

# Índice general

Índice de figuras	II
Índice de tablas	III
<b>1. Introducción</b>	<b>1</b>
1.1. Arritmias . . . . .	1
1.2. Algoritmo de deteccion . . . . .	1
1.2.1. Filtrado . . . . .	2
1.3. Pruebas con pacientes . . . . .	2
1.4. Utilizacion de las FPGAs . . . . .	3
1.5. Objetivos del proyecto y organización . . . . .	4
1.6. Analisis y optimizacion del algoritmo . . . . .	4
1.7. Implementacion en la FPGA . . . . .	5
1.8. Plantilla para usos de la herramienta . . . . .	5
<b>2. Planteamiento del algoritmo en software</b>	<b>7</b>
2.1. Recopilacion de los datos . . . . .	7
2.2. Filtrado de la señal original . . . . .	8
2.3. Detección de picos QRS . . . . .	9
2.4. Detección de arritmias . . . . .	9
2.5. Pruebas con el algoritmo . . . . .	9
<b>3. Idea</b>	<b>10</b>
<b>4. Implementation</b>	<b>11</b>
<b>5. Results</b>	<b>12</b>
<b>6. Conclusiones y trabajo futuro</b>	<b>13</b>
<b>Bibliography</b>	<b>15</b>

# Índice de figuras

1.1. Electrocardiogramas . . . . .	1
1.2. Complejo QRS . . . . .	2
1.3. Ejemplo de electrocardiograma original y filtrado de paciente 102 . . . . .	2
1.4. Ejemplo con paciente 102 . . . . .	3
1.5. Basys3 Artix-7 FPGA . . . . .	4
1.6. Sample figure . . . . .	5

# Índice de tablas

1.1. Sample table . . . . .	6
-----------------------------	---

# Capítulo 1

## Introducción

### 1.1. Arritmias

Las enfermedades cardiovasculares son la primera causa de muerte en el mundo y una de las causas mas comunes de estas enfermedades son las arritmias.

Una arritmia cardiaca es una alteración en el ritmo normal del corazón. Si se produce una arritmia, el corazón puede latir demasiado rápido, demasiado lento o de manera irregular. Esto puede provocar síntomas como palpitaciones, mareos, falta de aire e incluso desmayos y estas pueden llegar a ser mortales.

Los cardiólogos utilizan dispositivos como un Holter para generar tiras de ritmo o electrocardiogramas, que es un diagrama que representa los latidos del corazón y con eso pueden llegar a detectar arritmias.

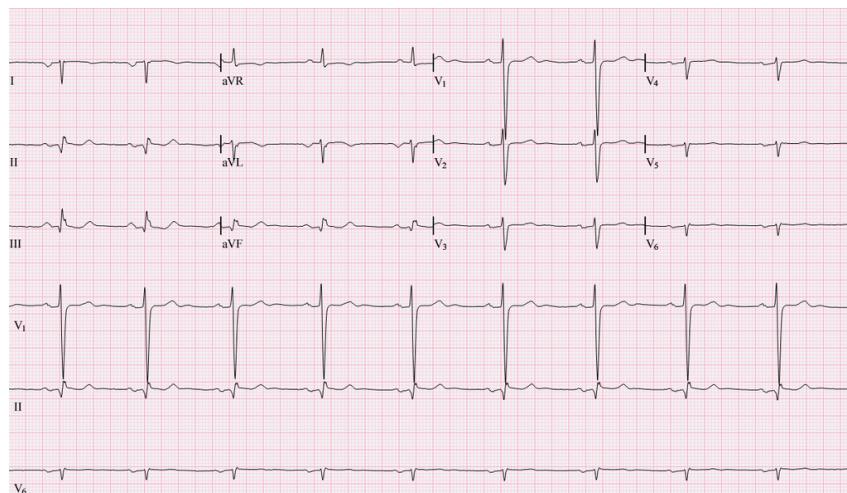


Figura 1.1: Electrocardiogramas

En este proyecto se tratara de solucionar las arritmias en las que se produce una contraccion prematura del corazón como las contracciones prematuras del corazón. Estas arritmias se pueden detectar con un electrocardiograma (ECG) que es un diagrama de los latidos del corazón.

### 1.2. Algoritmo de deteccion

Dado que para detectar arritmias correctamente se necesitan varios años de cardiología, el algoritmo de detección que se utilizara consistira en detectar las arritmias unicamente usando los picos QRS del electrocardiograma.

Un pico QRS como se muestra en la Figura 1.2 en un electrocardiograma es causado por la contracción del ventrículo al bombear la sangre por las arterias. Este es el impulso eléctrico mas fuerte que el corazón produce en cada latido. En este proyecto utilizaremos estos picos para comparar la distancia entre ellos y poder ver si se ha producido una arritmia.

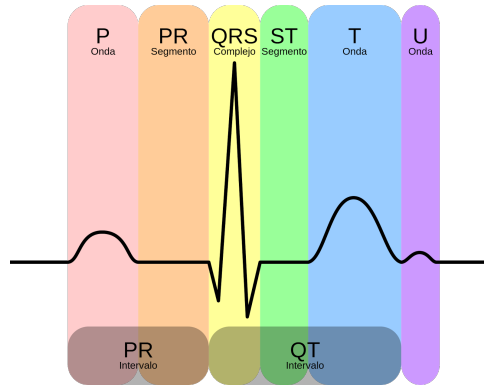


Figura 1.2: Complejo QRS

### 1.2.1. Filtrado

Como se puede ver en las imágenes es conveniente hacer un filtrado de las tiras de ritmo para poder detectar mejor los picos QRS. Ya que el filtrado centra la onda en el 0 y evita fallos en el algoritmo de detección de picos del que se hablará mas adelante.

En la creación del proyecto se ha intentado no filtrar la onda para comprobar si se obtienen mejores resultados que sin dicho filtrado pero no se ha dado el caso por las irregularidades de la misma.

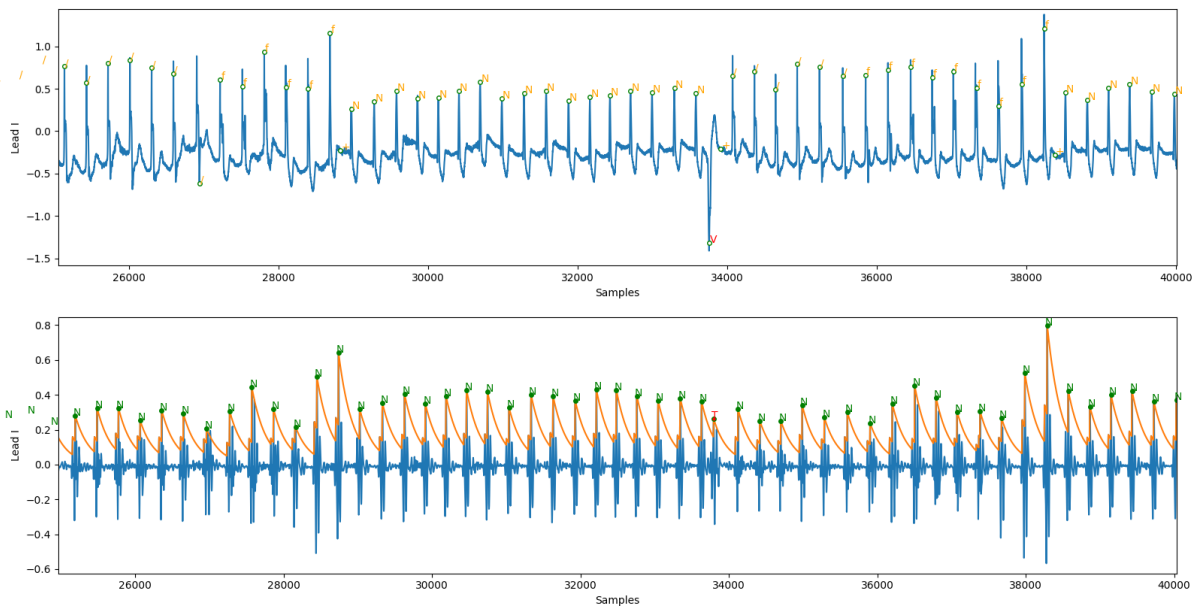


Figura 1.3: Ejemplo de electrocardiograma original y filtrado de paciente 102

## 1.3. Pruebas con pacientes

Se han realizado las pruebas con unos resultados del Instituto de Tecnología de Massachusetts (MIT) en el que se han recogido tiras de ritmo de media hora de varios pacientes con edades diversas y algunos de ellos llevan un marcapasos que actúa cuando el corazón no bombea la sangre lo suficientemente fuerte, es decir que el pico QRS no es tan prominente y se necesita la ayuda de dicho marcapasos para proporcionar el impulso eléctrico necesario.

Estas pruebas han sido analizadas por cardiólogos y se ha indicado donde el paciente padece una arritmia y donde el ritmo es normal y donde se ha producido un error en la lectura de la señal. También muestra información menos relevante como la activación del marcapasos.

### Record 102 (V5, V2; female, age 84)

*Medications:* Digoxin

Beats	Before 5:00	After 5:00	Total
Normal	98	1	99
PVC	1	3	4
Paced	243	1785	2028
Pacemaker fusion	24	32	56
Total	366	1821	2187

*Ventricular ectopy*

- 4 isolated beats

Rhythm	Rate	Episodes	Duration
Normal sinus rhythm	72-78	2	1:22
Paced rhythm	68-78	3	28:44

**Signal quality** Episodes Duration

Both clean 1 30:06

Notes:

The rhythm is paced with a demand pacemaker. The PVCs are multiform.

#### Points of interest:

[0:55](#) Paced rhythm  
[1:12](#) Transition from paced to normal sinus rhythm  
[1:28](#) PVC  
[2:30](#) Normal sinus rhythm  
[4:51](#) Pacemaker fusion beats  
[9:35](#) PVC  
[16:12](#) Paced rhythm

Figura 1.4: Ejemplo con paciente 102

## 1.4. Utilización de las FPGAs

Este proyecto requiere un gran procesamiento de señales, una alta cantidad de cálculos y un eficiente paralelismo entre módulos por ello la mejor forma de optimizar el algoritmo es utilizando una FPGA.

Los motivos son los siguientes:

- Las FPGA pueden procesar datos a velocidades muy altas, lo que lo hace indispensable para esta aplicación que está pensada para ejecutarse en tiempo real.
- Las FPGA son dispositivos de hardware programable que permite diseñar circuitos digitales personalizados, y por ello pueden reconfigurarse para adaptarse a tareas específicas. Además son susceptibles a cambios en el algoritmo para una posible mejora de este.
- El alto paralelismo que ofrecen las FPGA es perfecto para las multitareas que realiza el algoritmo.
- Puesto que las FPGA pueden ser diseñadas para realizar una tarea en concreto, estas son más energéticamente eficientes que otros dispositivos como los portátiles.

Para este proyecto se usara la FPGA Basys3 de Artix-7 para probar el funcionamiento del algoritmo. Aunque se debe considerar, segun la cantidad de datos introducidos, que en este caso seria la longitud de la señal segun el tiempo transcurrido, utilizar una FPGA cuyo hardware pueda soportar dicha cantidad de datos.

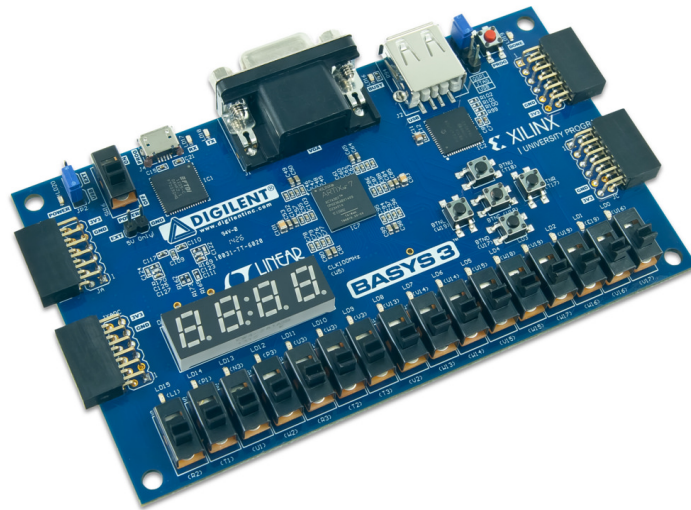


Figura 1.5: Basys3 Artix-7 FPGA

## 1.5. Objetivos del proyecto y organización

Los objetivos de este proyecto es tener una solución para detectar contracciones prematuras ventriculares a tiempo real en un largo periodo de tiempo y optimizar el algoritmo para que se ejecute de una forma mas eficiente y menos costosa en una FPGA

Para ello la organización de este proyecto comienza con la creación de el prototipado del algoritmo en software para facilitar la manera de probar el algoritmo con la solución proporcionada por la base de datos y poder ver resultados graficos, para mejorar la velocidad de compilación y depuración del algoritmo, para aumentar la claridad del algoritmo que se quiere conseguir en el prototipado y para validar la funcionalidad y eficacia del algoritmo.

## 1.6. Analisis y optimización del algoritmo

Para lograr los objetivos del algoritmo se centra en tres funciones.

1. Filtrado de la señal original: Lo que hace que la señal sea mas facil de procesar para encontrar los picos QRS. Esto se realiza multiplicando los valores de la señal original por los valores de filtrado.
2. Detección de picos sobre la señal filtrada: Se analiza cada señal y comparandola con otras señales anteriores se deduce si puede ser un posible pico y si lo es, se comprueba si es un pico QRS.
3. Detección de arritmias comparando la posición de los picos: una vez se tienen los picos QRS se calcula la distancia de el pico actual con el pico anterior y dependiendo de las otras distancias se calcula si hay una arritmia.

## 1.7. Implementacion en la FPGA

Para implementar el codigo en la FPGA se implementara varios modulos para tratar de imitar el proyecto creado en software los modulos mas importantes son.

1. Modulo de filtrado: Lo que hace que la señal sea mas facil de procesar para encontrar los picos QRS. Esto se realiza multiplicando los valores de la señal original por los valores de filtrado.
2. Deteccion de picos sobre la señal filtrada: Se analiza cada señal y comparandola con otras señales anteriores se deduce si puede ser un posible pico y si lo es, se comprueba si es un pico QRS.
3. Deteccion de arritmias comparando la posicion de los picos: una vez se tienen los picos QRS se calcula la distancia de el pico actual con el pico anterior y dependiendo de las otras distancias se calcula si hay una arritmia.

Estos modulos tratan de replicar las funcionalidades que realiza el algoritmo de software y se convertiran en la parte esencial de dicho programa.

Ademas de estos modulos se debe de crear un modulo que acompase a estos tres y un testbench para probar el funcionamiento del programa en la simulación.

## 1.8. Plantilla para usos de la herramienta

The document is divided into chapters, sections, and subsections.

Some important references are [1-3].

To add paragraphs in the document, one line break is not enough,  
two line breaks are needed.

An itemized list:

- An item.
- Another item.
- Final item.

An enumerated list:

1. First item.
2. Second item.
3. Third item.

A figure with an image is presented in Figura 1.6. Note that it floats away and latex places it where convenient.



Figura 1.6: Sample figure

Tables work in the same way, as seen in Tabla 1.1



Row	English	Español
1	One	Uno
2	Two	Dos

Tabla 1.1: Sample table

# Capítulo 2

## Planteamiento del algoritmo en software

### 2.1. Recopilacion de los datos

Para la recopilacion de los datos se utilizara la libreria wfdb que se encarga de proporcionar funciones para leer y escribir archivos de diferentes formatos que contienen señales biomédicas, como archivos de registro de señales (por ejemplo, formato .dat), archivos de anotaciones (por ejemplo, formato .atr) y archivos de cabecera (por ejemplo, formato .hea).

Los pacientes vienen identificados por un id (por ejemplo, 101) y hay 3 ficheros por paciente, con extensiones .dat, .atr y .hea

Se descarga la base de datos con la funcion de la libreria de wfdb, dldatabase que recoge la señal del paciente y las anotaciones de los cardiologos sobre cada pico QRS.

```
#download the database if not available
if os.path.isdir("mitdb"):
    print('You already have the data.')
else:
    wfdb.dl_database('mitdb', 'mitdb')
```

Los pacientes de la base de datos se han hecho una prueba de 30 mins lo que en la señal equivale a 650000 samples.

```
sampfrom = 0
sampto = 650000
record = wfdb.rdscamp('mitdb/102', sampfrom=sampfrom, sampto=sampto)
annotation = wfdb.rdann('mitdb/102', 'atr', sampfrom=sampfrom, sampto=sampto)
```

Por ultimo, para visualizar esta señal con las anotaciones de los cardiologos y poder comparar con las anotaciones que realiza el algoritmo se usara la libreria matplotlib.pyplot.

Con esto se mostrara la señal original con las anotaciones y la señal filtrada con las anotaciones del algoritmo como en Figura 1.3

```
#plot the signal
#add markers to the original signal
ax[0].plot(original_signal)
ax[1].plot(filtered_signal)
ax[0].set_xlabel('Samples')
ax[0].set_ylabel('Lead-I')
ax[1].set_xlabel('Samples')
ax[1].set_ylabel('Lead-I')

#Making the upper signal
for pos, sym in zip(annotation.sample, annotation.symbol):
    pos -= sampfrom
    ax[0].plot(pos, original_signal[pos], 'go', markersize=4, markerfacecolor='white')
    if(sym == "A" or sym == "V" or sym == "a"):
        ax[0].text(pos+10, original_signal[pos], sym, color='red')
    else:
        ax[0].text(pos+10, original_signal[pos], sym, color='orange')
```

## 2.2. Filtrado de la señal original

Este filtrado es llevado a cabo por el filtrado IIR.

El filtrado IIR, que significa "Infinite Impulse Response" (respuesta infinita al impulso), es un tipo de filtro utilizado en el procesamiento de señales digitales y analógicas.

La formula que se utilizara para el filtrado es

$$Y[i] = \sum_{k=0}^{N_x-1} b_k \cdot x[i - k]$$

Con lo que b son los coeficientes y x la señal a filtrar

Los coeficientes son

```
filter_taps_99_6_28 = [ -0.022064516719960285, -0.01248143191787449,
-0.014272402451704691, -0.01482693239340742, -0.013830723879265979,
-0.011121793043893711, -0.006713818327951467, -0.0008089345809688635,
0.006216127611731759, 0.013857892265470411, 0.02152515460626638,
0.028576965837020497, 0.034398765027779624, 0.03844530602431575,
0.04025284446627053, 0.039530808378156514, 0.0361389960690063,
0.030158595190338606, 0.021892977443216645, 0.01187877085962161,
0.0008679402497379659, -0.010209492943778627, -0.020330840764461847,
-0.02848685236456877, -0.03381377331912697, -0.03574615252534521,
-0.03413529962465352, -0.02933049408837693, -0.022191934029519987,
-0.014013910197915145, -0.006355535107839268, -0.0008277532244847638,
0.0011959567983135999, -0.0011634462076121899, -0.008076950035987168,
-0.018875074584752207, -0.0320701386231862, -0.04551445963999694,
-0.05668375253673195, -0.0630716859149452, -0.06260700837237966,
-0.05404472876437898, -0.037193610758084875, -0.013167513231520076,
0.01590495809018272, 0.04685141764647888, 0.07609319762164894,
0.10009947011806296, 0.11585338656670081, 0.12133902448458535,
0.11585338656670081, 0.10009947011806296, 0.07609319762164894,
0.04685141764647888, 0.01590495809018272, -0.013167513231520076,
-0.037193610758084875, -0.05404472876437898, -0.06260700837237966,
-0.0630716859149452, -0.05668375253673195, -0.04551445963999694,
-0.0320701386231862, -0.018875074584752207, -0.008076950035987168,
-0.0011634462076121899, 0.0011959567983135999, -0.0008277532244847638,
-0.006355535107839268, -0.014013910197915145, -0.022191934029519987,
-0.02933049408837693, -0.03413529962465352, -0.03574615252534521,
-0.03381377331912697, -0.02848685236456877, -0.020330840764461847,
-0.010209492943778627, 0.0008679402497379659, 0.01187877085962161,
0.021892977443216645, 0.030158595190338606, 0.0361389960690063,
0.039530808378156514, 0.04025284446627053, 0.03844530602431575,
0.034398765027779624, 0.028576965837020497, 0.02152515460626638,
0.013857892265470411, 0.006216127611731759, -0.0008089345809688635,
-0.006713818327951467, -0.011121793043893711, -0.013830723879265979,
-0.01482693239340742, -0.014272402451704691, -0.01248143191787449,
-0.022064516719960285]
```

Para el filtrado se usa la funcion lfilter de la libreria scipy.signal

```
filtered_signal = lfilter(filter_taps_99_6_28 , 1.0 , original_signal)
```

TODO a y formula completa ademas de una mejor explicacion

- 2.3. Detección de picos QRS
- 2.4. Detección de arritmias
- 2.5. Pruebas con el algoritmo

# Capítulo 3

## Idea

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# Capítulo 4

## Implementation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# Capítulo 5

## Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# Capítulo 6

## Conclusiones y trabajo futuro

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.



# Conclusions and future work

Translate the previous chapter

# Bibliografía

- [1] Albert Einstein. «Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]». En: *Annalen der Physik* 322.10 (1905), págs. 891-921.
- [2] Michel Goossens, Frank Mittelbach y Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Reading, Massachusetts: Addison-Wesley, 1993.
- [3] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~{uno}/abcde.html>.