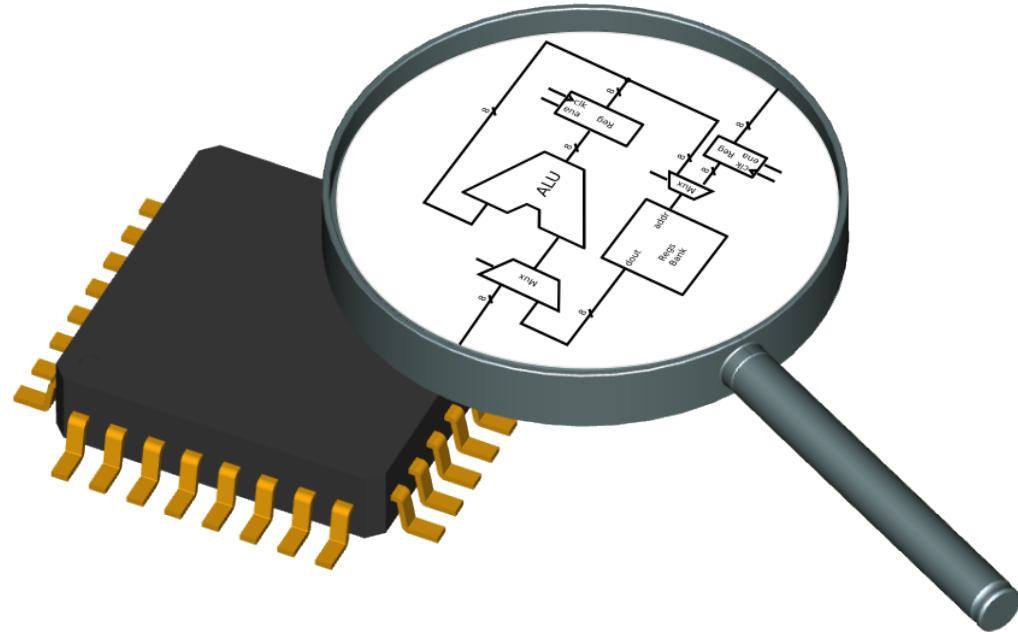


# FPGAs libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>

# Contenido

**PARTE I: Circuitos digitales y  
FPGAs**

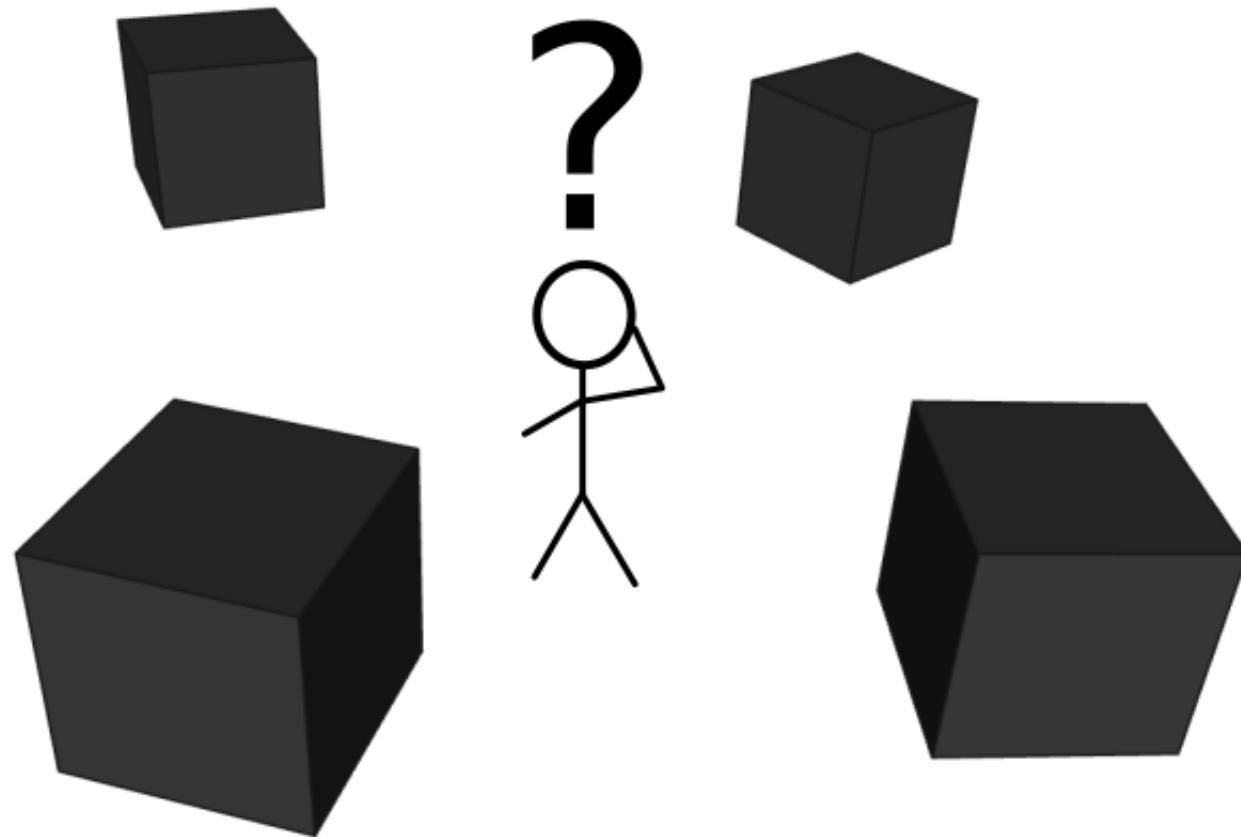
**PARTE II: Herramientas libres**

**PARTE III: ¿Cómo empiezo?**

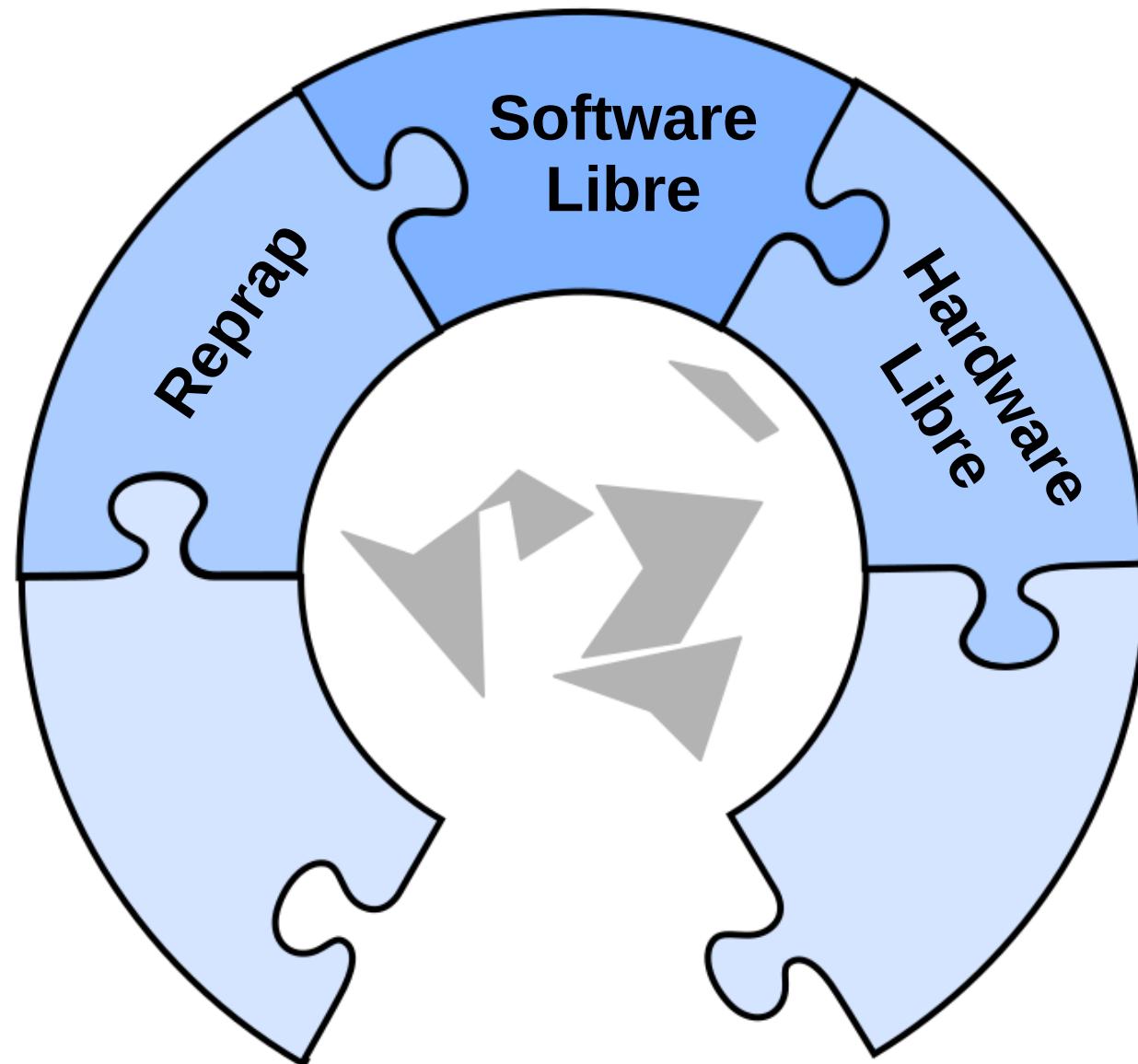
# **PARTE I:**

# **Circuitos digitales y FPGAs**

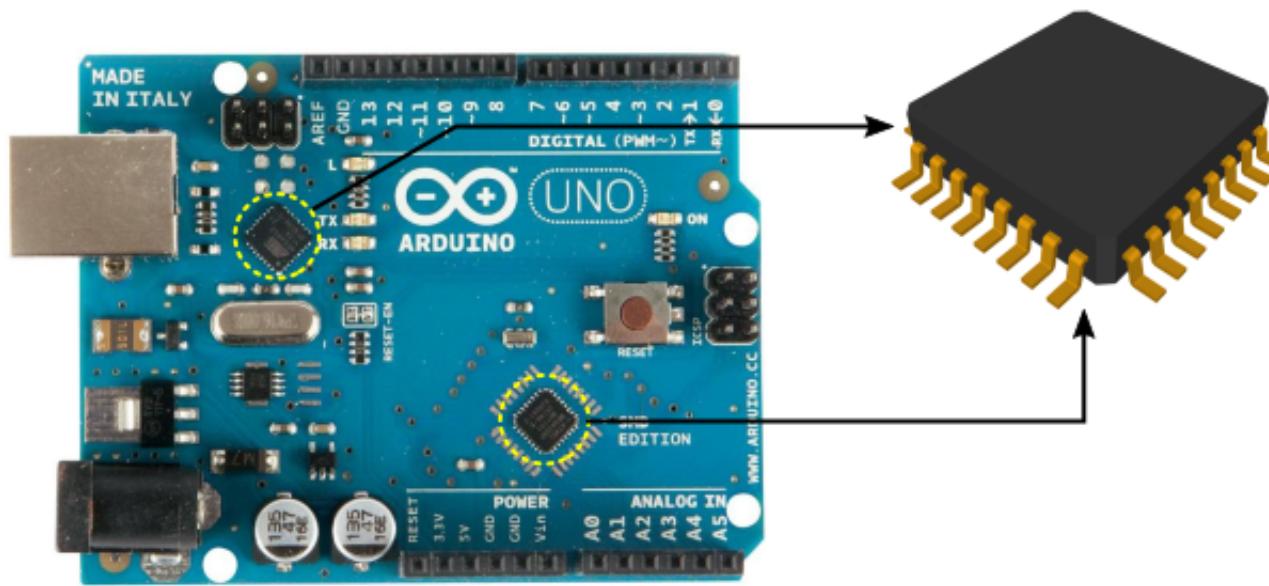
# Nuevas tecnologías: Mundo de cajas negras



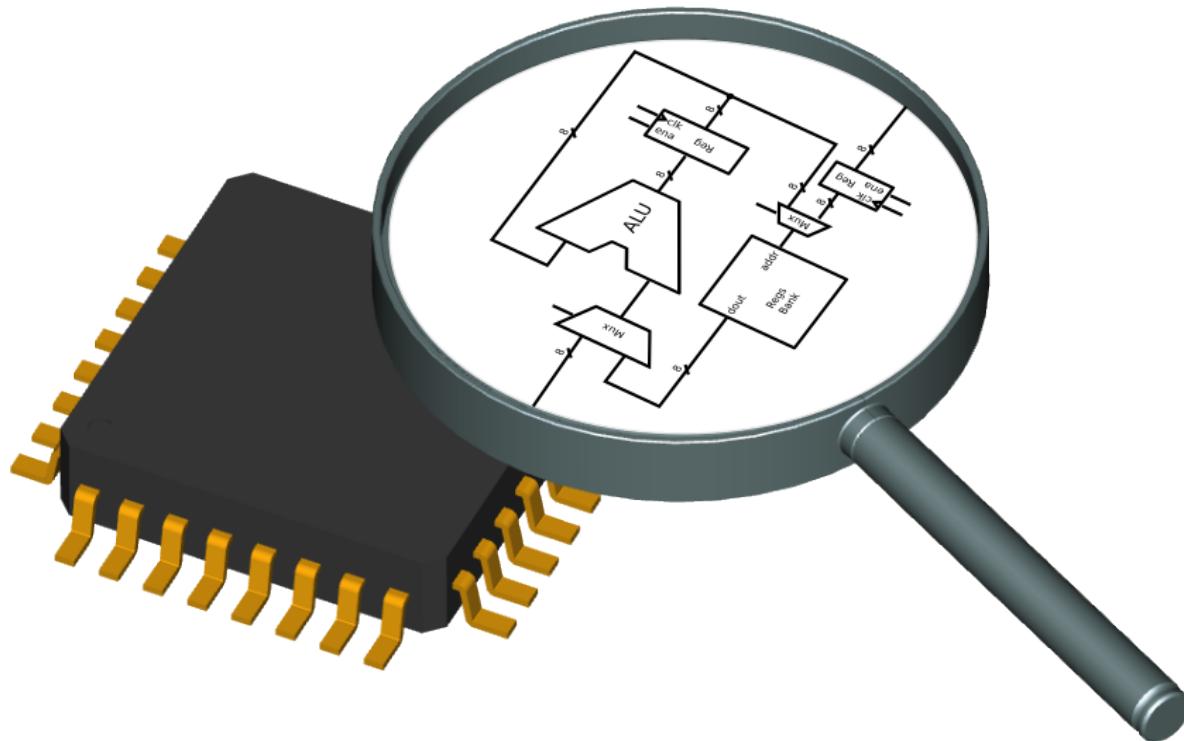
# Patrimonio Tecnológico de la humanidad



# Pasemos al siguiente nivel: Chips digitales



# Viaje al interior de los chips digitales

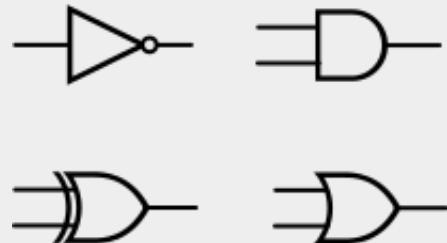


- Nivel de electrónica digital
- Información: Sólo 1s y 0s (Bits)
- Función: **Manipular, almacenar y transportar bits**

# Elementos en circuitos digitales

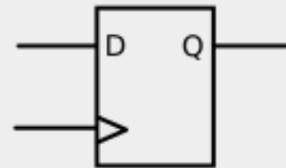
## Puertas lógicas

Manipulación bits



## Biestables

Almacenamiento bits

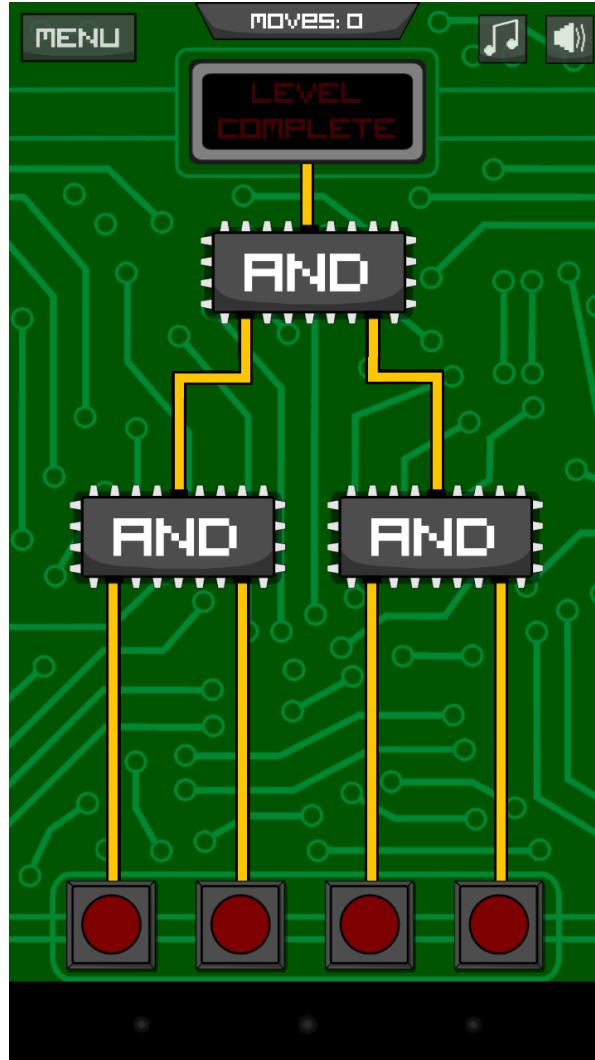


## Cables

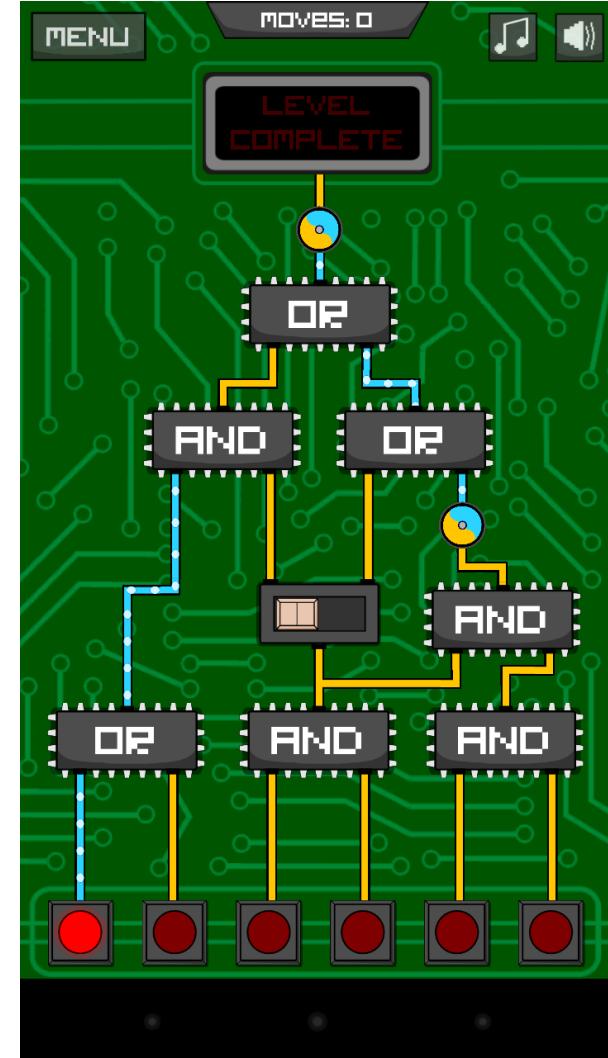
Transporte bits



Cualquier circuito digital, por muy complejo que sea, se descompone en estos 3 tipos de **componentes elementales**

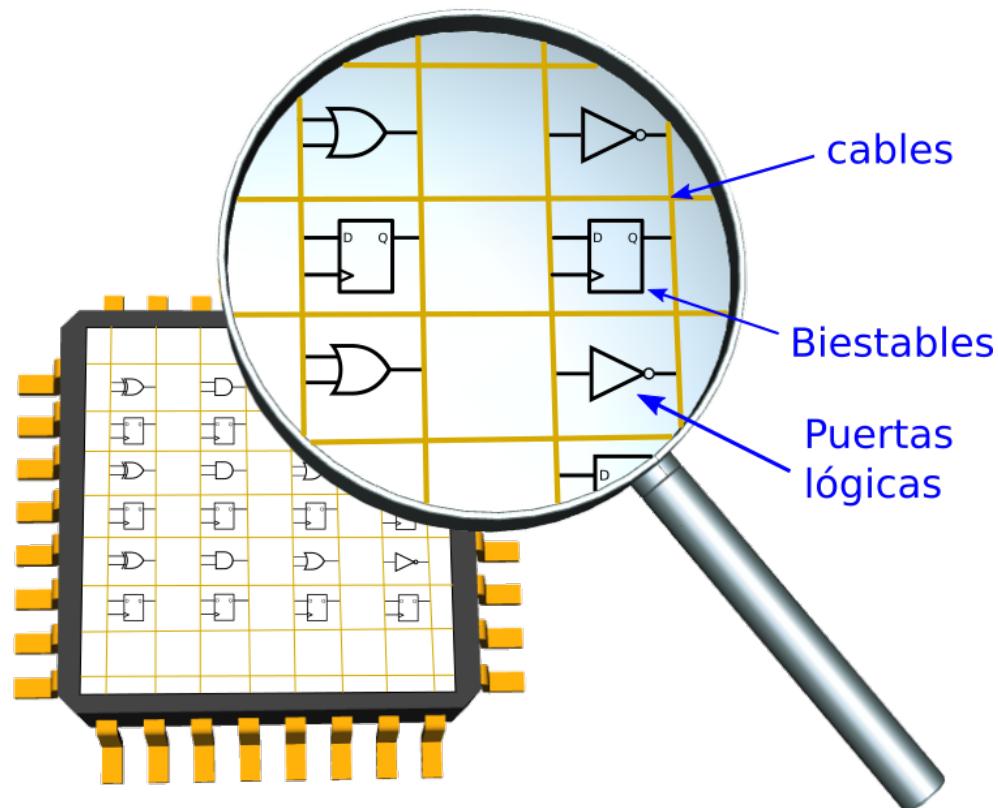


APP:  
Circuit  
Scramble

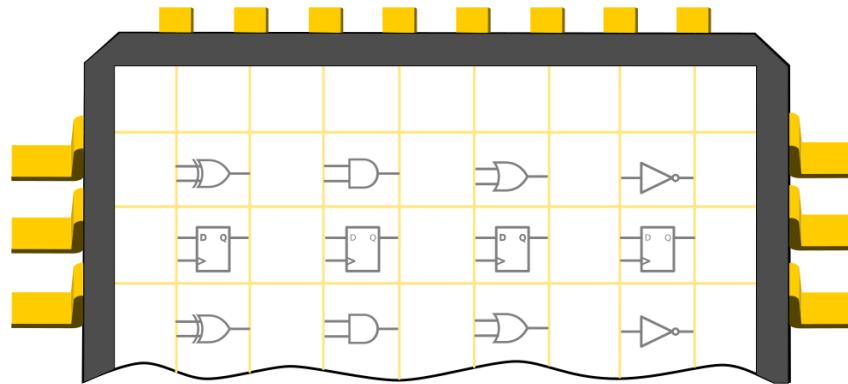


La electrónica digital es intuitiva y...  
¡Divertida!

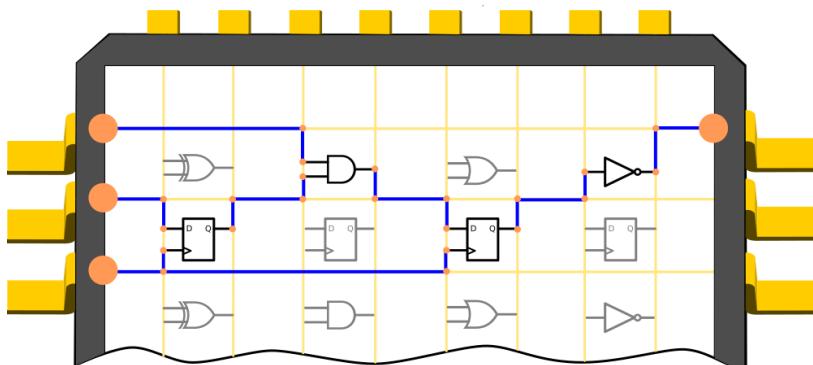
# FPGAs: Implementando circuitos digitales



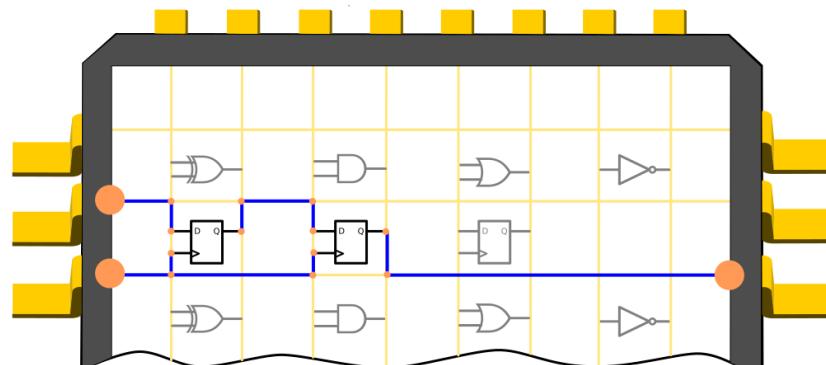
**FPGA:** Chip “en blanco” que contiene una matriz con los 3 componentes básicos: puertas lógicas, biestables y cables



Configuración

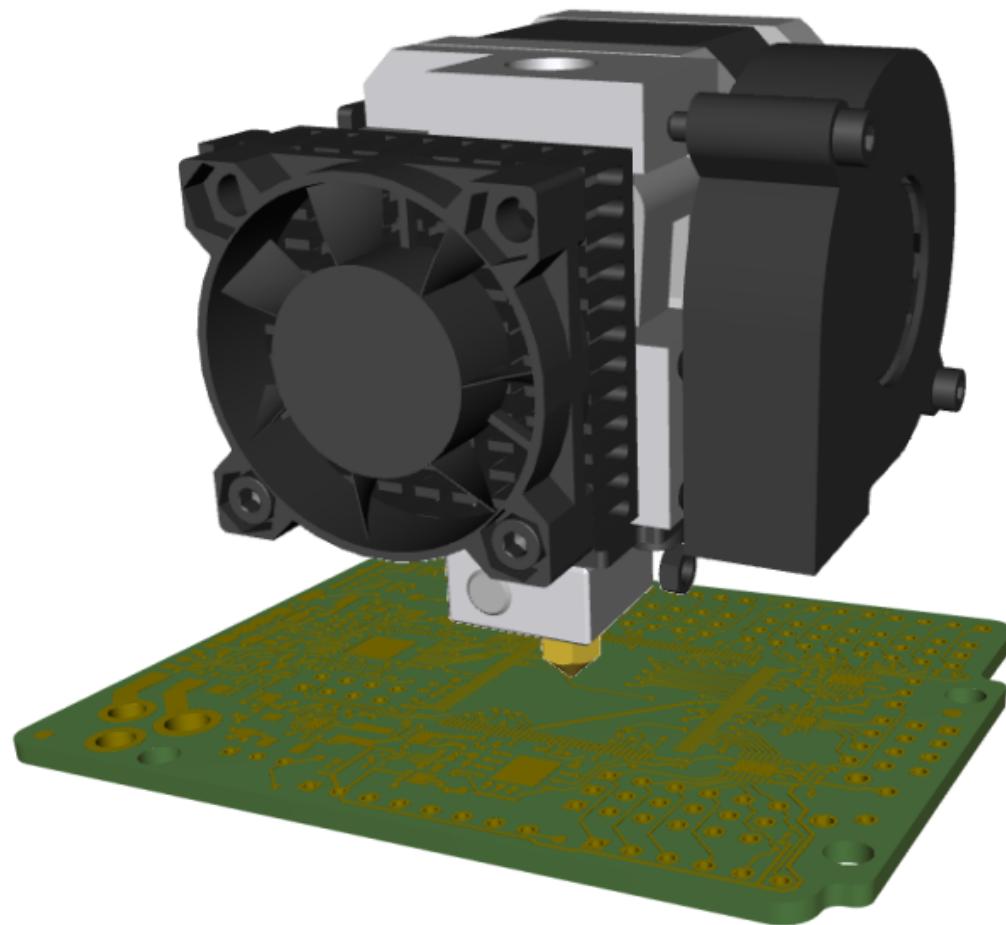


Circuito 1

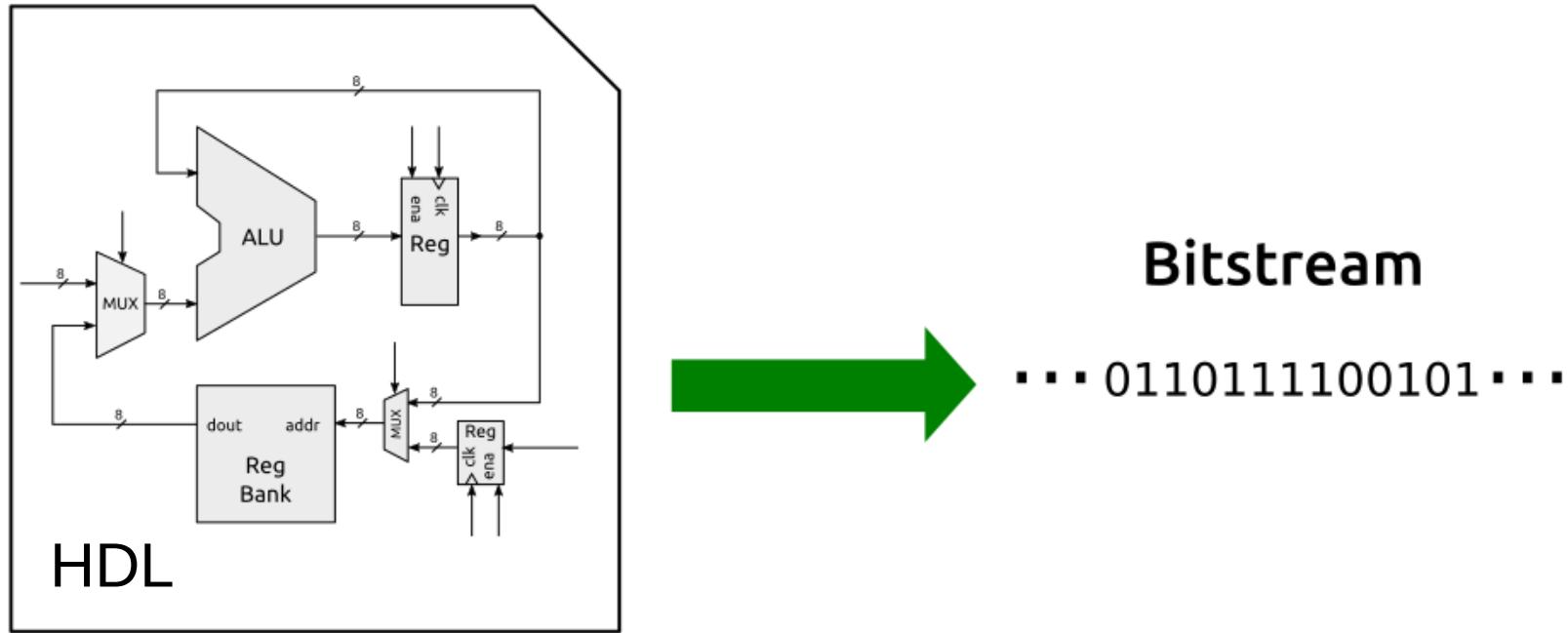


Circuito 2

**¡FPGAs = Impresoras 3D de circuitos digitales!**



# El hardware es software



Bitstream

... 0110111100101 ...

HDL

- El hardware libre es igual al software libre
- Muy fácil de compartir
- Telecopias del hardware
- Desarrollo de hardware en comunidad

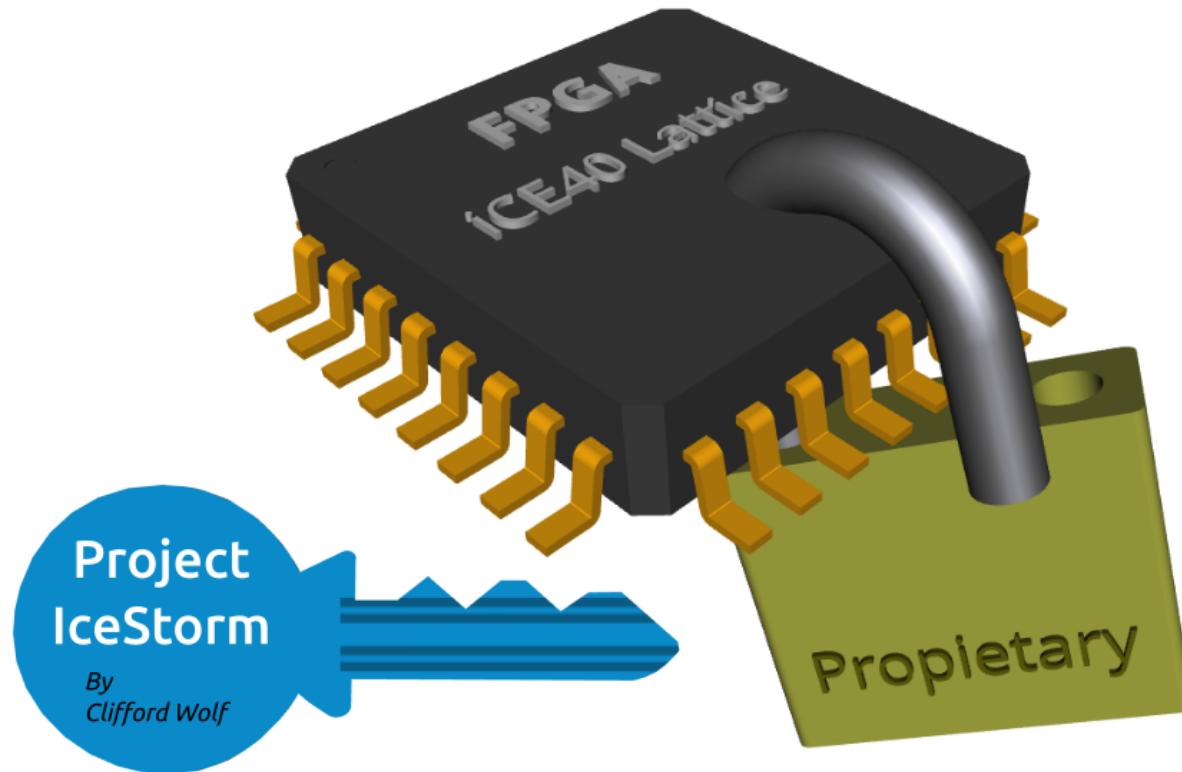
# **PARTE II:**

# **HERRAMIENTAS LIBRES**

# FPGAs: Sólo personal autorizado



# FPGAs libres: El renacimiento



- Proyecto Icestorm (Mayo, 2015)
- La primera *toolchain* que permiten pasar de Verilog al bitstream usando sólo Herramientas libres

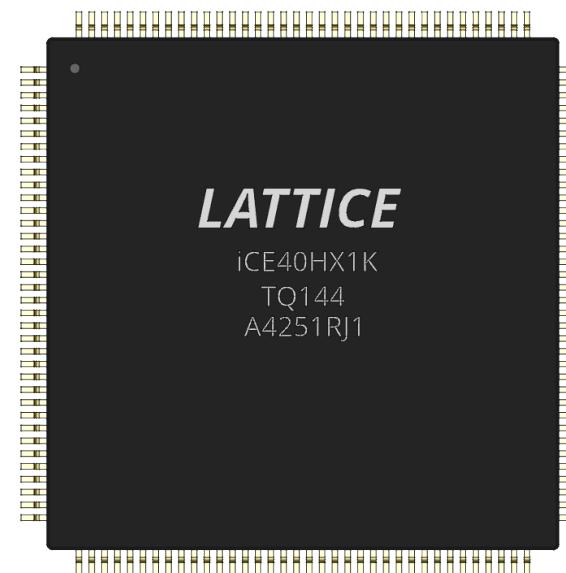
# FPGAs libres

- Definición:

Denominamos **FPGAs libres** a aquellas FPGAs que disponen de una **toolchain totalmente libre**

- **FPGAs libres actualmente:**

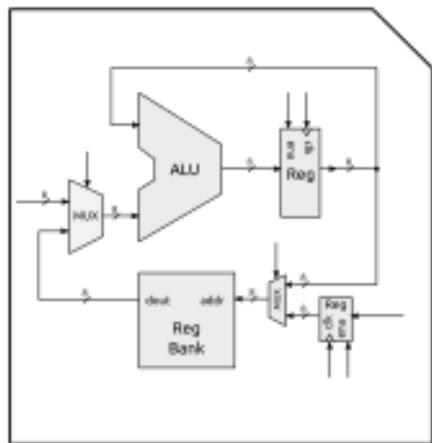
- Familia **Lattice iCE40**
- Sólo Lenguaje Verilog



<http://www.latticesemi.com/Products/FPGAandCPLD/iCE40.aspx>

# Flujo de trabajo

Diseño



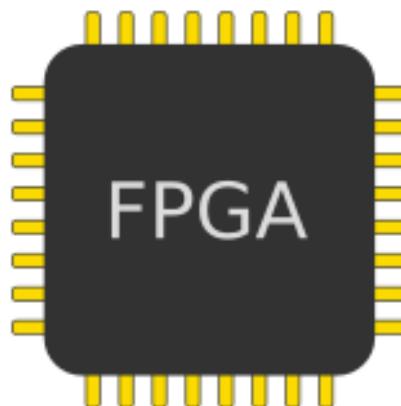
Ficheros HDL

```
module simplez #(  
    parameter BAUD = 'B115200,  
    parameter WIDTH_BELAT = 'T 299uS,  
    parameter ROMFILE = "prog.list",  
    parameter DEBUG_LED5 = 0  
)  
  
input wire clk;  
input wire ntrig_in;  
output wire [3:0] leds;  
output wire stop;  
output wire ta;  
input wire bc;  
  
reg [10:1:0] alu_out;  
reg reg2[2];  
always @ (posedge clk)  
begin  
    if (alu_sp2)  
        alu_out = alu_inc;  
    else if (alu_drd)  
        alu_out = 0;  
    else if (alu_deci)  
        alu_out = reg_2 + alu_inc;  
    else if (alu_hex)  
        alu_out = reg_2 - alu_inc;  
end
```

Síntesis

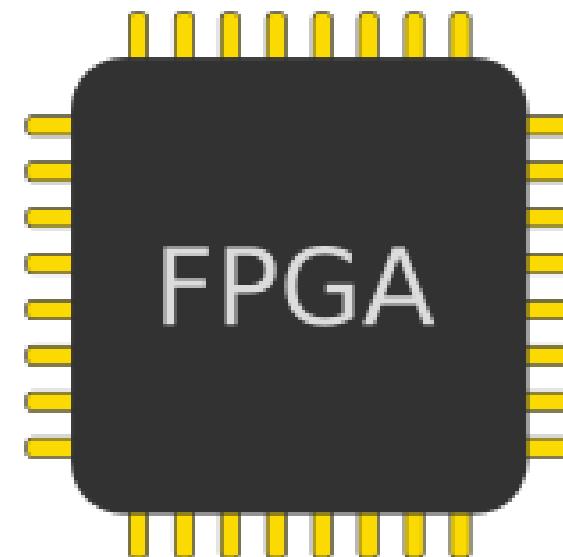
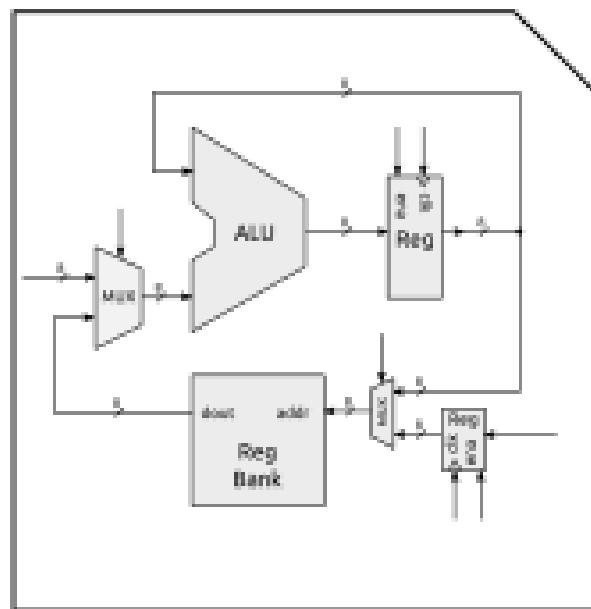
...011000010111001001110011  
011001010100100101101110  
011101000011101000111010  
0010100000110000000101100  
0011001000110000100110001  
001010010011110100100011  
0010100000110000000101100  
0011001000110000000110001  
001010010010110000101000  
001100000010110000110001  
001101010010100100101100  
0010100000110000000101100...

Configuración



Bitstream

# ¡Usando sólo herramientas libres!



# Pila de herramientas libres

Apio IDE

Icestudio

Apio

Icestorm

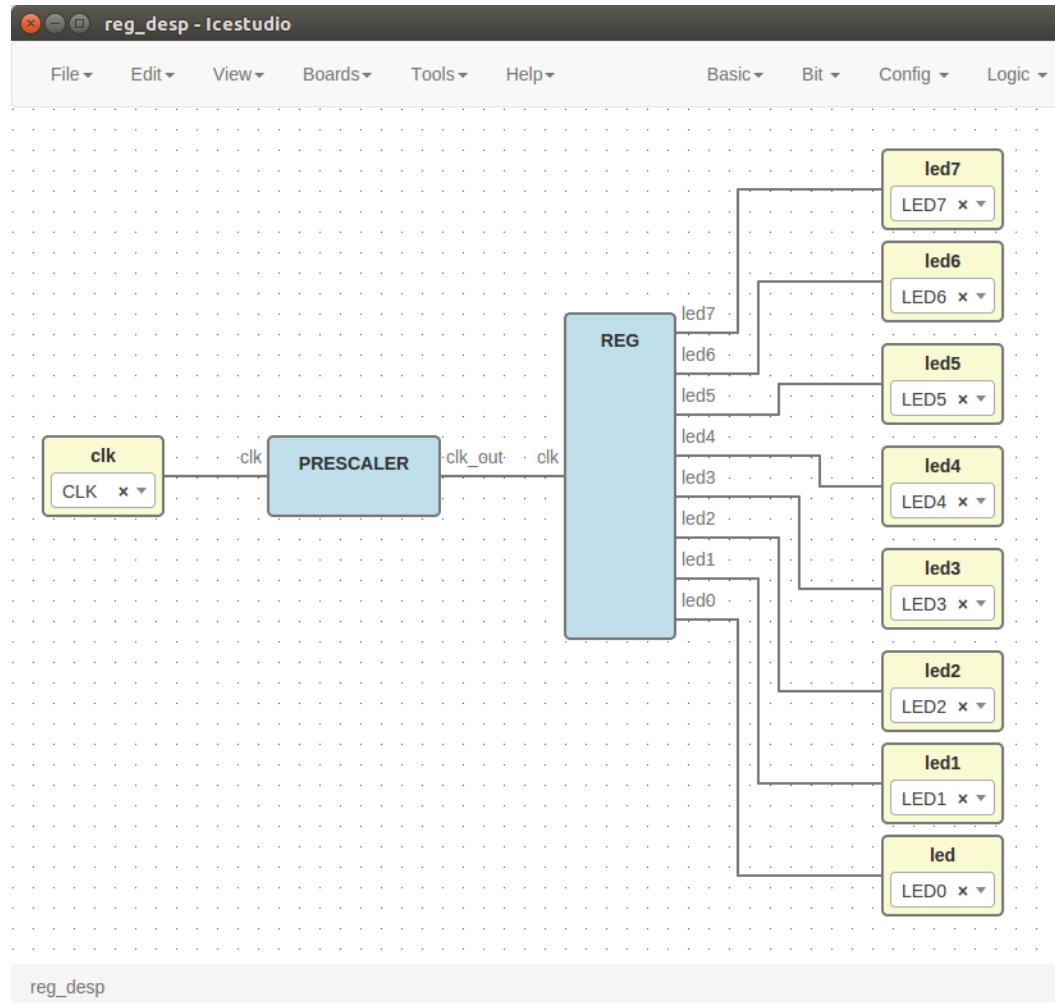
Iverilog

GTKWave



# icestudio

<https://github.com/FPGAwars/icestudio>



- Autor: **Jesús Arroyo**
- Electrónica digital para todos
- Sin conocimientos de verilog
- Herramienta visual
- Traduce a verilog

Demo



# Apio-ide

<https://github.com/FPGAwars/apio-ide>

The screenshot shows the Apio-ide plugin integrated into the Atom code editor. On the left, there's a sidebar with icons for file operations like new, open, save, and delete. The main area has tabs for 'leds.v' (the current file), 'leds.x', and 'leds.pdf'. The code editor displays a Verilog module named 'leds' with several 'assign' statements. Below the code editor is a terminal window showing the build process:

```
apio build
span_4      0 / 6944
span_12     2 / 1440

route time 0.01s
write_txt hardware.asc...
=====
[SUCCESS] Took 1.04 s
```

A green bar at the bottom of the terminal window indicates the build was successful.

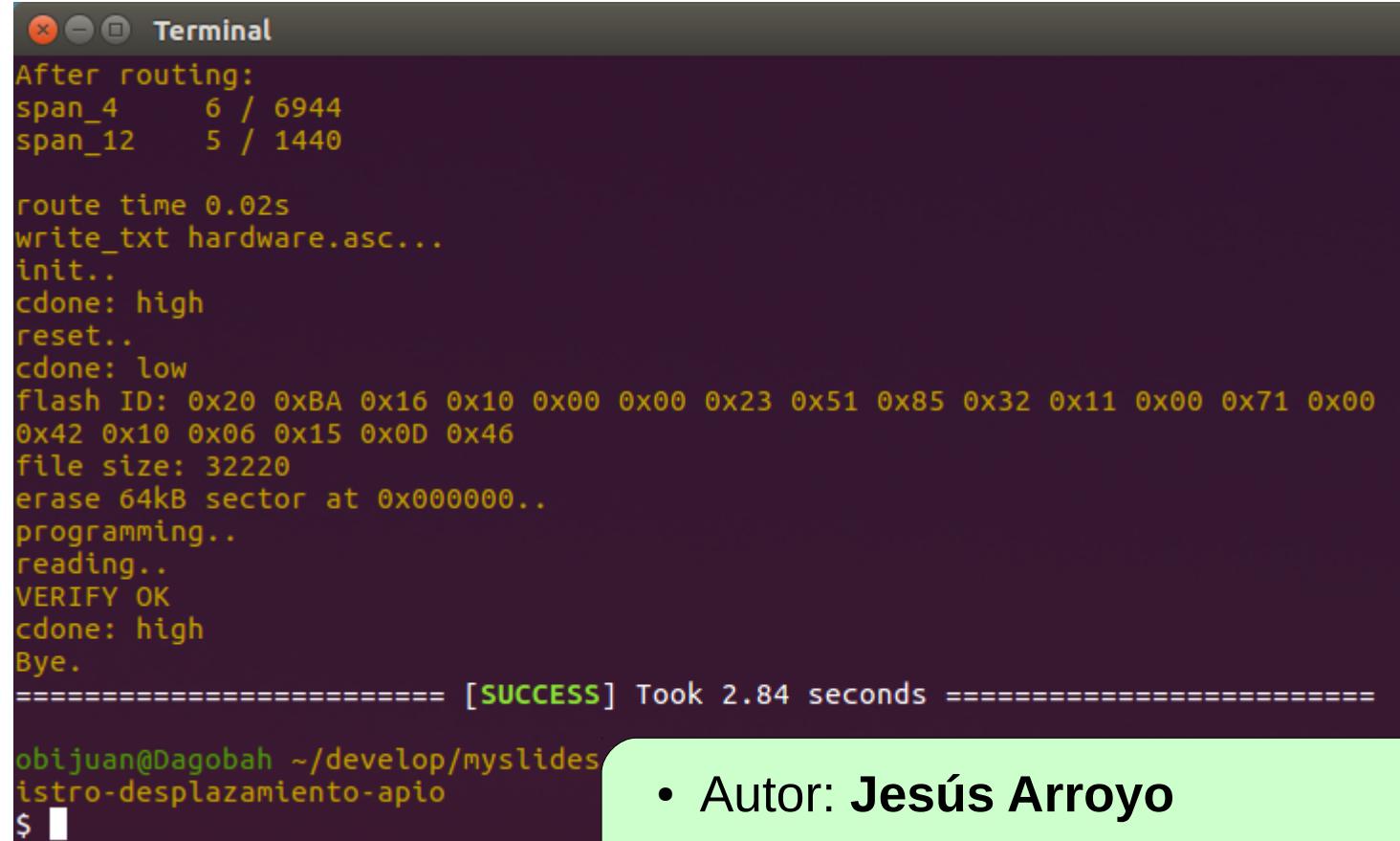
Demo

- Autores: Jesus Arroyo/Obijuan
- Plug-in para Atom
- No línea de comandos
- Llama a apio
- Aplicable a otros IDEs/editores
- Descripción en Verilog



<https://github.com/FPGAwars/apio>

Demo



A screenshot of a terminal window titled "Terminal". The window displays the output of the APIO tool, which includes routing information, file operations like "write\_txt hardware.asc...", and various status messages such as "cdone: high" and "VERIFY OK". The output concludes with a "[SUCCESS]" message. At the bottom of the terminal, the user's session details are shown: "obijuan@Dagobah ~/develop/myslides" and the command "\$".

```
After routing:  
span_4      6 / 6944  
span_12     5 / 1440  
  
route time 0.02s  
write_txt hardware.asc...  
init..  
cdone: high  
reset..  
cdone: low  
flash ID: 0x20 0xBA 0x16 0x10 0x00 0x00 0x23 0x51 0x85 0x32 0x11 0x00 0x71 0x00  
0x42 0x10 0x06 0x15 0x0D 0x46  
file size: 32220  
erase 64kB sector at 0x0000000..  
programming..  
reading..  
VERIFY OK  
cdone: high  
Bye.  
===== [SUCCESS] Took 2.84 seconds =====  
  
obijuan@Dagobah ~/develop/myslides  
istro-desplazamiento-apio  
$
```

- Autor: **Jesús Arroyo**
- Multiplataforma (Linux, Mac, Windows, Raspberry)
- Línea de comandos
- Programado en python
- Multiplaca: icestick, icezum, icoboard, go-board

# Bajo nivel: Icestorm

<http://www.clifford.at/icestorm/>

<https://github.com/cliffordwolf/icestorm>

- **Ficheros:** *cont8.v icezum.pcf*

- **Síntesis:**

Demo

```
$ yosys -p "synth_ice40 -blif cont8.blif" cont8.v
```

- **Emplazado y rutado:**

```
$ arachne-pnr -d 1k -p icezum.pcf cont8.blif -o cont8.asc
```

- **Empaquetado:**

```
$ icepack cont8.asc cont8.bin
```

- **Carga en FPGA:**

```
$ iceprog cont8.bin
```

- Autor: **Clifford Wolf**
- Herramientas programadas en C/C++
- Bajar del repo y compilar
- Línea de comandos
- Se usan típicamente junto con make

**PARTE III:**

**¿Cómo empiezo?**

# Paso 1: Consigue una placa Con FPGA libre

Icestick



iCE40-HX8K Breakout Board

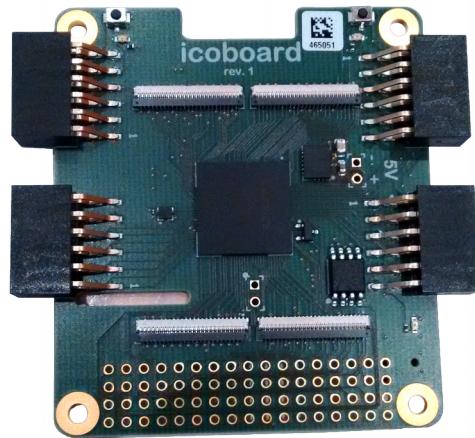


Go-board



- Conexión directa al PC (USB)
- Soportadas por Apio/Icestudio

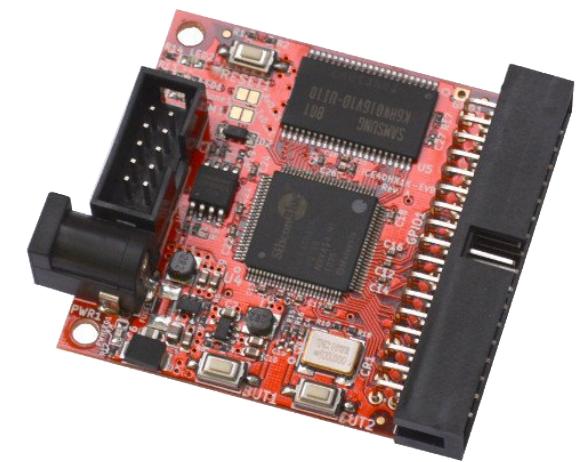
icoboard



Mystorm



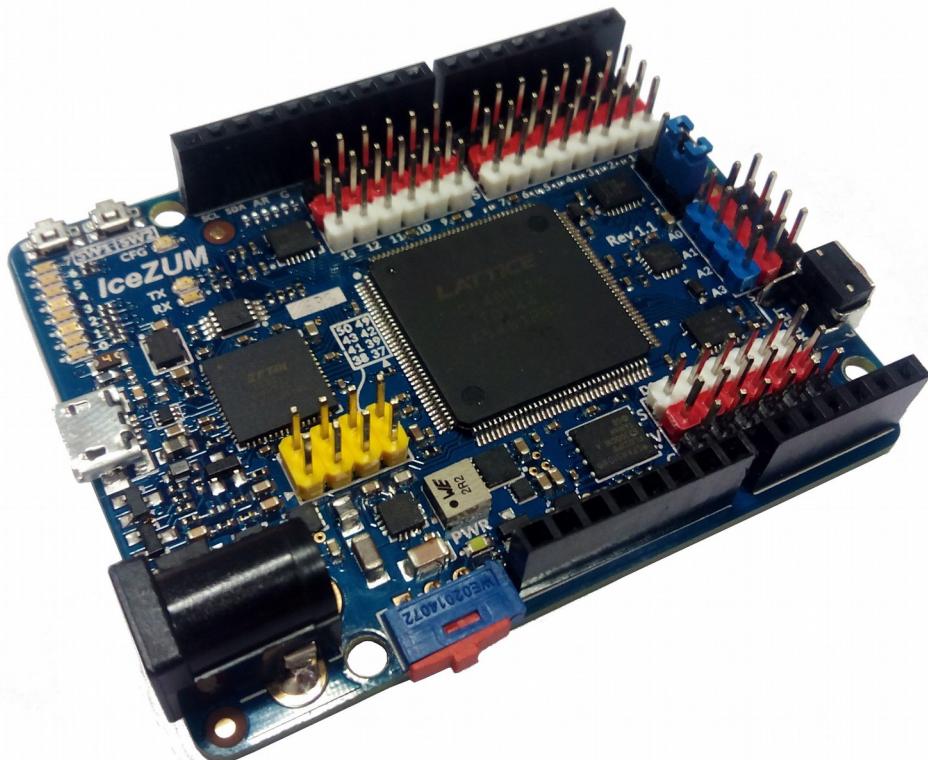
iCE40HX1K-EVB



- Conexión a Raspberry PI
- Soportada por Apio/Icestudio

NO Soportadas por Apio/Icestudio

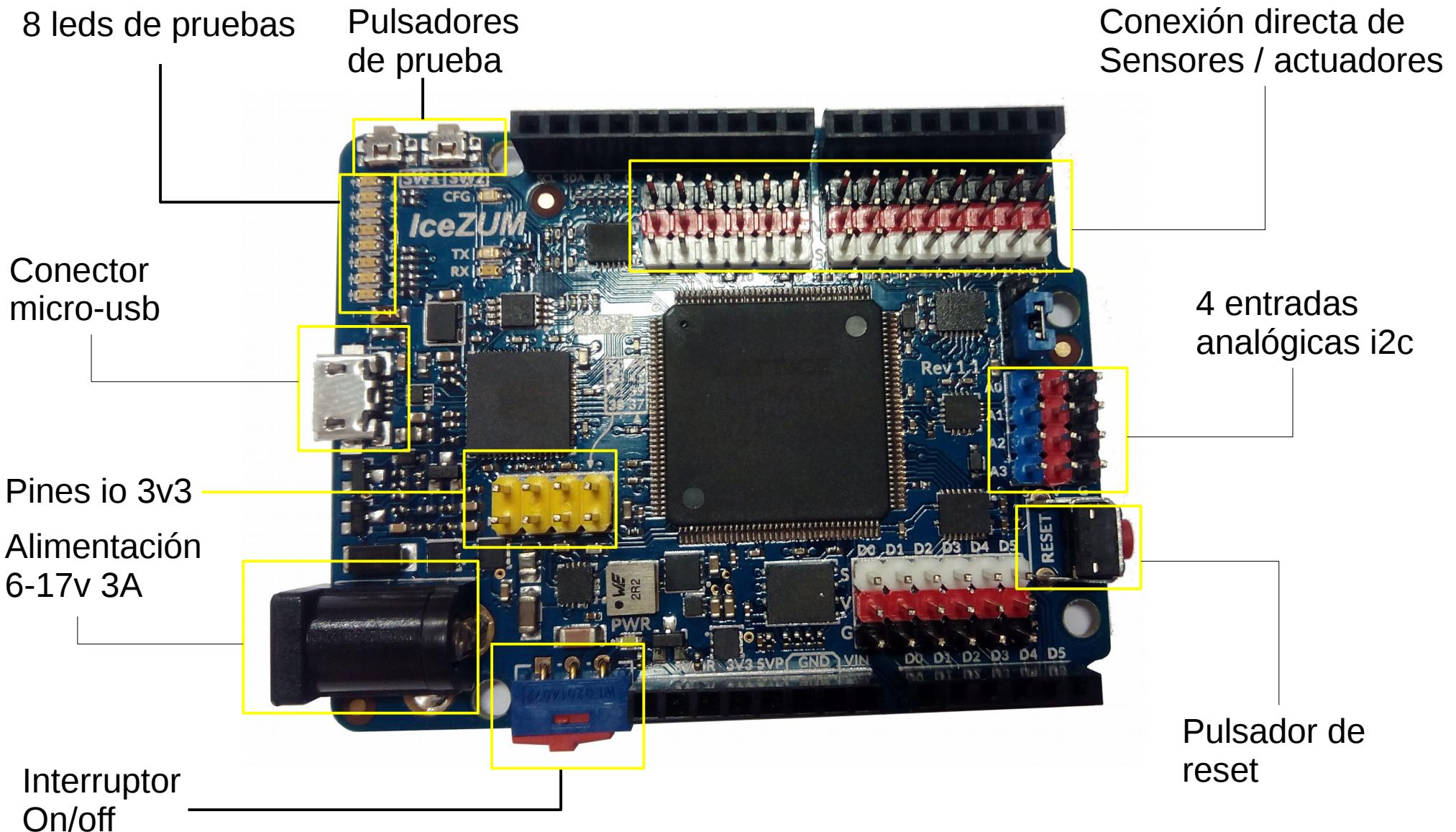
# Icezum Alhambra v1.1



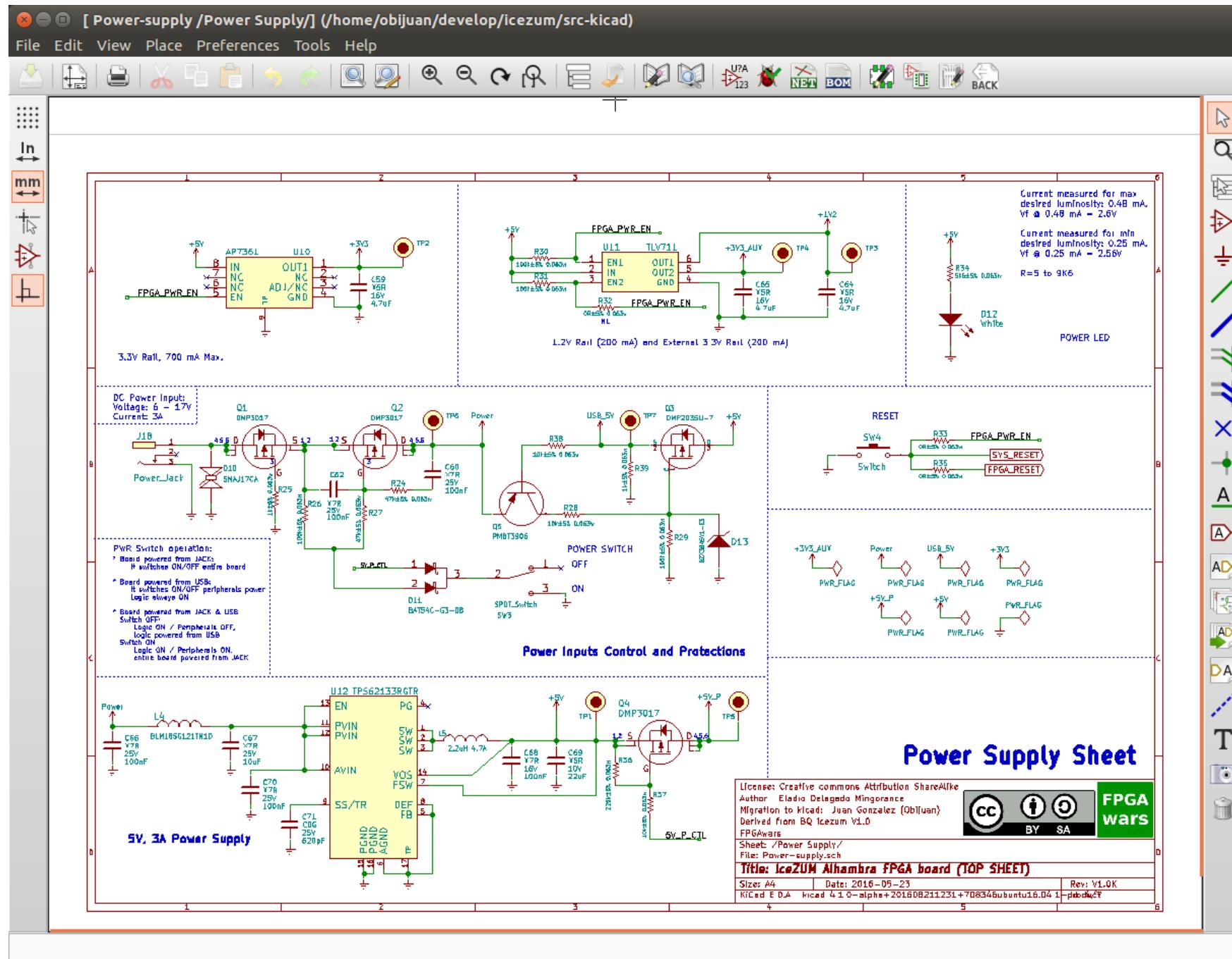
- Autor: **Eladio Delgado**
- Diseñada en Pinos del Valle (Granada)
- Arduino de las FPGAs
- Compatible Arduino
- Fácil conexión de circuitos externos/sensores/servos
- Reutilización de los shields de arduino
- 20 entradas/salidas de 5v
- 3A corriente de entrada
- Perfecta para hacer robots

<https://github.com/FPGAwars/icezum/wiki>

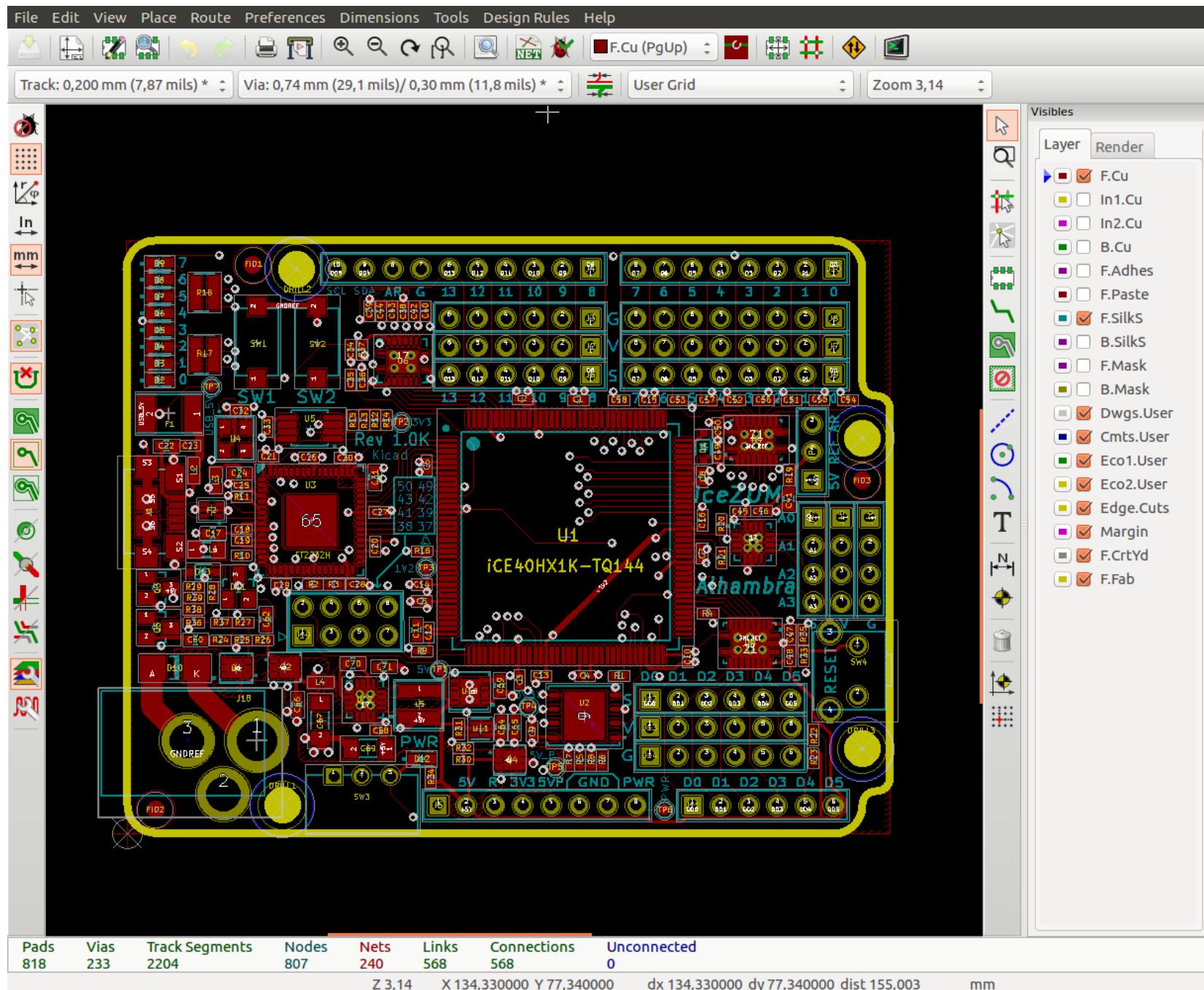
# Icezum Alhambra v1.1



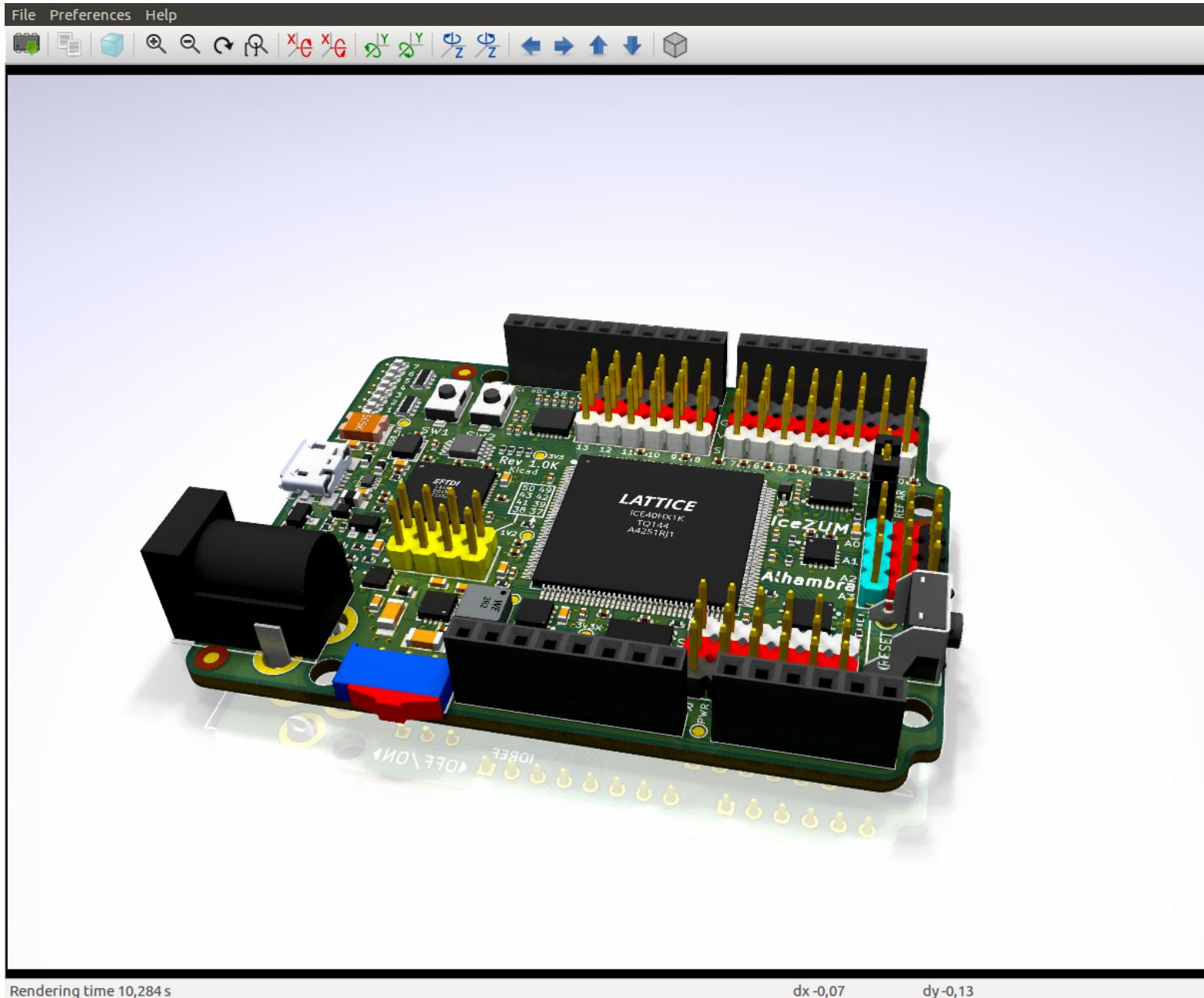
# Icezum Alhambra v1.0K en Kicad



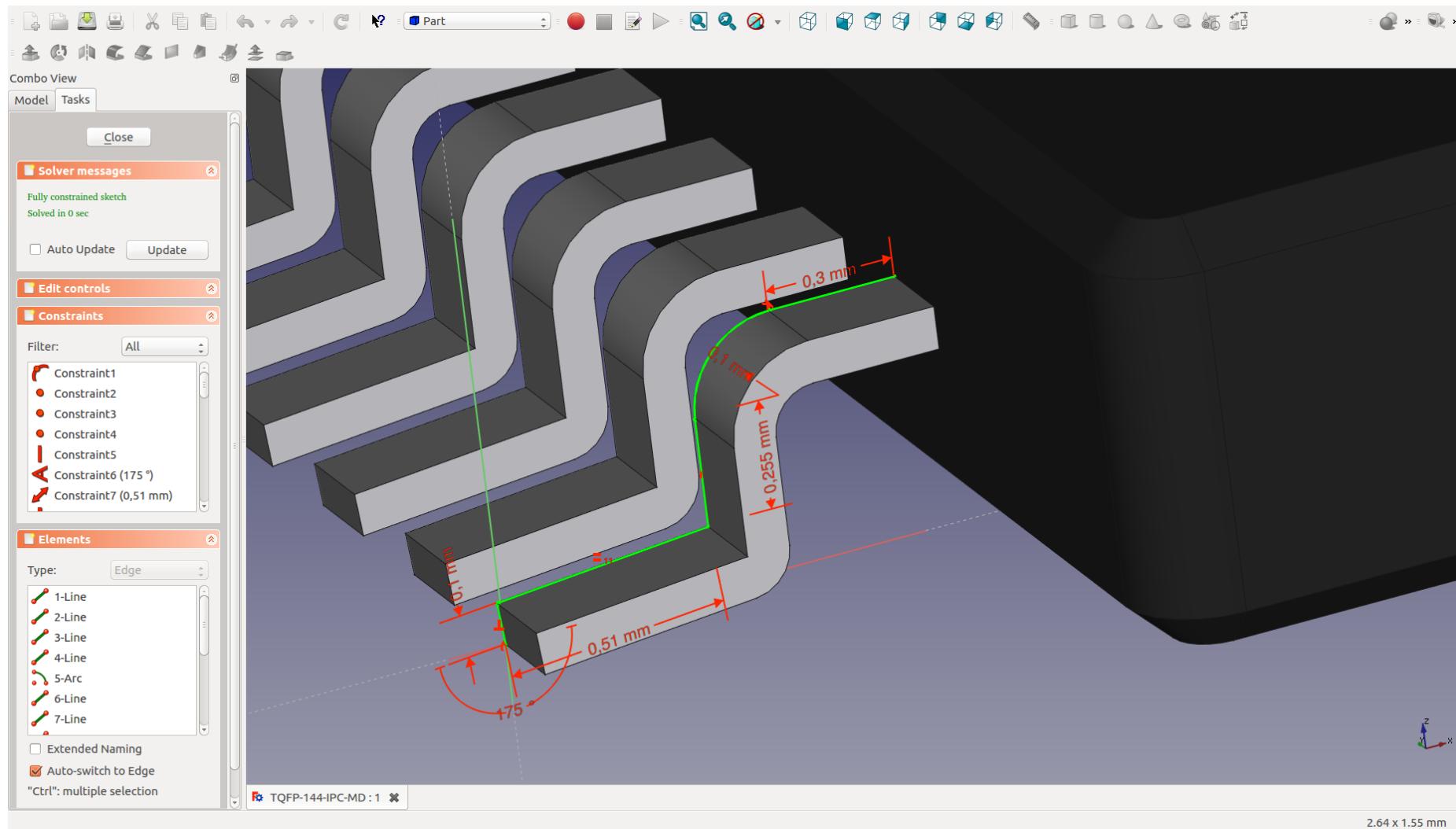
# Icezum Alhambra v1.0K en Kicad



# Icezum Alhambra v1.0K en Kicad

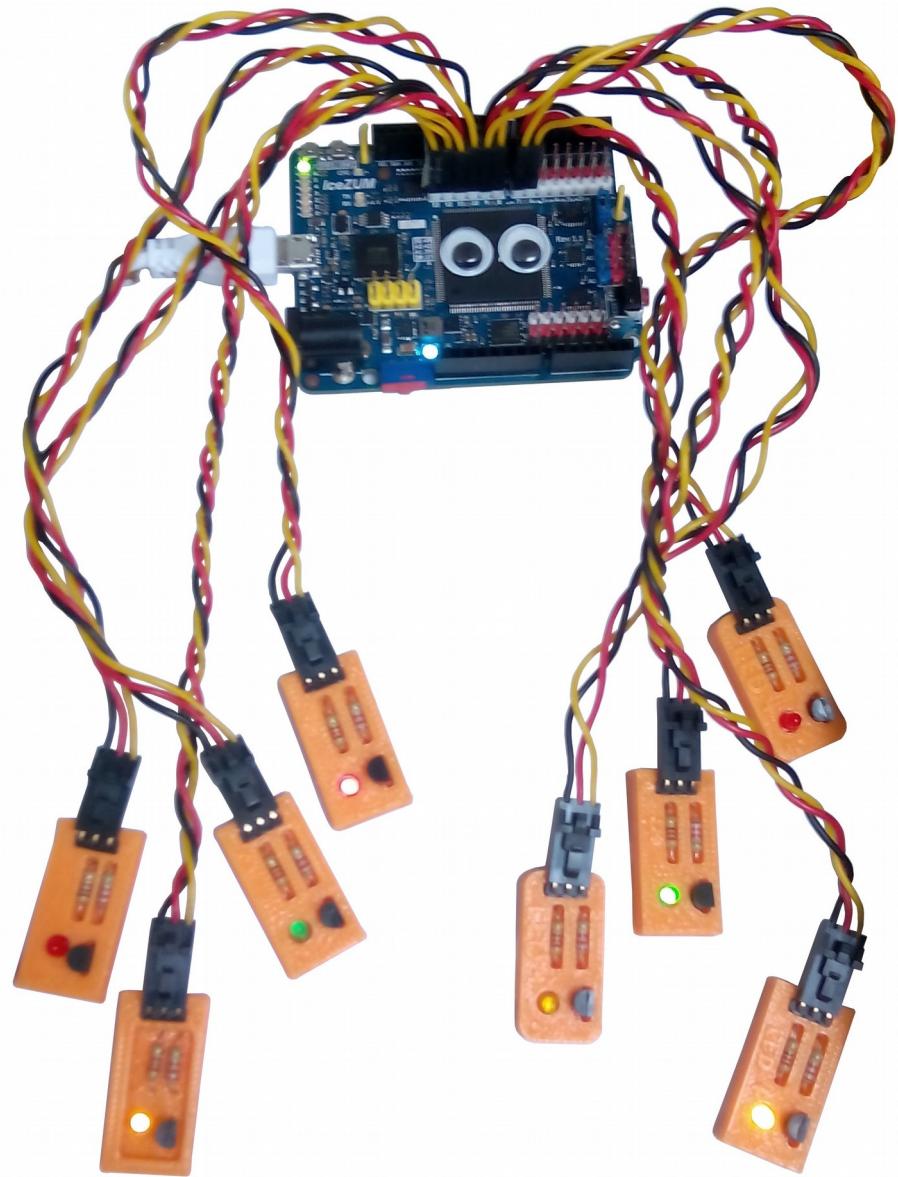


# ¡Fuentes de componentes en FreeCAD!



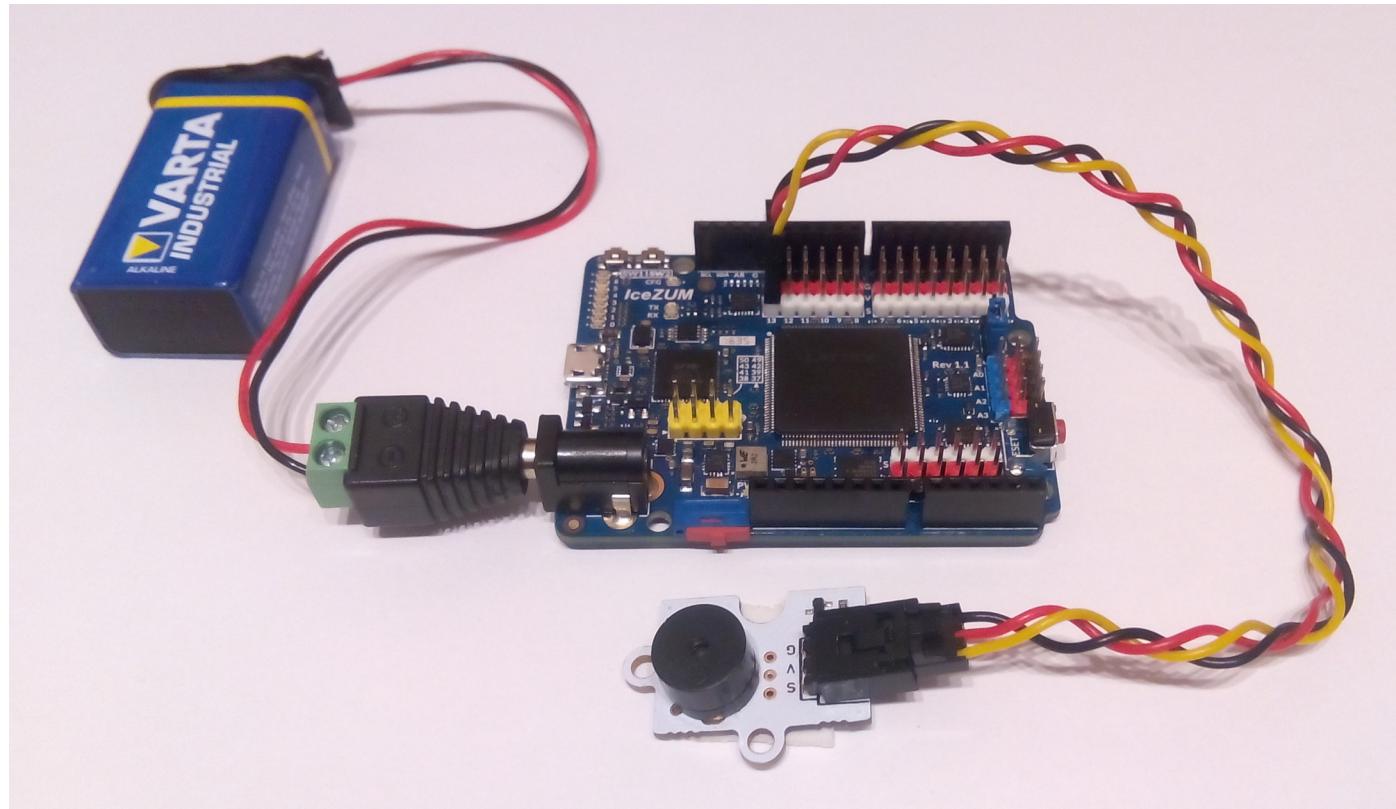
<https://github.com/FreeCAD/FreeCAD-library>

# Rasty



- FPGA Fun!
- 8 Alhambra-leds
- Alimentación: power bank

# Sonidos simples con Zumbador



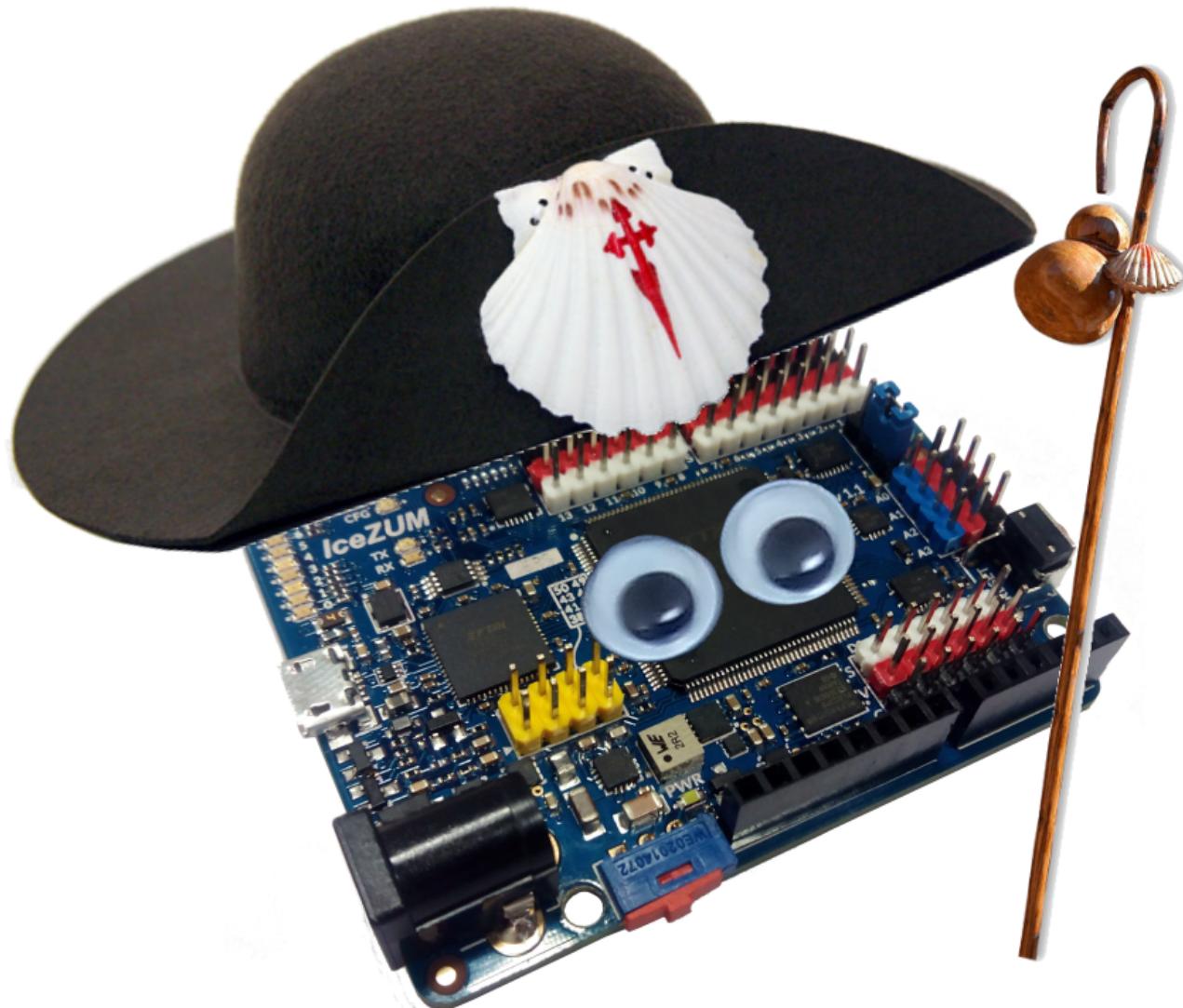
- Conexión a una placa con Zumbador
- Melodías básica
- ¡Fácil meter varios canales!

# Larby: Robot modular



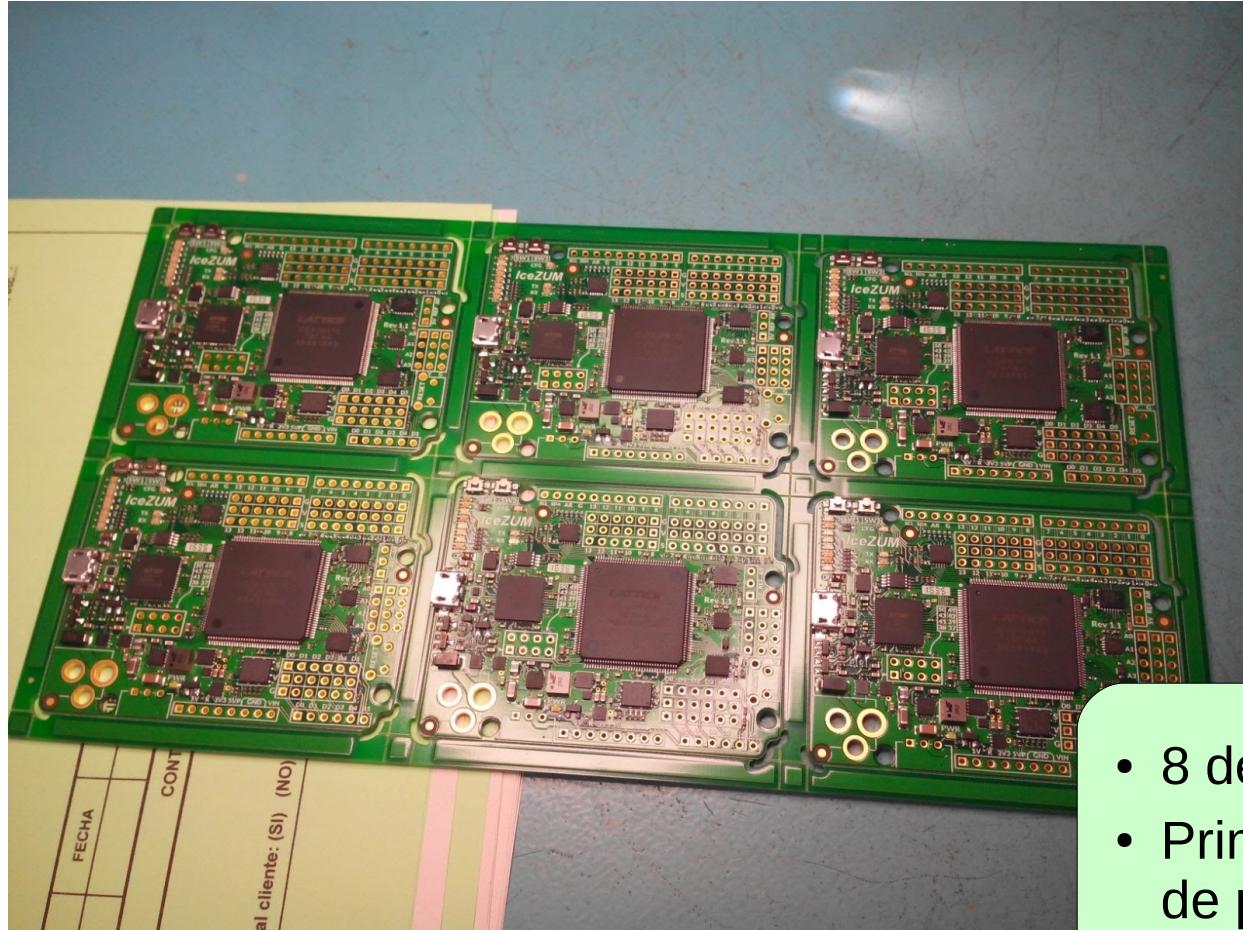
- Servos conectados directamente a Icezum Alhambra
- Configuración mínima pitch-pitch
- Módulo impresos en 3D

# Icezum Alhambra peregrina



- A.k.a **CalmaSAV**
- Empieza:  
[Rinconingenieril.es](http://Rinconingenieril.es)
- Ámbito: Toda España
- Coordinado desde la lista  
de FPGAwars

# ¡Últimas noticias!



Vídeo

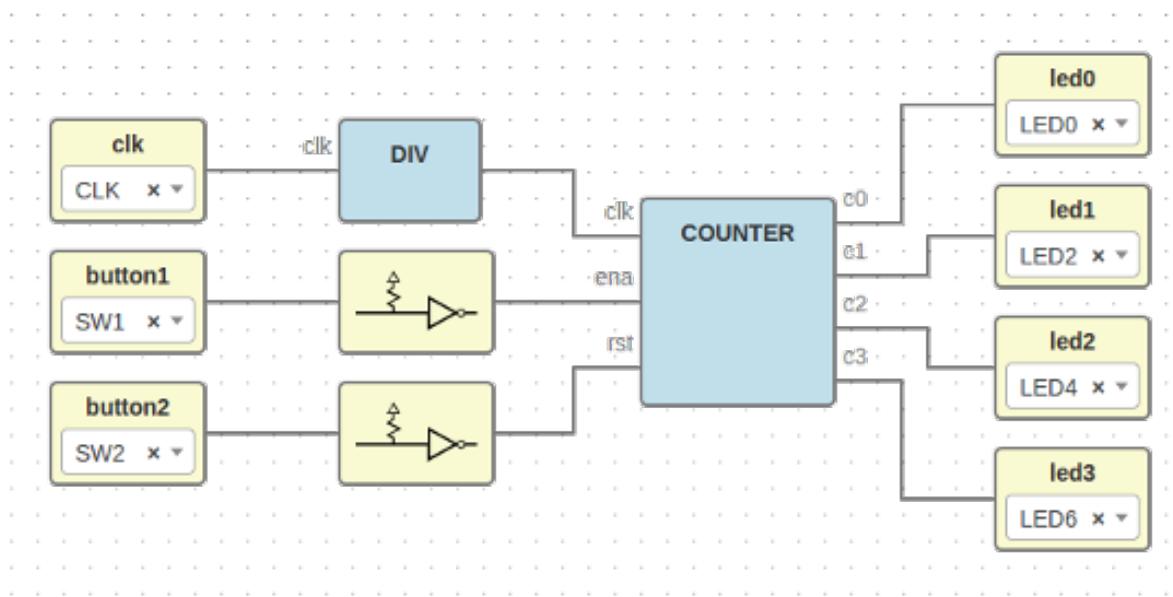
- 8 de Noviembre de 2016
- Primeras alhambra v1.1 de producción, para el Crowdfunding de 100 unidades
- Vídeo de montaje

# Paso 2: Instálate Icestudio/Apio



icestudio

APiO

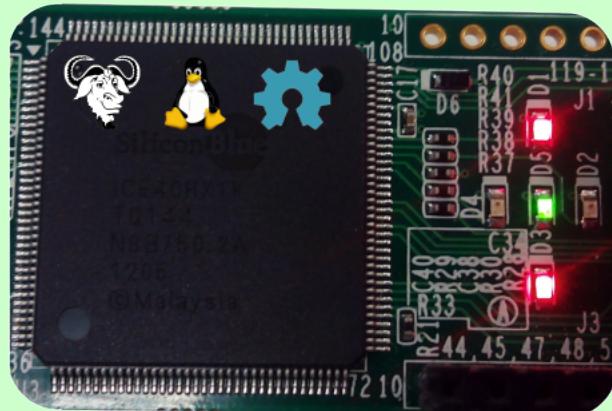


<https://github.com/FPGAwars/icestudio>

<https://github.com/FPGAwars/api>

# Paso 3: Aprende diseño digital y Verilog

Tutorial: Diseño Digital para FPGAs, con herramientas libres



- Hace 1 año (obsoleto)
- Herramientas de bajo nivel (make)
- Tarjeta icestick
- Útil mientras hago uno más actualizado :-)

<https://github.com/Obijuan/open-fpga-verilog-tutorial/wiki>

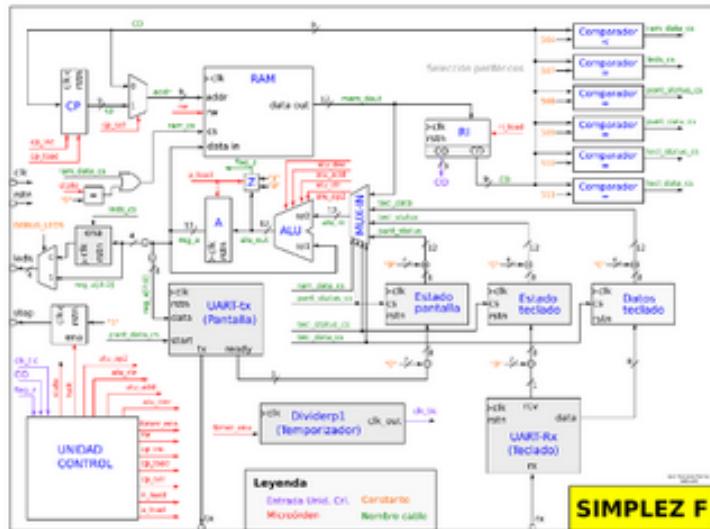
# Paso 4: Apúntate a la comunidad FPGAwars



- Comunidad para **compartir conocimiento** relacionado con **FPGAs libres**
- Es el **clonewars** de las FPGAs, pero en modesto :-)
- Idioma: Castellano
- 194 miembros
- Cualquier pregunta / comentario / sugerencia → Correo a la lista :-)

<http://fpgawars.github.io/>

# Paso 5: Haz tus proyectos con FPGAs libres

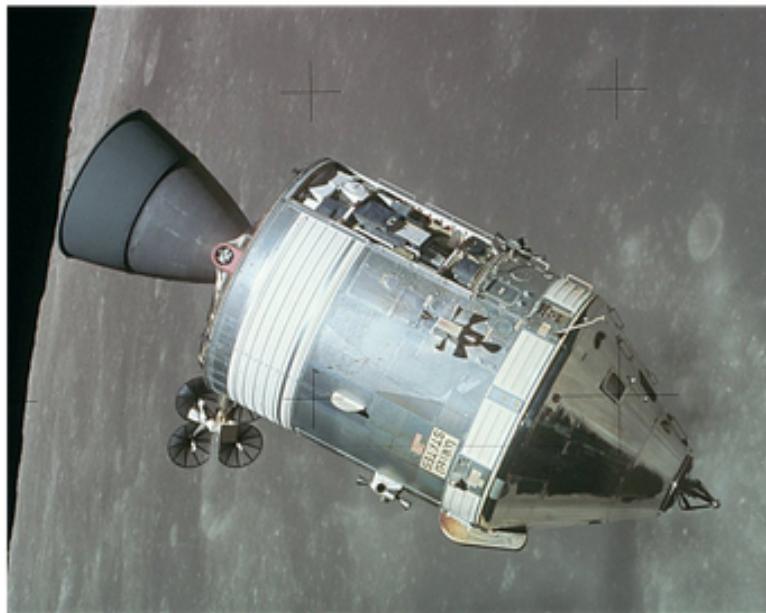


SIMPLEZ



*Educational CPU in Verilog*

<https://github.com/Obijuan/simplez-fpga/wiki/Procesador-SIMPLEZ-F>



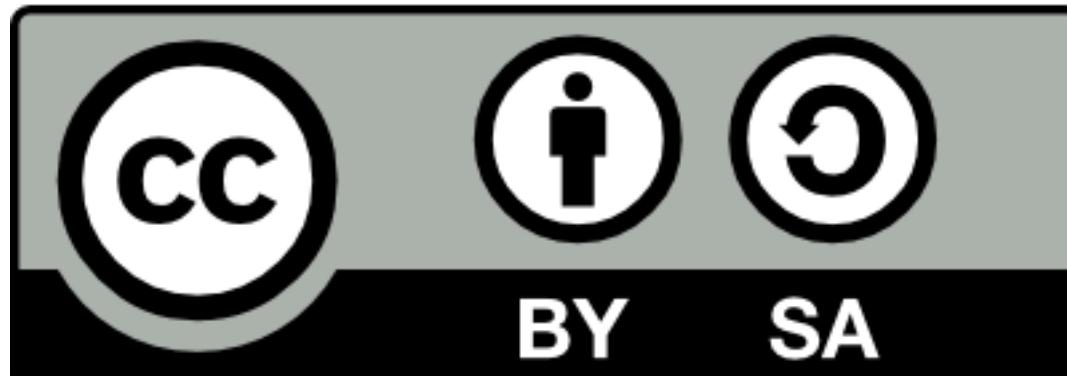
APOLLO CPU CORE



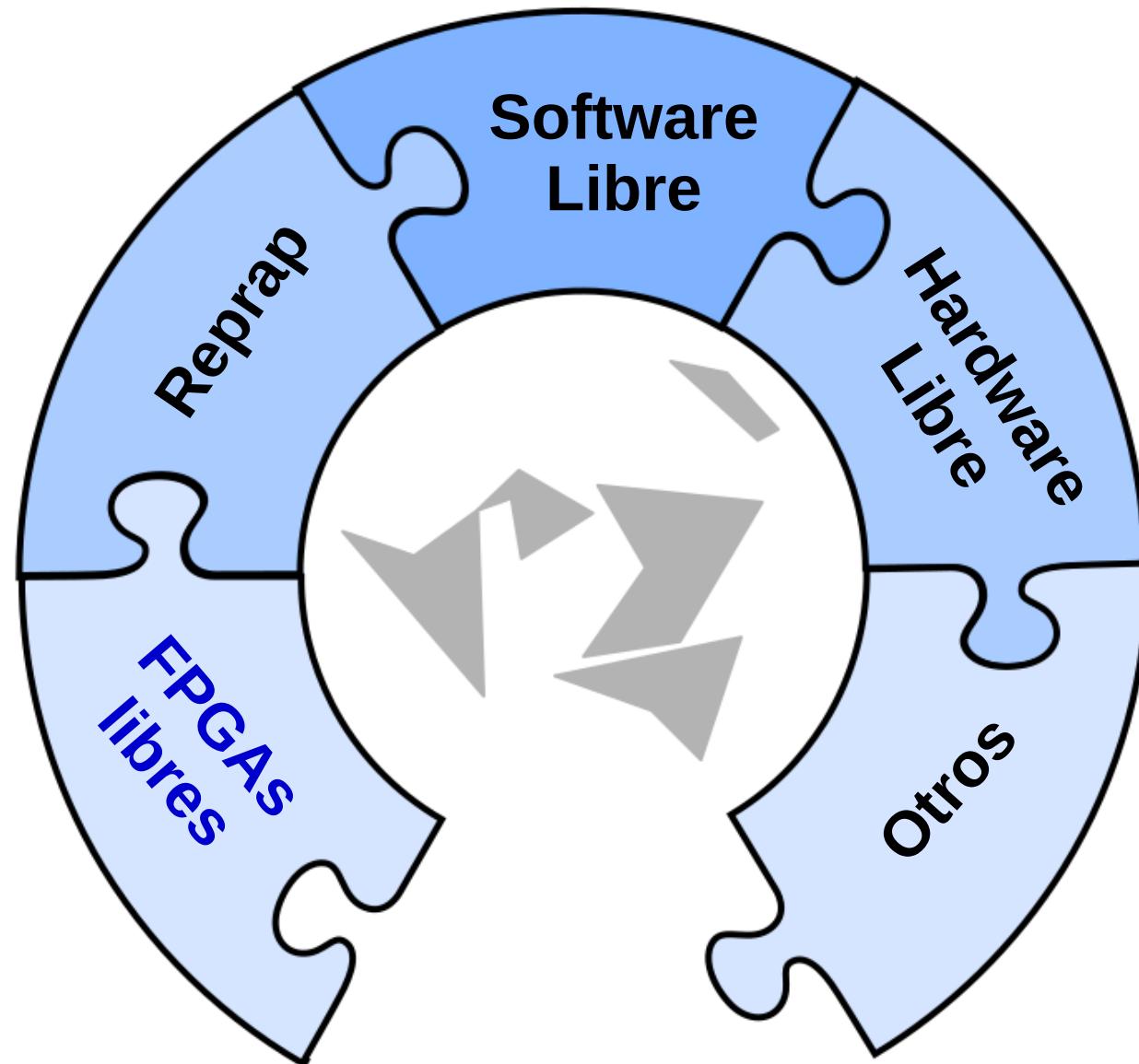
*Apollo CPU in Verilog*

<https://github.com/Obijuan/ACC/wiki>

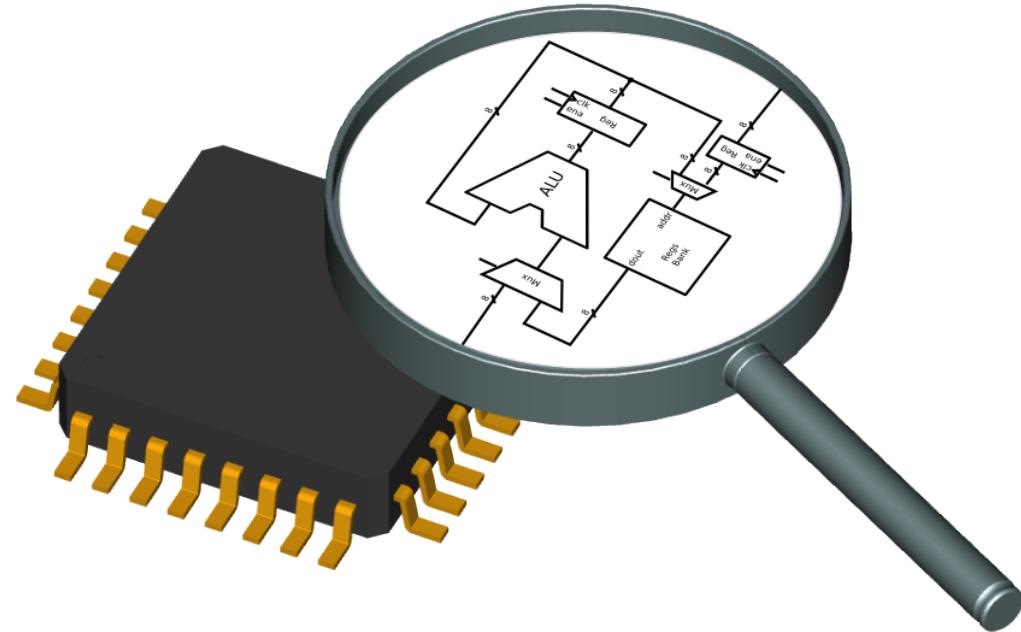
# Paso 6: ¡Comparte con la comunidad!



# FPGAs en Patrimonio Tecnológico de la humanidad



# FPGAs libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>