

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)

Curso: CE5302- Proyecto de Diseño de Ingeniería en Computadores

(Course: CE5302 Computer Engineering Design Project)



Documento de diseño de software: Herramienta de perfilado para una arquitectura basada en RISC V

(Software design document: Profiling Tool for an architecture based on RISC-V)

Realizado por:

(Made by:)

Arturo Salas Delgado, 201229804

Profesor:

(Professor)

Gustavo Cubas Euceda

Fecha: Cartago, 9 de Marzo, 2018

(Date: Cartago, March 9, 2018)

Índice

Revisión del documento	3
Introducción	4
Propósito	4
Alcance	4
Contexto	4
Resumen	5
Referencias	5
Glosario	5
Vistas de diseño	6
Vista de composición	6
Vista Lógica	7
Vista de dependencias	7
Vista de datos	8
Vista de patrones de diseño	8
Vista de interfaces	8
Vista de estructura	9
Vista de interacción	10
Vista dinámica de estados	11
Vista de algoritmos	11
Vista de recursos	12

1. Revisión del documento

- **Fecha de versión y estatus**

Versión 1.0 finalizada el 7 de Marzo de 2018

- **Organización**

Instituto Tecnológico de Costa Rica. Laboratorio de Diseño de Circuitos Integrados (DCI Lab) en conjunto con el Laboratorio de Computación de Alto Rendimiento (HPC Lab). Escuela de Ingeniería Electrónica.

- **Autor**

Arturo Salas Delgado

- **Historia de cambios**

En la siguiente tabla se muestra el historial de cambios del documento de diseño, donde se establece la versión en la que fue creada, una descripción de lo que se hizo y la fecha en que se trabajó sobre eso.

Versión	Descripción	Fecha
1.0	Se crea la introducción del documento.	3 de Marzo de 2018
1.0	Se crea el glosario del documento. Además, se trabaja en la vista de composición y vista lógica.	5 de Marzo de 2018
1.0	Se trabaja en la vista de dependencias, vista de datos, vista de patrones de diseño y vista de estructura.	6 de Marzo de 2018
1.0	Se trabaja en la vista de interacción, en la vista dinámica de estados, en la vista de algoritmos y en la vista de recursos.	7 de Marzo de 2018

Tabla 1. Historial de cambios del documento de diseño.

2. Introducción

2.1. Propósito

Este documento muestra todos los aspectos de diseño que se necesitan para el desarrollo del proyecto “Herramienta de perfilado para una arquitectura basada en RISC V”.

La escuela de Ingeniería Electrónica, junto con el área de Ingeniería en Computadores e Ingeniería en Mecatrónica trabajan sobre la creación de distintos bloques de hardware, de manera personalizada, para la realización de tareas complejas. Es acá donde se pretende crear esta herramienta de manera libre, bajo términos de OpenSource, para que se pueda utilizar en la recopilación de estadísticas de un procesador pipeline basado en el set de instrucciones RISC V versión RV32I.

2.2. Alcance

El proyecto es limitado para la escuela de Ingeniería Electrónica y las áreas de Ingeniería en Computadores e Ingeniería en Mecatrónica, específicamente para ser utilizado en laboratorios o para futuras generaciones que requieran adiciones o modificaciones en la herramienta.

2.3. Contexto

Investigadores y estudiantes de la escuela de Ingeniería Electrónica, junto con las áreas de Ingeniería en Computadores e Ingeniería en Mecatrónica, colaboran con científicos del Departamento de Neurociencia del Centro Médico Erasmus, en Rotterdam, Países Bajos, para el desarrollo de dispositivos electrónicos que apoyen la creación de sustitutos para zonas dañadas del cerebro humano.

Es por esto que los ingenieros involucrados trabajan para la resolución de múltiples problemas en la utilización de dispositivos de hardware que emulen el proceso cognitivo. Se utilizan FPGAs (Field Programmable Gate Array) para la creación de distintos bloques de hardware, de manera personalizada, para la realización de tareas complejas. De acá surge la necesidad de la creación de un nuevo procesador basado en la arquitectura RISC V para la ejecución de programas escritos en código C estándar.

Consigno, los diseños de arquitecturas de procesadores y los ISAs (Instruction Set Architecture) en su mayoría son de carácter privativo y es necesario de un pago monetario para poder ser utilizados. En el año 2010 se comenzó con una iniciativa Open Source que consiste en el desarrollo de un ISA, para que la persona que lo desee lo pueda utilizar con fines ingenieriles; esta herramienta fue llamada con el nombre de RISC V.

El conjunto de Instrucciones de RISC V a pesar de ser OpenSource, tener herramientas para la compilación de código y de varios simuladores, no cuenta con hardware libre para obtener estadísticas o perfilados sobre algún procesador con dicha arquitectura, es por esto

que se ve en la necesidad de emplear conocimientos ingenieriles para el desarrollo de esta nueva herramienta de perfilado basada en la arquitectura RISC V.

En general, este proyecto busca brindar una herramienta sencilla para la generación de estadísticas, específicamente tiempos de ejecución, para una microarquitectura predefinida y para brindar una mejora en su rendimiento al momento de utilizar un conjunto de instrucciones RISC V.

2.4. Resumen

El producto consta de tres etapas: se encarga de la generación de código binario a partir de un programa escrito en C estándar, la optimización de ese código binario y la generación de estadísticas sobre un procesador basado en la arquitectura RISC V ejecutando ese programa.

3. Referencias

- Documento de diseño basado en las normas establecidas por el estándar *IEEE 1016 Standard for Information Technology - System Design - Software Design Description, 2009, clause 4, appendix C.*

4. Glosario

A continuación, se muestran las siglas y abreviaturas utilizadas en este documento:

- **RISC V.** Consiste en un set de instrucciones y procesadores abiertos a la comunidad para su libre uso. Es un proyecto *Open Source*.
- **FPGA.** *Field Programmable Gate Array*, por sus siglas en inglés, es un dispositivo donde se puede cargar descripciones de hardware mediante diferentes protocolos de transmisión de datos.
- **RV32I.** Versión 2.2 del conjunto de instrucciones RISC V.
- **GNU.** Es un sistema operativo de software libre de tipo Unix.

5. Vistas de diseño

A continuación, se muestra una serie de vistas en las cuáles es definido el proyecto “Herramienta de perfilado para una arquitectura basada en RISC V”, y donde se muestra el comportamiento del sistema a desarrollar. No todas las vistas aplican, por lo que se especifica explícitamente cuáles vistas aplican y cuáles no para el diseño del proyecto.

5.1. Vista de composición

La vista de composición describe los roles de las partes que conforman el sistema en general.

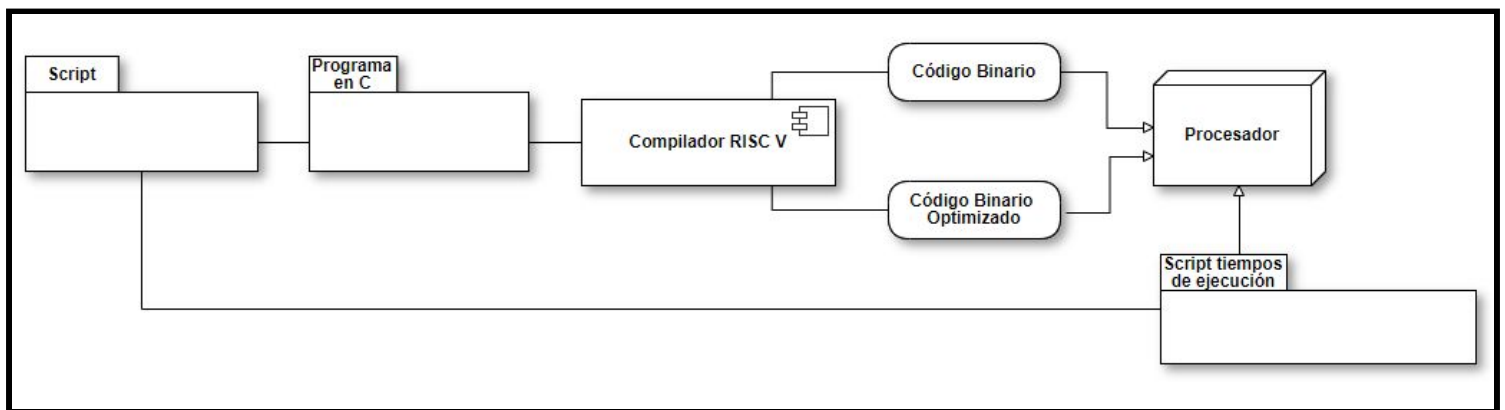


Figura 1. Vista de composición.

A continuación, se detalla cada una de las partes de la vista de composición:

- **Script**: corresponde a una interfaz de usuario mediante la terminal de linux, donde se elige lo que se desea hacer, ya sea crear un nuevo programa optimizado o un nuevo programa sin optimización y seleccionar si se desea obtener estadísticas de tiempos de ejecución sobre el procesador basado en RISC V.
- **Programa en C**: corresponde al módulo donde se crea un nuevo programa en C estándar para ser ejecutado por el procesador basado en RISC V. Este módulo de programación soporta instrucciones básicas como sumas, restas multiplicaciones y divisiones.
- **Compilador RISC V**: es un módulo que se debe integrar al sistema con la finalidad de generar código binario a partir del programa en C escrito por el usuario.
- **Código binario**: es el código binario generado por el sistema sin optimizaciones sobre el uso de coma flotante.
- **Código binario optimizado**: es el código binario generado por el sistema con optimización sobre el uso de coma flotante.
- **Procesador**: corresponde a un procesador existente basado en el conjunto de instrucciones RISC V, que ejecutará el código binario generado por el sistema.
- **Script tiempos ejecución**: corresponde a un módulo de software que verifica el tiempo de ejecución de los binarios generados sobre el procesador. Este módulo genera un archivo de lectura con las estadísticas obtenidas.

5.2. Vista Lógica

Esta vista define las implementaciones de las interfaces y clases con las distintas relaciones. La vista es la siguiente:

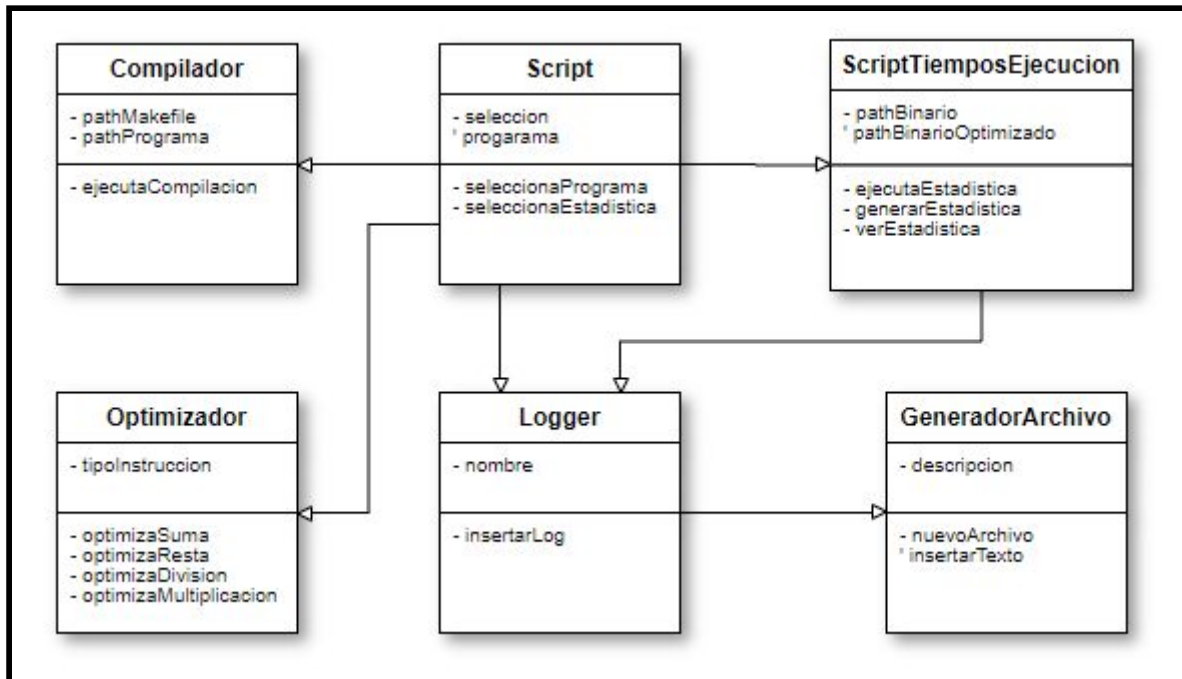


Figura 2. Vista lógica.

5.3. Vista de dependencias

Esta vista muestra las relaciones que existen entre entidades, además, define las clases que utiliza cada entidad o si se necesita de otra clase que tiene el sistema.

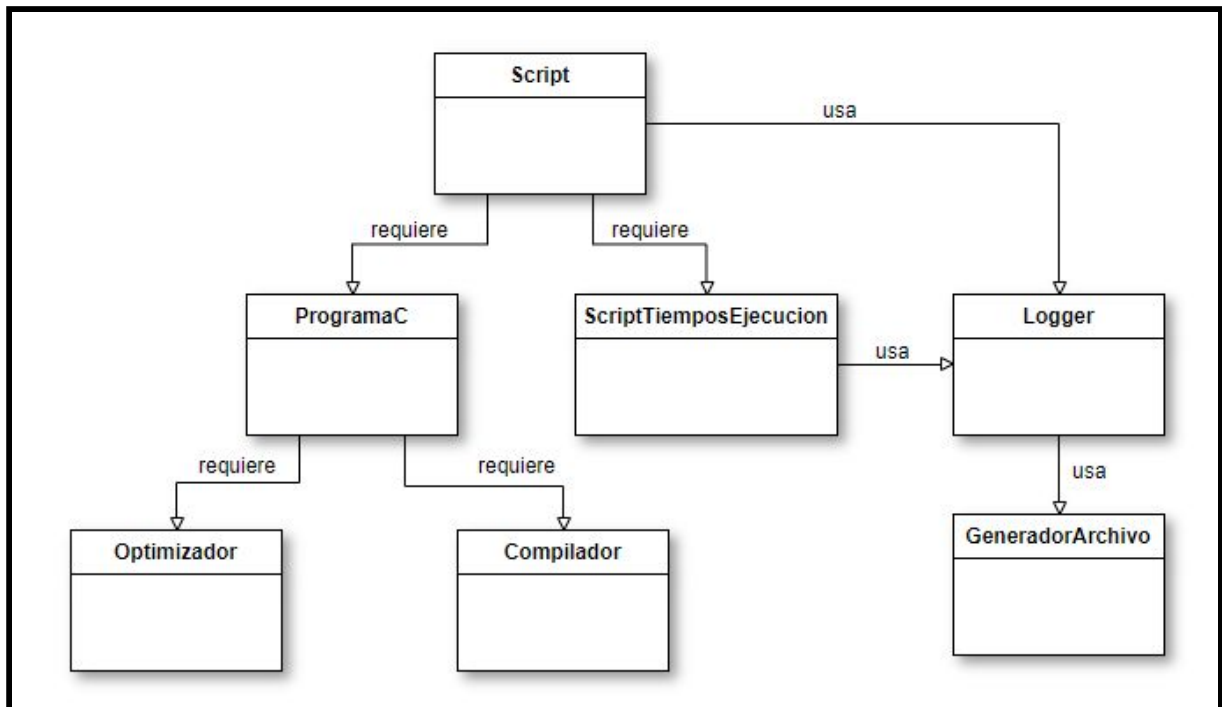


Figura 3. Vista de dependencias.

5.4. Vista de datos

No aplica para el diseño de este proyecto.

5.5. Vista de patrones de diseño

No aplica para el diseño de este proyecto.

5.6. Vista de interfaces

Esta vista describe la relación entre el sistema con otros sistemas. Para el diseño de la herramienta de perfilado se requiere de una interfaz con un procesador basado en el conjunto de instrucciones RISC V versión RV32I, por lo cual debe existir un medio de comunicación entre ambos. El archivo de salida de la herramienta de perfilado (código binario) va hacia el procesador para ser ejecutado. La vista se muestra a continuación:

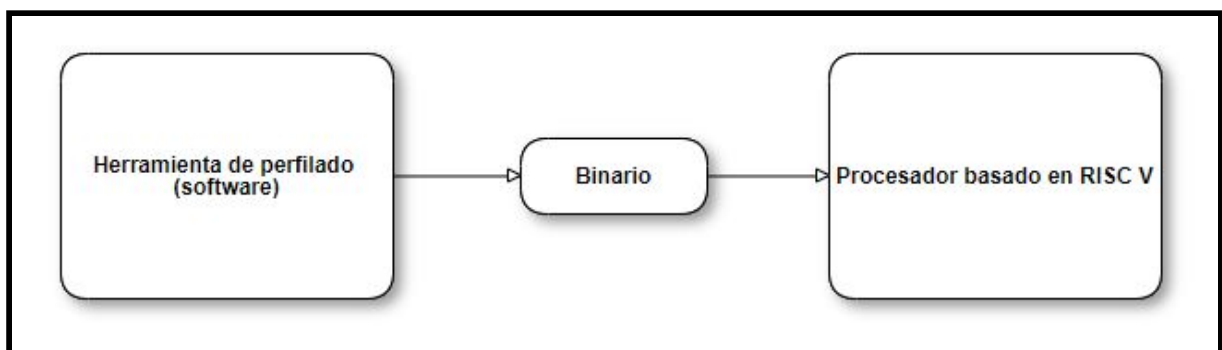


Figura 4. Vista de interfaces.

5.7. Vista de estructura

Esta vista muestra la estructura interna de todas las clases y componentes del sistema en términos de sistemas más pequeños. De esta manera, el sistema puede ser descompuesto en módulos, por lo que se facilita su implementación. La vista se muestra a continuación:

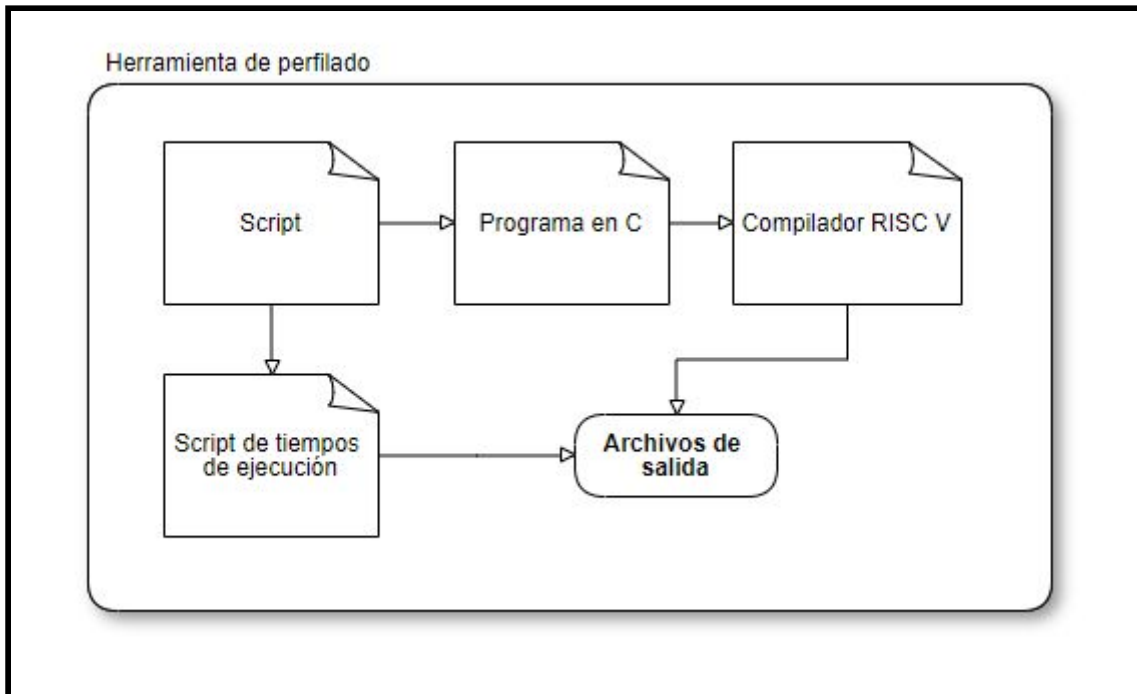


Figura 5. Vista de estructura.

La vista de cada uno de los componentes de la figura anterior se muestran a continuación:

- Script: proporciona al usuario la capacidad de seleccionar entre crear un nuevo programa o seleccionar uno existente para ser ejecutado en el procesador basado en RISC V.

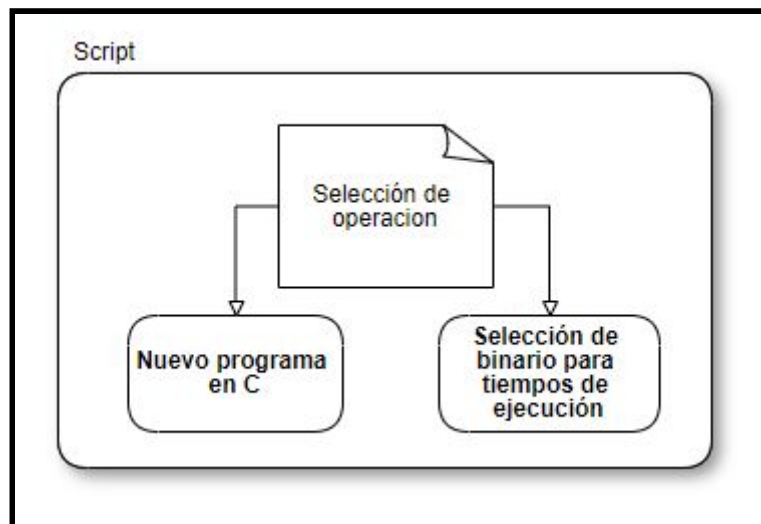


Figura 6. Estructura de Script.

- Programa en C: el sistema habilita la opción de que el usuario pueda ingresar un nuevo programa en C estándar para generar un binario. El usuario puede seleccionar el binario optimizado o sin optimizar para ejecutarlo en el procesador basado en RISC V.

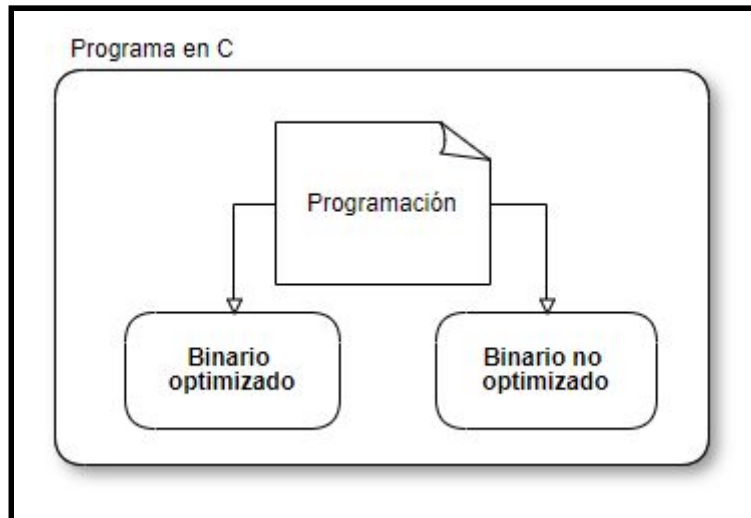


Figura 7. Estructura de Programa en C.

- Compilador RISC V: es un *toolchain* que proporciona GNU para la utilización de su compilador basado en RISC V, el cual es un compilador cruzado de C y C++ que permite la generación de código binario basado en una arquitectura RISC V ingresada, para el caso de este proyecto, la arquitectura RV32I.
- Script de tiempo de ejecución: es un programa que obtiene el tiempo que dura ejecutando el procesador basado en RISC V un programa ingresado por el usuario.
- Archivos de salida: el sistema debe de generar dos archivos de salida, uno que es el código binario generado por el compilador de RISC V y otro que es un archivo con las estadísticas obtenidas sobre la ejecución de un programa en el procesador basado en RISC V, el cual contiene el tiempo de ejecución del programa.

5.8. Vista de interacción

Esta vista define la comunicación entre las entidades del sistema, por medio de mensajes o variables. La vista se muestra a continuación:

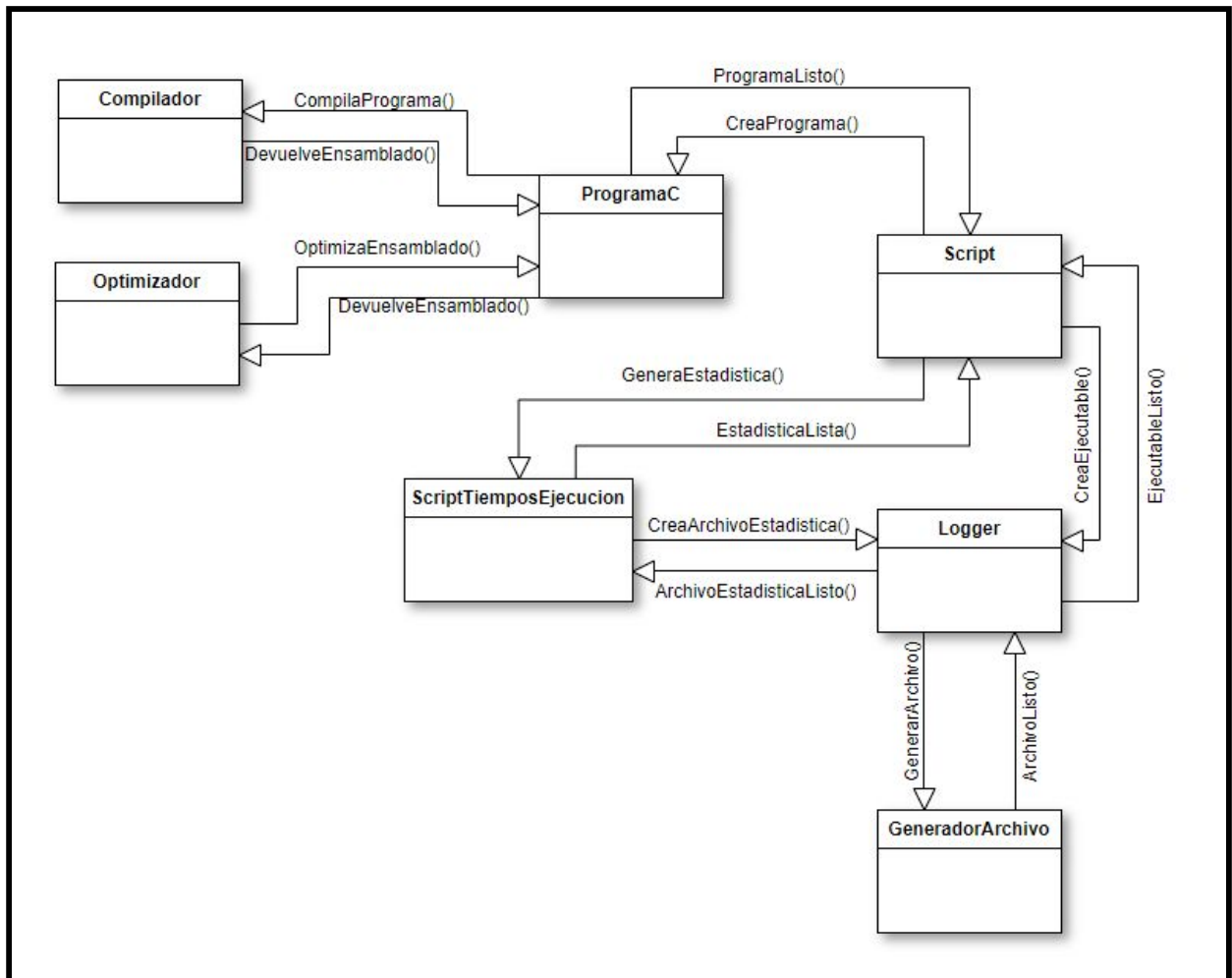


Figura 8. Vista de interacción.

5.9. Vista dinámica de estados

No aplica para el diseño de este proyecto.

5.10. Vista de algoritmos

Esta vista muestra los procedimientos que llevan a cabo por cada una de las clases que posee el sistema. Se explican los algoritmos utilizados en cada una de las entidades, son las que se muestran a continuación:

- **Script**: posee un algoritmo para la comunicación con el usuario mediante la terminal de linux, en los cuales es necesario verificar lo que el usuario ingresa, como por ejemplo el *path* de un programa a compilar, si desea crear un nuevo programa optimizado o sin optimizar o si desea generar estadísticas a partir de un archivo existente.

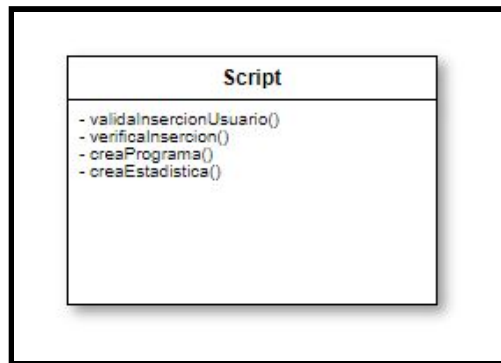


Figura 9. Vista de algoritmos del script.

- Compilador RISC V: usa algoritmos predefinidos por el compilador basado en RISC V versión RV32I.
- Optimizador: implementa algoritmos de emulación de código flotante para optimizar el uso de punto flotante en la arquitectura RISC V versión RV32I.

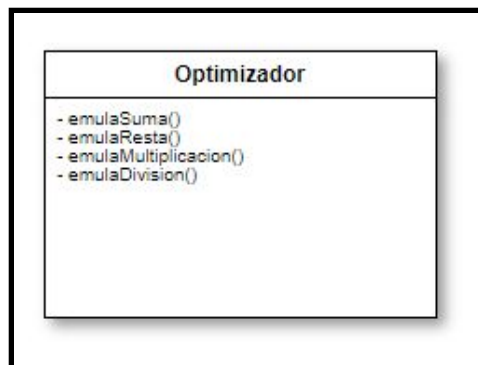


Figura 10. Vista de algoritmos del optimizador.

- Script tiempos ejecución: implementa un algoritmo que emula la ejecución de un programa creado por el usuario mediante un archivo de configuración de la arquitectura de hardware predefinida, o sea, el procesador basado en RISC V.

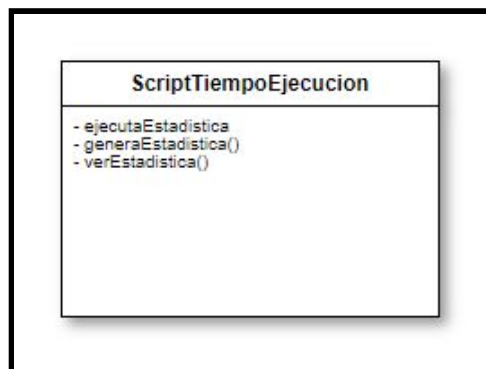


Figura 11. Vista de algoritmos del script de tiempos de ejecución.

5.11. Vista de recursos

No aplica para el diseño de este proyecto.