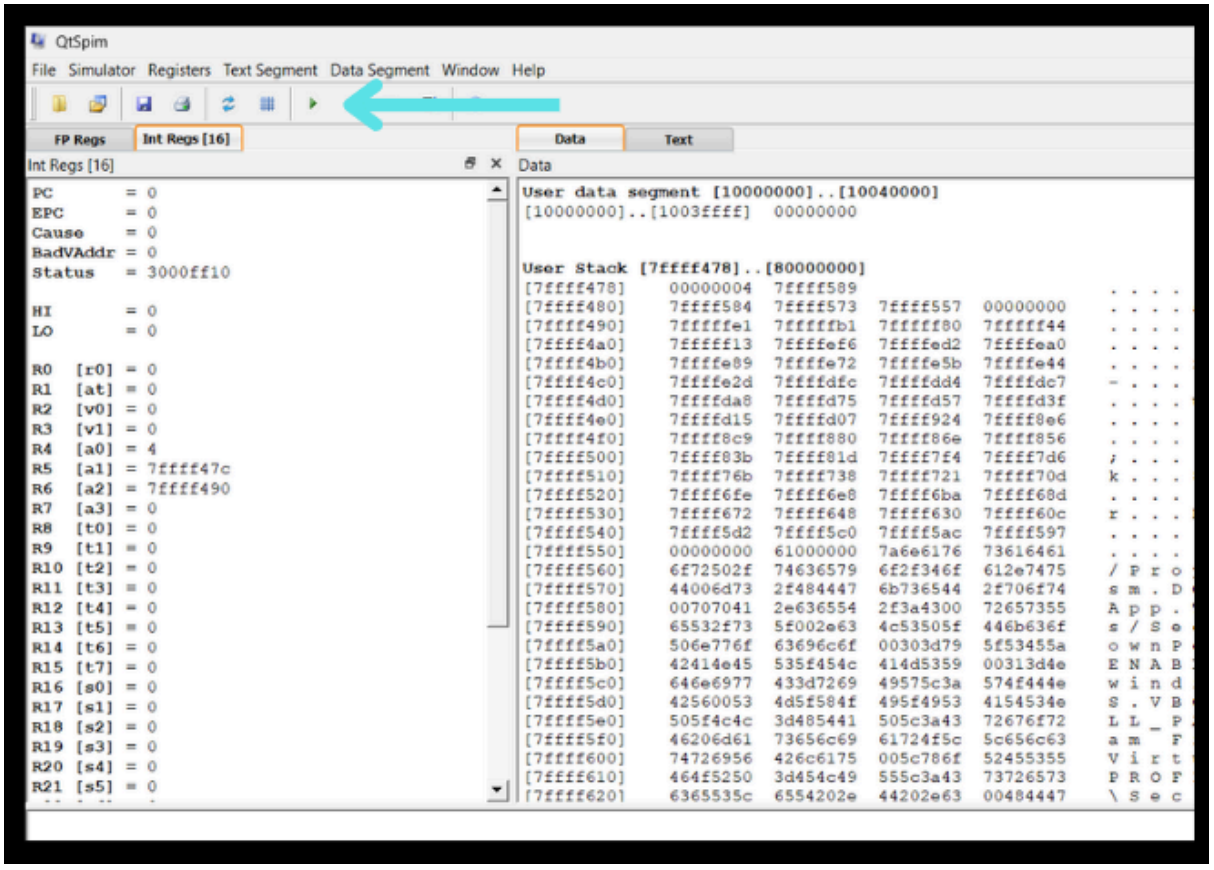
Ejecución y simulación en QTSPIM

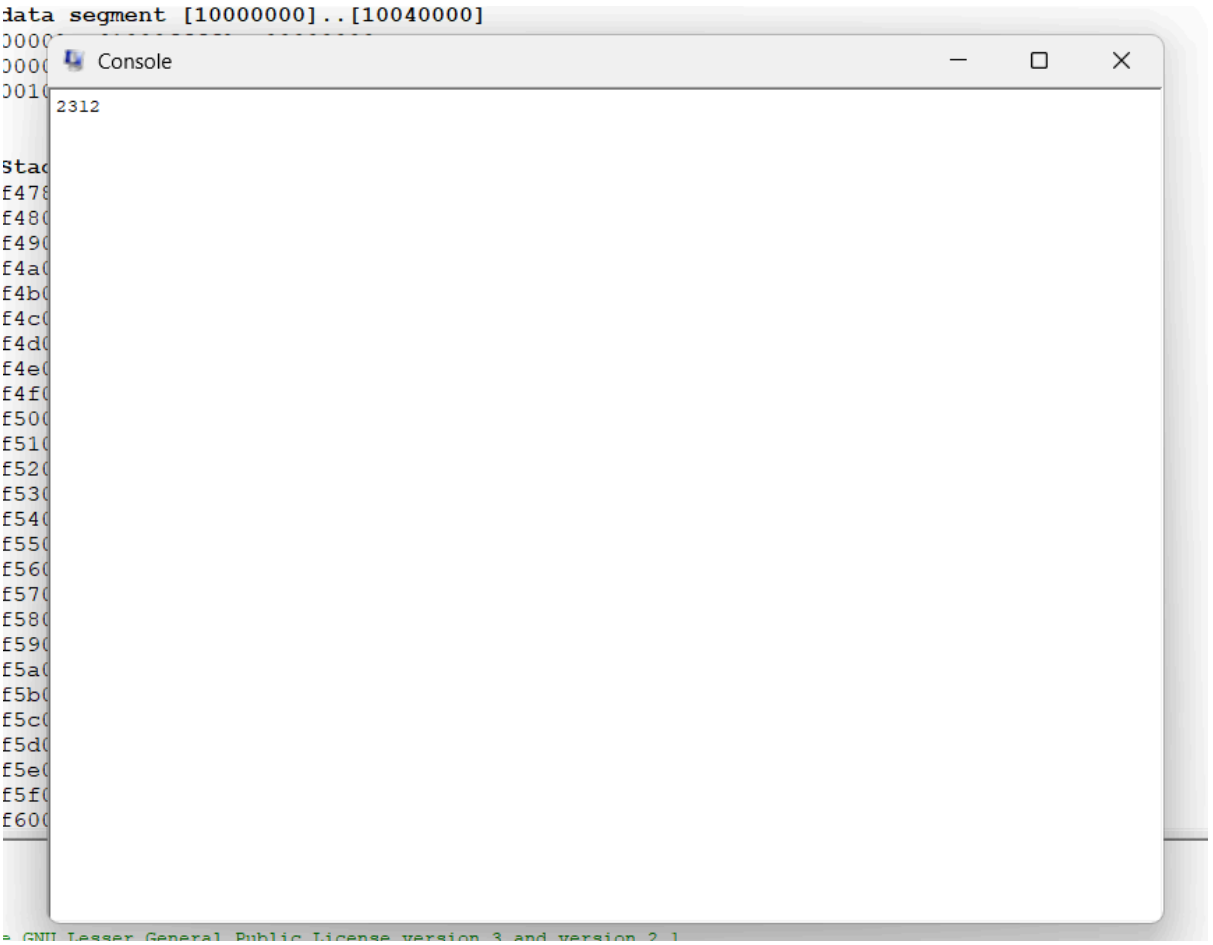
- Ejecute mainCodeGen.py. Si no hay errores, se generará automáticamente un archivo llamado out.asm.

- Abra **QTSPIM**.

- Cargue el archivo out.asm, ensamble y ejecútalo para observar el comportamiento del código traducido.


QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Data
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 4
R5 [a1] = 7ffff47c
R6 [a2] = 7ffff490
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000
User Stack [7ffff478]..[80000000]
[7ffff478] 00000004 7ffff589
[7ffff480] 7ffff584 7ffff573 7ffff557 00000000
[7ffff490] 7fffffe1 7fffffb1 7fffff80 7fffff44
[7ffff4a0] 7fffff13 7ffffef6 7ffffed2 7ffffea0
[7ffff4b0] 7ffffe89 7ffffe72 7ffffe5b 7ffffe44
[7ffff4c0] 7ffffe2d 7ffffdfc 7ffffdd4 7ffffdc7
[7ffff4d0] 7ffffda8 7ffffd75 7ffffd57 7ffffd3f
[7ffff4e0] 7ffffd15 7ffffd07 7ffff924 7ffff8e6
[7ffff4f0] 7ffff8c9 7ffff880 7ffff86e 7ffff856
[7ffff500] 7ffff83b 7ffff81d 7ffff7f4 7ffff7d6
[7ffff510] 7ffff76b 7ffff738 7ffff721 7ffff70d
[7ffff520] 7ffff6fe 7ffff6e8 7ffff6ba 7ffff68d
[7ffff530] 7ffff672 7ffff648 7ffff630 7ffff60c
[7ffff540] 7ffff5d2 7ffff5c0 7ffff5ac 7ffff597
[7ffff550] 00000000 61000000 7a6e6176 73616461
[7ffff560] 6f72502f 74636579 6f2f346f 612e7475
[7ffff570] 44006d73 2f484447 6b736544 2f706f74
[7ffff580] 00707041 2e636554 2f3a4300 72657355
[7ffff590] 65532f73 5f002e63 4c53505f 446b636f
[7ffff5a0] 506e776f 63696c6f 00303d79 5f53455a
[7ffff5b0] 42414e45 535f454c 414d5359 00313d4e
[7ffff5c0] 646e6977 433d7269 49575c3a 574f444e
[7ffff5d0] 42560053 4d5f584f 495f4953 4154534e
[7ffff5e0] 505f4c4c 3d485441 505c3a43 72676f72
[7ffff5f0] 46206d61 73656c69 61724f5c 5c656c63
[7ffff600] 74726956 426c6175 005c786f 52455355
[7ffff610] 464f5250 3d454c49 555c3a43 73726573
[7ffff620] 6365535c 6554202e 44202e63 00484447


QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Data
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 4
R5 [a1] = 7ffff47c
R6 [a2] = 7ffff490
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000
User Stack [7ffff478]..[80000000]
[7ffff478] 00000004 7ffff589
[7ffff480] 7ffff584 7ffff573 7ffff557 00000000
[7ffff490] 7fffffe1 7fffffb1 7fffff80 7fffff44
[7ffff4a0] 7fffff13 7ffffef6 7ffffed2 7ffffea0
[7ffff4b0] 7ffffe89 7ffffe72 7ffffe5b 7ffffe44
[7ffff4c0] 7ffffe2d 7ffffdfc 7ffffdd4 7ffffdc7
[7ffff4d0] 7ffffda8 7ffffd75 7ffffd57 7ffffd3f
[7ffff4e0] 7ffffd15 7ffffd07 7ffff924 7ffff8e6
[7ffff4f0] 7ffff8c9 7ffff880 7ffff86e 7ffff856
[7ffff500] 7ffff83b 7ffff81d 7ffff7f4 7ffff7d6
[7ffff510] 7ffff76b 7ffff738 7ffff721 7ffff70d
[7ffff520] 7ffff6fe 7ffff6e8 7ffff6ba 7ffff68d
[7ffff530] 7ffff672 7ffff648 7ffff630 7ffff60c
[7ffff540] 7ffff5d2 7ffff5c0 7ffff5ac 7ffff597
[7ffff550] 00000000 61000000 7a6e6176 73616461
[7ffff560] 6f72502f 74636579 6f2f346f 612e7475
[7ffff570] 44006d73 2f484447 6b736544 2f706f74
[7ffff580] 00707041 2e636554 2f3a4300 72657355
[7ffff590] 65532f73 5f002e63 4c53505f 446b636f
[7ffff5a0] 506e776f 63696c6f 00303d79 5f53455a
[7ffff5b0] 42414e45 535f454c 414d5359 00313d4e
[7ffff5c0] 646e6977 433d7269 49575c3a 574f444e
[7ffff5d0] 42560053 4d5f584f 495f4953 4154534e
[7ffff5e0] 505f4c4c 3d485441 505c3a43 72676f72
[7ffff5f0] 46206d61 73656c69 61724f5c 5c656c63
[7ffff600] 74726956 426c6175 005c786f 52455355
[7ffff610] 464f5250 3d454c49 555c3a43 73726573
[7ffff620] 6365535c 6554202e 44202e63 00484447



Apéndice

Lexer: Proyecto1 lexer

Parser: EBNF C-

Semántica: REGLAS de semantica