



# Trabajo de Fin de Grado

---

## Simulador didáctico de arquitectura de computadores

*Didactic simulator for Computer Architecture .*

Adrián Abreu González

---

La Laguna, 9 de mayo de 2017

D. **Iván Castilla Rodríguez**, con N.I.F. 78.565.451-G profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Simulador didáctico de arquitectura de computadores.”*

ha sido realizada bajo su dirección por D. **Adrián Abreu González**, con N.I.F. 54.111.250-R.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 9 de mayo de 2017

# Agradecimientos

XXX

XXX

XXX

XXX

# Licencia

\* Si quiere permitir que se compartan las adaptaciones de tu obra y quieres permitir usos comerciales de tu obra (licencia de Cultura Libre) indica:



© Esta obra está bajo una licencia de Creative Commons Reconocimiento 4.0 Internacional.

## Resumen

*El objetivo de este trabajo ha sido .... bla, bla, bla bla, bla, bla bla, bla, bla*

*La competencia [E6], que figura en la guía docente, indica que en la memoria del trabajo se ha de incluir: antecedentes, problemática o estado del arte, objetivos, fases y desarrollo del proyecto, conclusiones, y líneas futuras.*

*Se ha incluido el apartado de 'Licencia' con todas las posibles licencias abiertas (Creative Commons). En el caso en que se decida hacer público el contenido de la memoria, habrá que elegir una de ellas (y borrar las demás). La decisión de hacer pública o no la memoria se indica en el momento de subir la memoria a la Sede Electrónica de la ULL, paso necesario en el proceso de presentación del TFG.*

*El documento de memoria debe tener un máximo de 50 páginas.*

*No se deben dejar páginas en blanco al comenzar un capítulo, ya que el documento no está pensado para se impreso sino visionado con un lector de PDFs.*

*También es recomendable márgenes pequeños ya que, al firmar digitalmente por la Sede, se coloca un marco alrededor del texto original.*

*El tipo de letra base ha de ser de 14ptos.*

**Palabras clave:** Palabra reservada1, Palabra reservada2, ...

## Abstract

Here should be the abstract in a foreing language...

***Keywords:*** *Keyword1, Keyword2, Keyword3, ...*

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Sección Uno . . . . .	1
1.2. Sección Dos . . . . .	1
1.3. Sección Tres . . . . .	1
1.4. Sección Cuatro . . . . .	1
<b>2. Antecedentes</b>	<b>3</b>
2.1. Primera sección de otro capítulo . . . . .	3
<b>3. Estado del arte</b>	<b>4</b>
3.1. Primera sección de este capítulo . . . . .	4
3.2. Segunda sección de este capítulo . . . . .	4
3.3. Tercera sección de este capítulo . . . . .	4
<b>4. Objetivos y fases</b>	<b>5</b>
<b>5. Desarrollo del proyecto</b>	<b>6</b>
5.1. Migración del núcleo de la aplicación . . . . .	6
5.2. Migración de la máquina superescalar . . . . .	7
5.3. Desarrollo de la interfaz . . . . .	7
5.4. Integración interfaz - máquina superescalar . . . . .	7
<b>6. Ampliación de funcionalidades</b>	<b>8</b>
6.1. Documentación . . . . .	8
6.2. Parseador . . . . .	8
6.3. Batch mode . . . . .	9
6.4. Histórico . . . . .	9
<b>7. Conclusiones y líneas futuras</b>	<b>10</b>
7.1. Conclusiones . . . . .	10
7.2. Líneas futuras . . . . .	10
<b>8. Summary and Conclusions</b>	<b>11</b>
8.1. First Section . . . . .	11

<i>Título del proyecto</i>	II
<b>9. Presupuesto</b>	<b>12</b>
9.1. Sección Uno . . . . .	12
<b>A. Título del Apéndice 1</b>	<b>13</b>
A.1. Algoritmo XXX . . . . .	13
A.2. Algoritmo YYY . . . . .	13
<b>B. Título del Apéndice 2</b>	<b>14</b>
B.1. Otro apéndice: Sección 1 . . . . .	14
B.2. Otro apéndice: Sección 2 . . . . .	14
<b>Bibliografía</b>	<b>14</b>



# Índice de figuras

1.1. Ejemplo . . . . .	2
------------------------	---

# Índice de tablas

9.1. Presupuesto . . . . .	12
----------------------------	----

# Capítulo 1

## Introducción

### 1.1. Sección Uno

- Item 1
- Item 2
- Item 3
- Item 4

### 1.2. Sección Dos

- Item 1
- Item 2
- Item 3

### 1.3. Sección Tres

Bla, bla, bla

### 1.4. Sección Cuatro

Bla, bla, bla

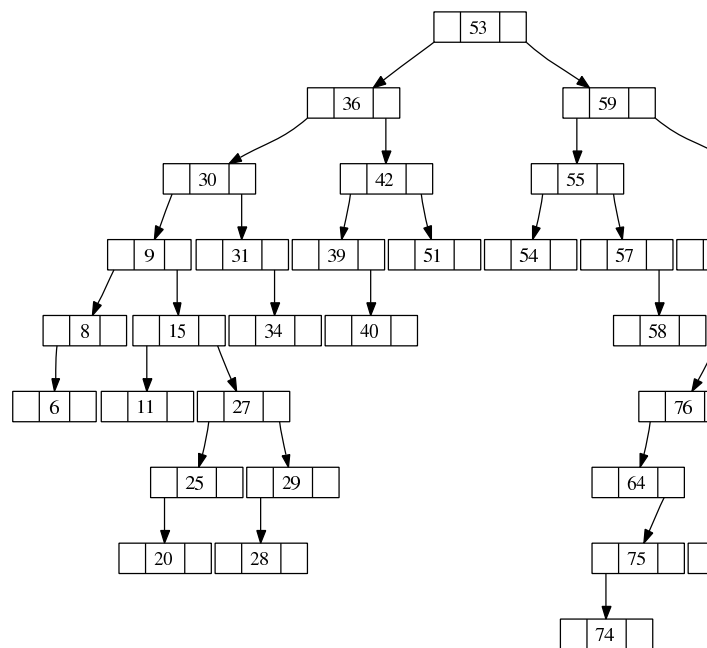


Figura 1.1: Ejemplo

# Capítulo 2

## Antecedentes

Los capítulos intermedios servían para cubrir los siguientes aspectos: antecedentes, problemática o estado del arte, objetivos, fases y desarrollo del proyecto.

En el capítulo anterior se ha introducido bla, bla, bla ....

### 2.1. Primera sección de otro capítulo

# Capítulo 3

## Estado del arte

Los capítulos intermedios servirán para cubrir los siguientes aspectos: antecedentes, problemática o estado del arte, objetivos, fases y desarrollo del proyecto.

Bla, Bla, Bla, .....

**3.1. Primera sección de este capítulo**

**3.2. Segunda sección de este capítulo**

**3.3. Tercera sección de este capítulo**

# Capítulo 4

## Objetivos y fases

El desarrollo del trabajo se puede dividir en cuatro fases principales que han permitido llevar a cabo la tarea de reescribir esta aplicación en web.

1. Migración del “núcleo de la aplicación”: Es decir las estructuras básicas comunes a las máquinas y el punto de entrada para la generación de las estructuras correspondientes, es decir, el analizador léxico de MIPS.
2. Migración de la máquina superescalar: El proceso completo de reescribir todas las estructuras de la máquina y su correcto funcionamiento en javascript.
3. Desarrollo de la interfaz: El desarrollo tanto de un diseño remodelado con las tecnologías web, como de componentes que gestionen correctamente su propio estado.
4. Integración interfaz-máquina: Realizar los enlaces necesarios entre el código desarrollado y la interfaz que permitan que .

# Capítulo 5

## Desarrollo del proyecto

### 5.1. Migración del núcleo de la aplicación

El núcleo de la aplicación se basa en el uso del generador de analizadores léxicos FLEX para parsear un conjunto de instrucciones similar al MIPS IV. Para realizar el proceso de migración he tenido que comprender primero el funcionamiento de los analizadores léxicos.

Aislando la implementación original y haciendo uso de una librería desarrollada para Javascript de Lex, se pudo conseguir en una escasa cantidad de tiempo tener en funcionamiento el código en Javascript.

Además, era imprescindible migrar las estructuras básicas que se crean cuando se carga el código, es decir: Las unidades funcionales y los bloques básicos y sucesores.

En este punto ya se tomó una de las decisiones más importantes para el desarrollo. Se decidió utilizar Typescript.

Typescript:

Podemos considerar a Typescript una pieza clave en el desarrollo del proyecto. A pesar de las distintas comodidades que han ido apareciendo en el mundo de Javascript con los distintos estándares, (es2015, es2016). Sigue siendo complicado trabajar con programación orientada a objetos de forma natural en Javascript.

Typescript es un lenguaje libre y de código abierto desarrollado por Microsoft que actúa como un superconjunto de javascript, y donde una de sus características más destacables es la capacidad de añadir tipado estático.

Este tipado no se refleja en el código final, de hecho una interfaz, por ejemplo, añade 0 sobrecarga en el código final. Pero si que es interesante por las capacidades de autocompletar (a través de Microsoft Intellisense) que añade. Typescript se puede transpilar directamente a código javascript es5, el cuál es el estándar actual en todos los navegadores.

Si tuviéramos que barajar una posible alternativa, sin duda la más destacada sería Flow, un comprobador de tipos para javascript desarrollado por Facebook.



Pero existen múltiples razones para que haya escogido typescript sobre Flow:

1. Flow no está siendo parte de un proyecto de gran envergadura, se suponía que se introduciría en el desarrollo de react. Sin embargo no ha sido así.
2. Typescript es la opción de defacto para Angular 2, ha ido ganando mucho entusiasmo por parte de la comunidad y tiene un gran soporte.
3. Tengo cierta experiencia con Typescript.

## **5.2. Migración de la máquina superescalar**

## **5.3. Desarrollo de la interfaz**

## **5.4. Integración interfaz - máquina superescalar**

# Capítulo 6

## Ampliacion de funcionalidades

En este proyecto de fin de grado se han realizado cuatro ampliaciones de las funcionales.

### 6.1. Documentación

El primero de ellos es recuperación de la documentación. En su proyecto original el profesor Iván Castilla elaboró una extensa documentación sobre SIMDE que contenía consejos sobre el desarrollo de aplicaciones para las distintas máquinas y no solo eso, sino que además explicaba el funcionamiento de las máquinas.

Por desgracia, esta documentación fue realizada en formato .HLP, un formato de windows que quedó en desuso desde Windows Vista. Es por eso, que utilizando herramientas de extracción, se pudieron recuperar los textos originales de parte del profesor.

Para integrar esta documentación en la aplicación web lo mejor era hacer un sistema web también. Por ello se ha migrado la documentación a un formato muchísimo más mantenible como es markdown y utilizando un generador de contenido estático basado en NodeJS (Hexo) y un tema apropiado, se ha conseguido que la documentación esté integrada de nuevo.

IMAGEN DOCUMENTACIÓN

### 6.2. Parseador

Una de las características más deseadas por parte de los usuarios de SIMDE (entre los que yo mismo me puedo incluir), es un sistema de errores más descriptivo.

Por desgracia, en la versión original de SIMDE solo se muestra el mensaje

”Error”.

Ahora, tras una serie de modificaciones en el parseados de código, se muestran los siguientes errores:

1. Operando erróneo
2. Opcode desconocido
3. Etiqueta repetida

Además, se muestra la línea del error. Esto resulta tremendamente importante, ya que una de las aplicaciones de SIMD requiere realizar mejoras en el rendimiento de código haciendo uso de técnicas como el desenrollado de bucles, dando lugar a códigos de considerable tamaño.

IMAGENES ERROR

## 6.3. Batch mode

Otra característica que habría sido resultado muy deseable por mi parte y por parte de mis compañeros, es no tener el limite de velocidad que tiene la versión original.

En mi caso, la cantidad media de ciclos de los ejercicios que se me propusieron superaba los 500 ciclos, haciendo un tiempo medio de ejecución de 2 - 3 minutos, lo cual sumado a la media de pruebas, a la depuración de errores, y a las distintas optimizaciones, alargaba de forma innecesaria el desarrollo de los ejercicios.

Es por eso que ahora, cuando el campo de velocidad está en 0, se ejecuta la simulación a velocidad máxima, y solo se refresca la interfaz cuando termina la ejecución. Ahorrando así recursos.

## 6.4. Histórico

# Capítulo 7

## Conclusiones y líneas futuras

### 7.1. Conclusiones

### 7.2. Líneas futuras

Tras el desarrollo de este trabajo se abren varias líneas futuras:

- Implementación de la máquina VLIW: Esta línea no resulta sorprendente. Con el desarrollo de este trabajo de fin de grado también se han implementado las estructuras básicas que se comparte con la máquina VLIW. Esta línea de trabajo tiene la mayor prioridad, pues equipara la funcionalidad de la aplicación web de SIMD E a la aplicación original.
- Realizar una mayor cantidad de test: En el mundo web no resulta sencillo realizar test para los distintos casos, sin embargo, la lógica que acompaña al simulador es un gran candidato a ser testeado. Con las bases asentadas en los tests realizados para la estructura de la cola y del parseador del código, se podría extender este funcionamiento a pequeñas simulaciones.
- Realizar un tutorial de inicio: Ahora que SIMD E es una aplicación accesible para todo el mundo, sería deseable que no supusiera una barrera para cualquier usuario que sintiera la necesidad de probar
- Implementar un sistema de gestión de estados: En la aplicación actual se ha hecho un sistema de estados rudimentario debido al volumen de trabajo de sistemas como Redux y a las dificultades que entrañan los Observables que vienen en un sistema como Mobx. En líneas futuras este sistema rudimentario podría sustituirse por un sistema más robusto desarrollado por terceros.

# Capítulo 8

## Summary and Conclusions

This chapter is compulsory. The memory should include an extended summary and conclusions in english.

### 8.1. First Section

# Capítulo 9

## Presupuesto

### 9.1. Sección Uno

Descripción	Coste
200 Horas de trabajo	8000 €
Ordenador para desarrollo	1300 €
Total	9300 €

Tabla 9.1: Presupuesto

# Apéndice A

## Título del Apéndice 1

### A.1. Algoritmo XXX

```
*****
*
* Fichero .h
*
*****
*
* AUTORES
*
*
* FECHA
*
*
* DESCRIPCION
*
*
*****/
```

### A.2. Algoritmo YYY

```
/*****
*
* Fichero .h
*
*****
*
* AUTORES
*
* FECHA
*
* DESCRIPCION
*
*
*****/
```

# Apéndice B

## Título del Apéndice 2

### B.1. Otro apéndice: Sección 1

Texto

### B.2. Otro apéndice: Sección 2

Texto



# Bibliografía

- [1] ACM LaTeX Style. [http://www.acm.org/publications/latex\\_style/](http://www.acm.org/publications/latex_style/).
- [2] FACOM OS IV SSL II USER'S GUIDE, 99SP0050E5. Technical report, 1990.
- [3] D. H. Bailey and P. Swarztrauber. The fractional Fourier transform and applications. *SIAM Rev.*, 33(3):389–404, 1991.
- [4] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comp. Phys.*, 49:443–457, 1983.
- [5] C. Darwin. *The Origin Of Species*. November 1859.
- [6] C. Goldstein. Multigrid methods for elliptic problems in unbounded domains. *SIAM J. Numer. Anal.*, 30:159–183, 1993.
- [7] P. Swarztrauber. *Vectorizing the FFTs*. Academic Press, New York, 1982.
- [8] S. Taásan. *Multigrid Methods for Highly Oscillatory Problems*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1984.