

---

# **Proyecto ICARO, Documentación**

## **Documentation**

***Release 0***

**Valentin Basel**

February 08, 2016



## CONTENTS

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	software libre en la educación . . . . .	3
1.2	hardware libre . . . . .	4
<b>2</b>	<b>placa robotica np07</b>	<b>7</b>
2.1	características tecnicas . . . . .	7
2.2	listado de componentes . . . . .	8
2.3	herramientas . . . . .	10
2.4	fabricacion . . . . .	15
<b>3</b>	<b>icaro-bloques</b>	<b>37</b>
3.1	barra de herramientas . . . . .	37
3.2	barra de componentes . . . . .	37
3.3	editor de bloques . . . . .	37
3.4	editor de codigo fuente . . . . .	37
<b>4</b>	<b>trabajando con icaro-bloques</b>	<b>39</b>
4.1	compilando un proyecto . . . . .	39
4.2	cargando el firmware a la placa . . . . .	39
4.3	referencias de los bloques . . . . .	39
<b>5</b>	<b>guia de ejemplos</b>	<b>41</b>
5.1	usando LEDS . . . . .	41
5.2	usando motores CC . . . . .	41
5.3	usando comunicación CDC . . . . .	42
5.4	usando una pantalla LCD . . . . .	42
5.5	usando un sensor hc-sr04 . . . . .	42
<b>6</b>	<b>python</b>	<b>45</b>
6.1	cargando firmware tortucaro . . . . .	45
6.2	apicaro . . . . .	45
6.3	referencia apicaro . . . . .	45
<b>7</b>	<b>clemente</b>	<b>47</b>
7.1	activando servidor clemente . . . . .	47
7.2	cargando firmware PILAS . . . . .	47
7.3	graficador . . . . .	47



Contents:



## INTRODUCCIÓN

La robótica pedagógica, dado su carácter multidisciplinario, permite el abordaje de conocimientos variados como la electrónica, informática, física y matemática mediante la construcción de un juguete-objeto como puede ser un robot. El desarrollo de estos juguetes-objetos implica una experiencia que contribuye a expandir la creatividad y el pensamiento reflexivo y científico de los alumnos, en relación a la formulación de hipótesis, la experimentación y la elaboración de conclusiones, los cuales al enfrentarse a un “problema” dado, aprenden a experimentar, diseñar y resolver situaciones de carácter constructivista. En el proceso de “pensar el robot”, se generan las condiciones de apropiación del conocimiento por parte del alumno. Se trata de otorgar a los alumnos un rol activo en sus aprendizajes, colocándolos como diseñadores de sus propios proyectos y constructores de conocimientos. El uso de software libre en el ámbito escolar, permite tener control sobre las características del mismo, permitiendo adaptarlo a las necesidades concretas del ámbito escolar y las realidades socio-económicas de la institución, es neutro frente a fabricantes (el alumno no es un “potencial cliente”) y todo el material usado puede ponerse a disposición de otros docentes.

### 1.1 software libre en la educación

Antes de explicar que es el software libre, necesitamos conocer algunos aspectos claves del diseño y creación de cualquier programa, como el concepto de código fuente y como afecta a nuestra libertad el hecho de no poder tener acceso al mismo.

Una computadora es básicamente una máquina electrónica que recibe y procesa datos para convertirlos en información útil, y esos datos apenas son cadenas de ceros y/o unos. Por lo tanto una computadora solo entiende código binario. Para hacer un programa más o menos complejo, necesitamos muchísimos “ceros y unos”, lo que hace extremadamente complejo entender y programar en “lenguaje de máquina”. Para ello se diseñaron “lenguajes” más parecidos a lo que un humano podría entender y mediante un software compilador podríamos convertirlo en archivo binario que nuestra computadora usa.

Podemos considerar al software como el Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora por lo tanto es una parte central de una computadora porque es el que controla todos los procesos electrónicos de la misma. Para que una computadora haga algo, necesita un programa, y este tuvo que ser creado escribiendo en un lenguaje especial las instrucciones que dicho programa ejecutara, esto se conoce como Código fuente.

El código fuente es un texto escrito en un lenguaje de programación específico que puede ser leído por un programador, se puede considerar como un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora. El código fuente está escrito en primera instancia por un programador en un formato de texto plano y para que este archivo de texto pueda ser interpretado por una computadora, tiene que ser procesado por un software llamado «compilador», el cual en su definición más genérica, es un programa que toma como entrada un texto escrito usando la semántica de cierto lenguaje y produce como salida el texto de un programa en otro lenguaje. De esta manera podemos obtener de un lenguaje natural similar al lenguaje humano, un archivo en lenguaje de código máquina (código binario).

Para los programadores, leer el código máquina o código binario es una actividad muy difícil, por eso para poder modificar o estudiar un programa, se vuelve fundamental tener el código fuente original. A causa de que muchas

empresas desarrolladoras de software no permiten conocer ni estudiar el código fuente de sus programas (software privativo), en el año 1985 Richard Stallman y otros entusiastas fundaron la Free Software Foundation con el objetivo de promover el uso del software libre.

Cuando Hablamos de software libre, nos referimos a todo software que por elección manifiesta de su autor, puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras. Para eso se definió 4 «libertades» básicas que tiene que tener todo software para poder ser considerado de software libre:

1. La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).
2. La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
3. La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
4. La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

## **1.2 hardware libre**

Menos difundido que el concepto de software libre, la idea del «hardware libre» (o mas precisamente, Hardware de fuentes abierta) trata de hacer un paralelismo entre las «cuatro libertades» del software libre, aplicadas a el diseño y protocolos de un dispositivo fisico.

Decimos que el hardware es libre en tanto el diseño de fabricación y el software con el que fueron echos esos diagramas, sean distribuidos con licencias libres. Si bien, por su naturaleza física, un dispositivo (electrónico o mecánico) no puede ser equiparado a un programa de software, y por lo tanto, no es correcto equiparar las 4 libertades que se aplican al software libre a un objeto físico único, si se puede especificar que su diseño, esquemas, planos y demás componentes no-fisicos, tengan un tipo de licencia libre. De esta forma un diseño electrónico, puede ser estudiado, modificado y distribuido bajo los principios de las licencias GNU GPL (GNU General Public License) o similares.

Un PCB (Printed Circuit Board por sus siglas en ingles) es la superficie constituida por pistas de material conductor sobre una base no conductora, se utiliza para interconectar los distintos componentes electronicos (resistencias electricas, capacitores, microcontroladores etc.) y generar un circuito por donde circule la corriente electrica, esta «tarjeta de circuito impreso» funciona como soporte fisico y esquema de conexión electronico.

Cuando se diseña un PCB, generalmente se necesitan 3 diagramas , el plano esquematico, el diseño del PCB y los ficheros de fabricación archivos (GERBER). Con esa información podemos fabricar nuestra propia versión de la placa donde van soldados los componentes electronicos.

La ventaja del hardware libre es que permite adaptar nuestro diseño a las distintas realidades locales donde se implementara la fabricación de dicho dispositivo. De esta forma permite tener cierta independencia de un fabricante en particular, y hasta permitir la fabricación a pequeña escala (si el diseño asi lo permite), ademas de fomantar la industria nacional y pequeñas PYMES o cooperativas que fabriquen el hardware, aprovechando una comunidad de usuarios que documentan y dan soporte al mismo.

Un caso muy significativo de la movida del hardware libre, son las «impresoras 3d», dispositivo mecanico que permite fabricar componentes de 3 dimensiones mediante superponer capas de plastico derretido en una plataforma (usando distintos tipos de plasticos y hasta materiales como cemento o metales fundidos en maquinas mas experimentales) para lograr mediante la adición sucesiva de capas de material, un objeto tri dimensional con cierta consistencia y dureza.

Si bien la tecnologia de fabricación automatizada de piezas mecanicas existe desde hace mucho (los sistemas CNC o control numérico por computadora por ejemplo), el proyecto REPRAP se propuso fabricar una maquina-herramienta de bajo costo capas de «auto replicarse» fabricando las piezas para otras maquinas iguales. los creadores del proyecto visionan la posibilidad de distribuir a bajo costo máquinas RepRap a personas y comunidades, permitiéndoles crear (o descargar de Internet) productos y objetos complejos sin la necesidad de maquinaria industrial costosa. En ese

sentido la idea del proyecto es poder diseñar una impresora que pueda ser facilmente fabricada por otras personas, sin necesidad de componentes industriales o maquinaria especial para su fabricación. Obviamente, implica una desición de compromiso entre cierta calidad de componentes y la facilidad de obtenerlos a pequeña escala, ademas de los conocimientos minimos necesarios para poder fabricar una impresora (desde conocimientos sobre electronica a cuestines mecanicas y cierta habilidad manual), por eso, comenzo a proliferar pequeñas compañias o micro emprendimientos de personas que fabrican y venden impresoras 3d basadas en los modelos de reprap (prusa I2 o tambien prusa I3). De esta forma un proyecto global que nace gracias a internet, genera trabajo para comunidades en distintas partes del mundo.

En cierta forma, se podria decir que el hardware libre rescata la cultura del «hagalo usted mismo» (o DIY por sus siglas en ingles), aunque hay proyectos comerciales muy exitosos (como el caso de ARDUINO® y RASPBERRY PI®) que son de carácter mas industrial y que sin embargo aprovechan la ventaja de liberar sus diseños para contar con ua gran scomunidad de usuarios que utilizan su hardware y crean documentación o tutoriales a traves de internet.



---

CHAPTER  
**TWO**

---

## **PLACA ROBOTICA NP07**

El hardware *np07*

### **2.1 caracteristicas tecnicas**

bla

## 2.2 listado de componentes

Cantidad	Componente	Ubicación	Imagen
11	Resistencias 470 Ohm - 1/4W	R1 R2 R3 R4 R5 R6 R7 R8 R9 R12 R17	
5	Resistencias 10k Ohm - 1/4W	R11 R13 R14 R15 R16	
2	Capacitores Cerámicos 22pF	C2 C3	
5	Capacitores Cerámicos 0.1uF	C9 C10 C11 (C12 C13)*	
1	Capacitor Cerámico 220nF	C1	
1	Capacitor Electrol. 10uF 16V	C5	
4	Capacitor Electrol. 100uF	C4 C6 C7 C8	
3	Diodos 1N4007	D9 D12 D14	
11	Leds difusos 5mm	D1 D2 D3 D4 D5 D6 D7 D8 D10 D11 D12	
1	Conector USB hembra Tipo B	J1	

Cantidad	Componente	Ubicación	Imagen
1	Push Button (Soft Touch)	SW2	
1	Regulador de Voltaje LM7805	U4	
1	Regulador de Voltaje 78L05	U5	
7	Borneras Dobles	P8 P9 P10 P11 P12 P13 P14	
1	Zócalo de 8x2 Pines	U3	
1	Zócalo de 20x2 Pines	U2	
1	Zócalo de 9x2 Pines	P6	
1	Cristal de 20Mhz	X1	
2	Tira Postes Macho de 40 Pines	K2 K3 K4 K5 K6 SW1 SW3 K1 K8 P4	
1	Tira de Postes Hembra de 40 Pines	P1 P7 P5 P15 P16 P17 P18	
1	Driver L293D (Puente H)	U3	
1	Integrado ULN2803	P6	
1	Microcontrolador PIC18F4550	U2	
4	jumper	SW1 SW3 K1 K8	

## 2.3 herramientas

Las herramientas que necesitamos para armar una placa robotica *np07* son faciles de conseguir y muy comunes para cualquier hobbista de la electronica.



Figure 2.1: Soldador

Un soldador eléctrico o de estaño, también conocido como cautín, es una herramienta eléctrica usada para soldar. Funciona convirtiendo la energía eléctrica en calor, que a su vez provoca la fusión del material utilizado en la soldadura, como por ejemplo el estaño.



Figure 2.2: Estaño

El estaño que se utiliza en electrónica tiene alma de resina con el fin de facilitar la soldadura. Para garantizar una buena soldadura es necesario que tanto el estaño como el elemento a soldar alcancen una temperatura determinada, si esta temperatura no se alcanza se produce el fenómeno denominado soldadura fría. La temperatura de fusión depende de la aleación utilizada, cuyo componente principal es el estaño y suele estar comprendida entre unos 200 a 400 °C.

En realidad, el término “estaño” se emplea de forma impropia porque no se trata de estaño sólo, sino de una aleación de este metal con plomo, generalmente con una proporción respectiva del 60% y del 40%, que resulta ser la más indicada para las soldaduras en Electrónica.

Para realizar una buena soldadura, además del soldador y de la aleación descrita, se necesita una sustancia adicional, llamada pasta de soldar, cuya misión es la de facilitar la distribución uniforme del estaño sobre las superficies a unir y evitando, al mismo tiempo, la oxidación producida por la temperatura demasiado elevada del soldador. La composición de esta pasta es a base de colofonia (normalmente llamada “resina”) y que en el caso del estaño que utilizaremos, está contenida dentro de las cavidades del hilo, en una proporción del 2~2.5%.

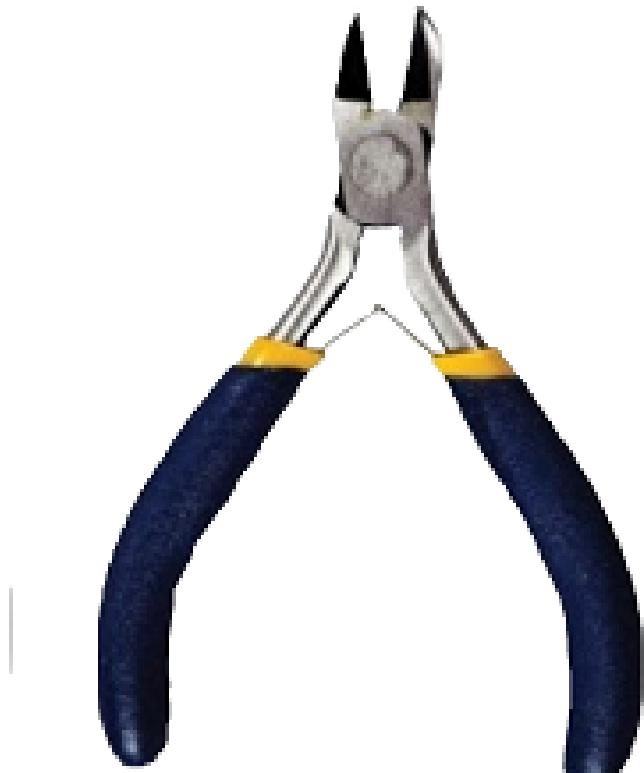


Figure 2.3: alicate para electronica

Un pequeño alicate, para poder cortar el excedente de material (estaño, alambres de las resistencias por ejemplo).



Figure 2.4: destornillador plano pequeño

Nos sirve para ajustar las borneras y para hacer palanca para sacar un integrado que hayamos puesto en un zocalo.



Figure 2.5: desoldador de estaño

El desoldador de estaño, nos permite sacar el estaño que hayamos puesto de mas o para remplazar algun componente efectuoso de la placa robotica *np07*

## 2.4 fabricacion

A continuación veremos el paso a paso del armado de la placa *np07*.

### 2.4.1 paso 0

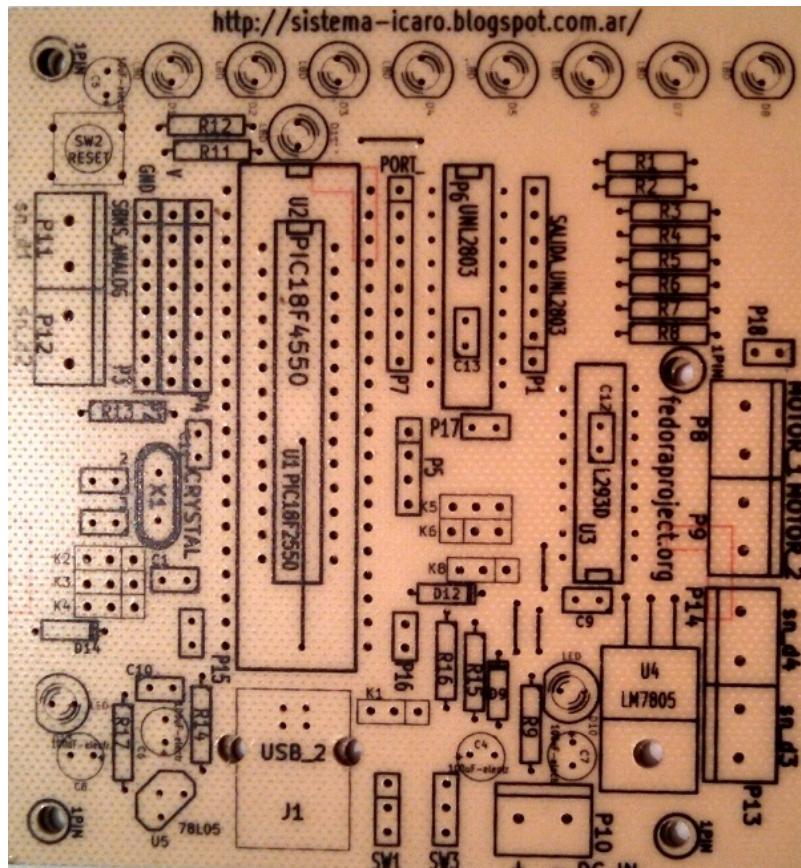


Figure 2.6: Vista de la Placa

## 2.4.2 paso 1

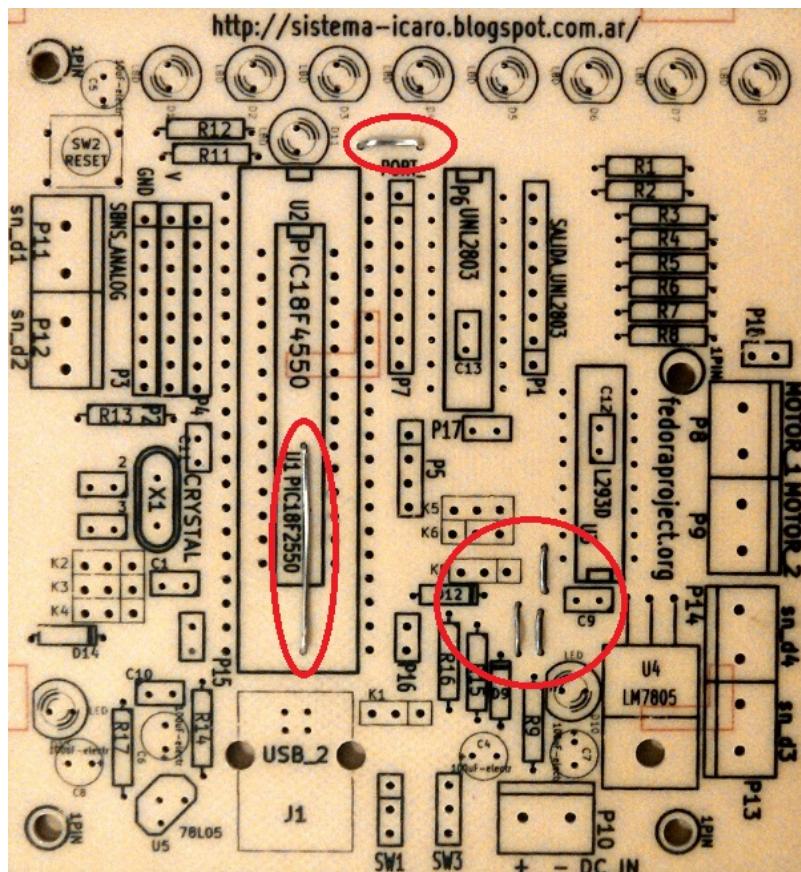


Figure 2.7: Colocar 5 Puentes

### 2.4.3 paso 2

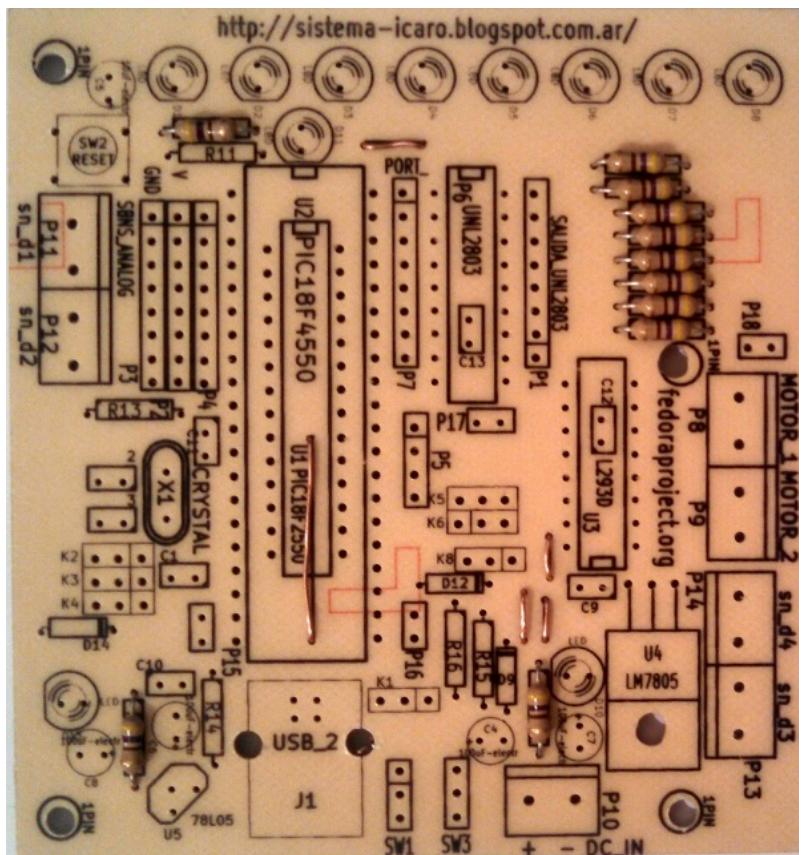


Figure 2.8: Resistencias de 470 Ohm

## 2.4.4 paso 3

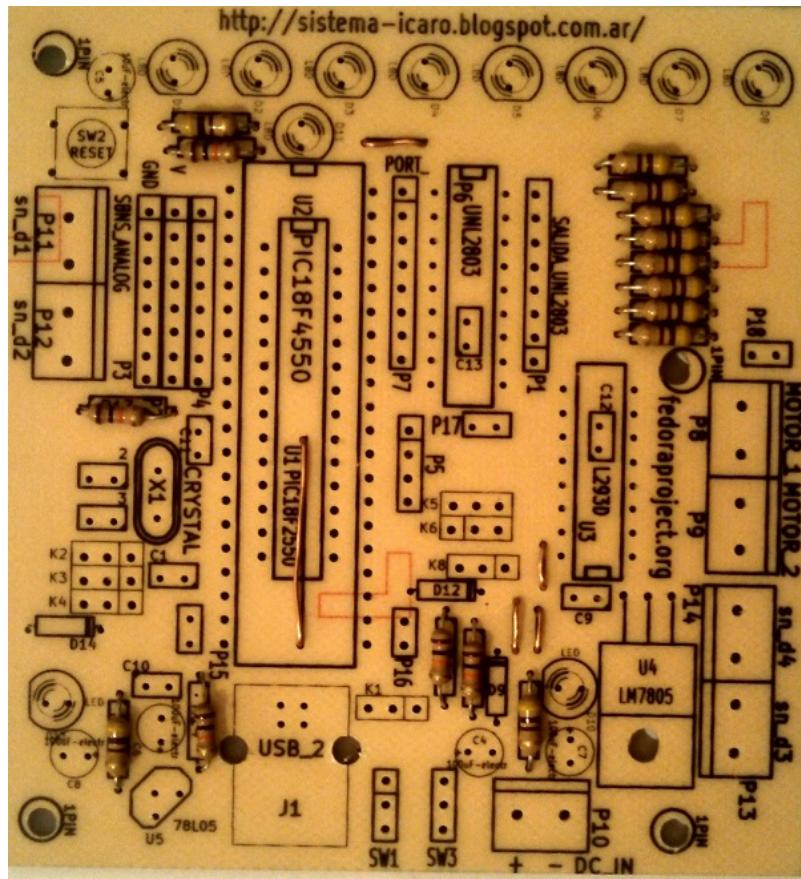


Figure 2.9: Resistencias de 10K Ohm

## 2.4.5 paso 4

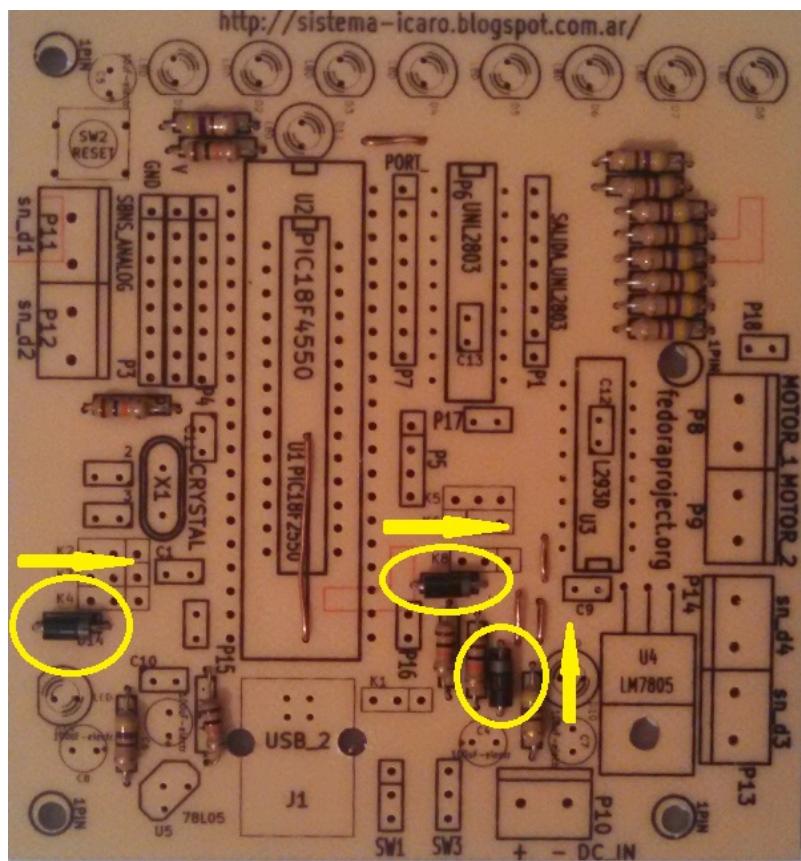


Figure 2.10: Diodos 1N4007

## 2.4.6 paso 5

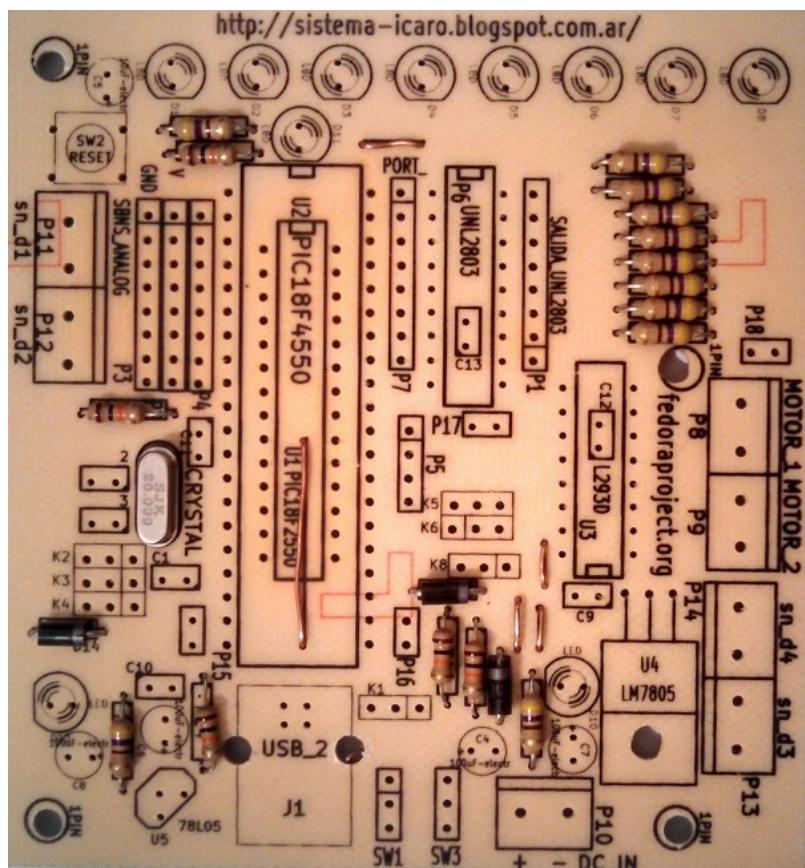


Figure 2.11: Cristal de 20MHz

## 2.4.7 paso 6

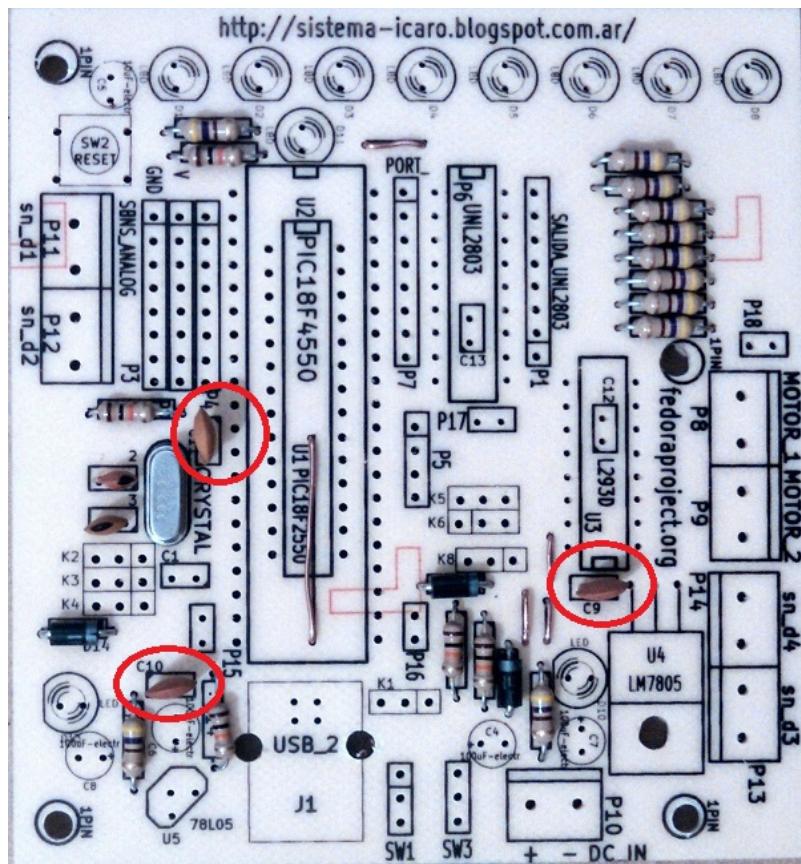


Figure 2.12: Capacitores Cerámicos 0,1uF

## 2.4.8 paso 7

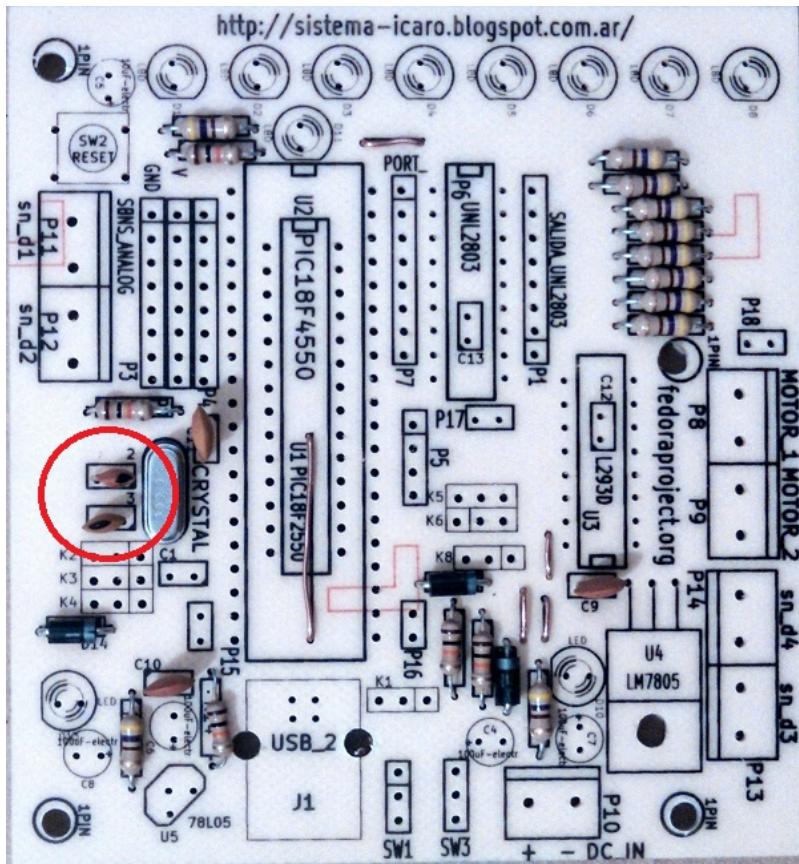


Figure 2.13: Capacitores Cerámicos 22pF

## 2.4.9 paso 8

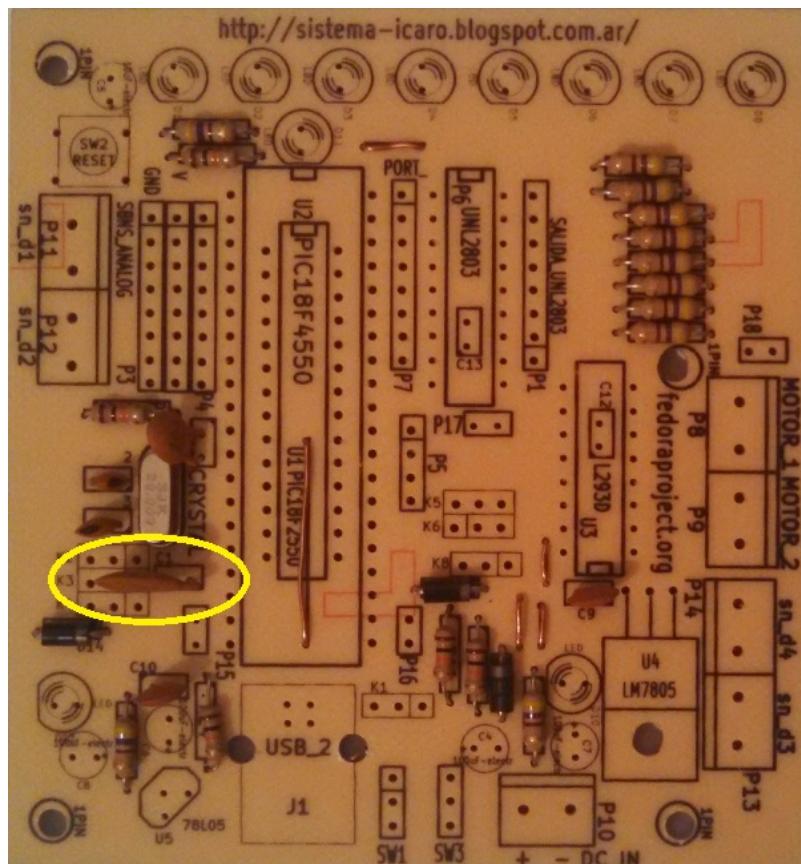


Figure 2.14: Capacitor Cerámico 220nF

## 2.4.10 paso 9

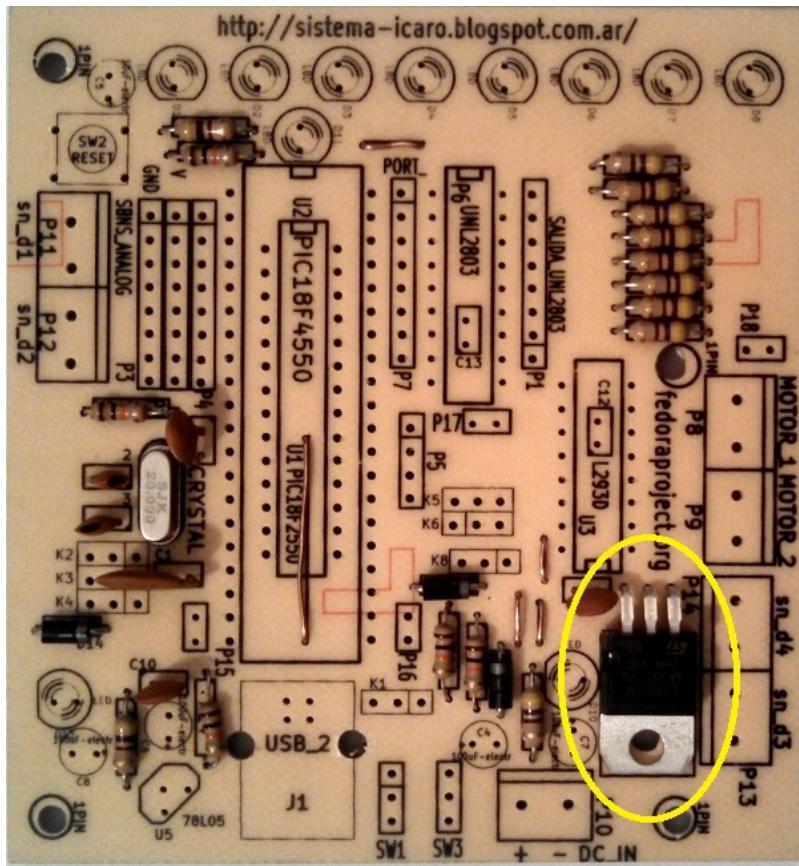


Figure 2.15: Regulador LM7805

#### 2.4.11 paso 10

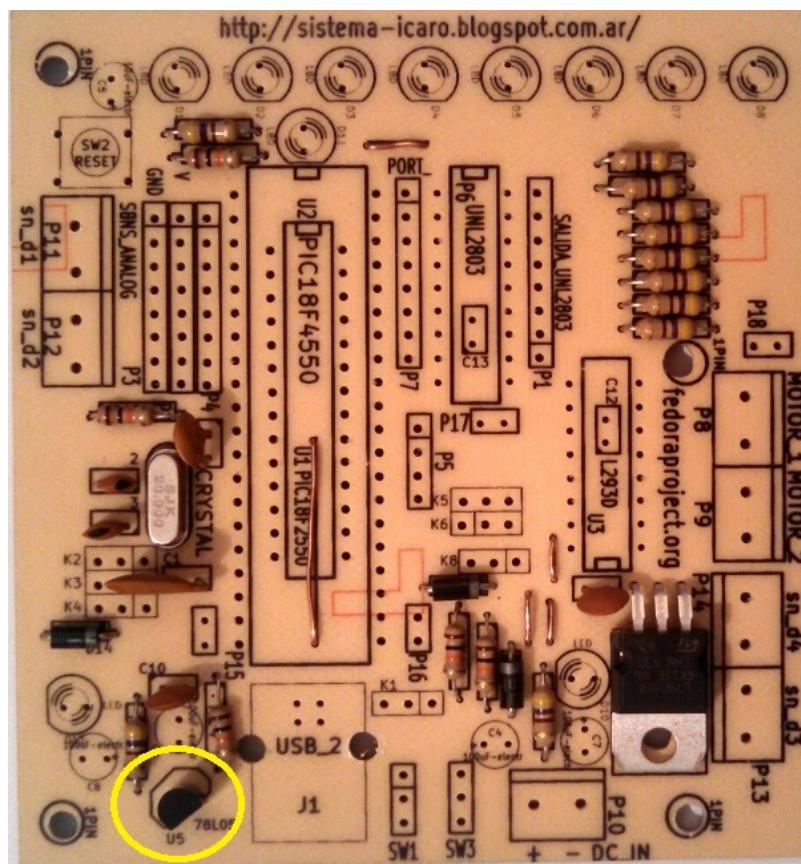


Figure 2.16: Regulador 78L05

#### 2.4.12 paso 11

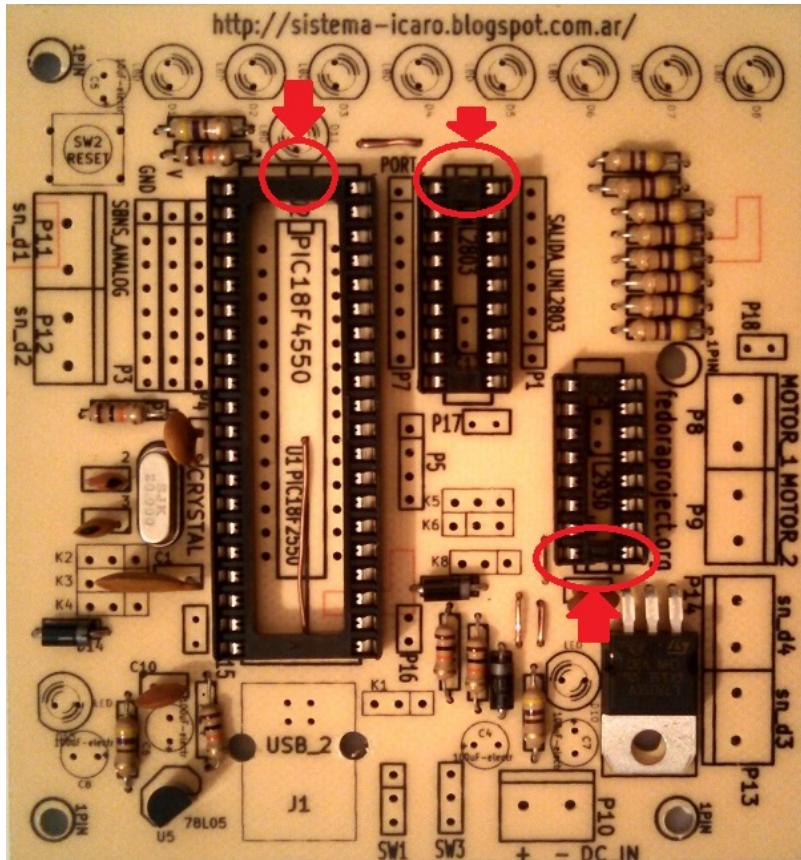


Figure 2.17: Colocar Zócalos

#### 2.4.13 paso 12

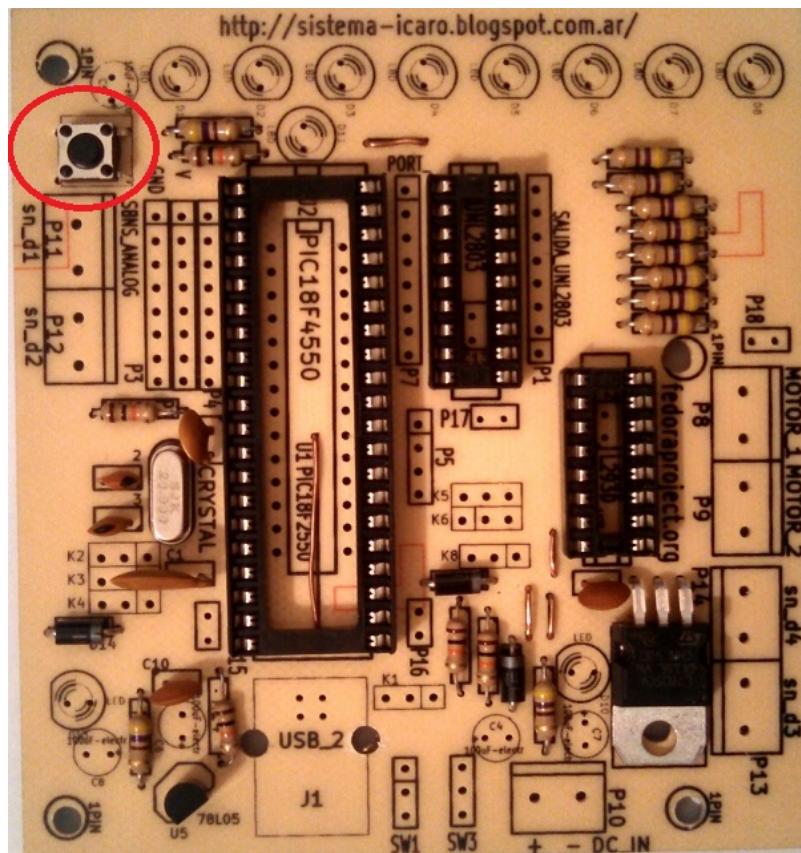


Figure 2.18: Push Button

#### 2.4.14 paso 13

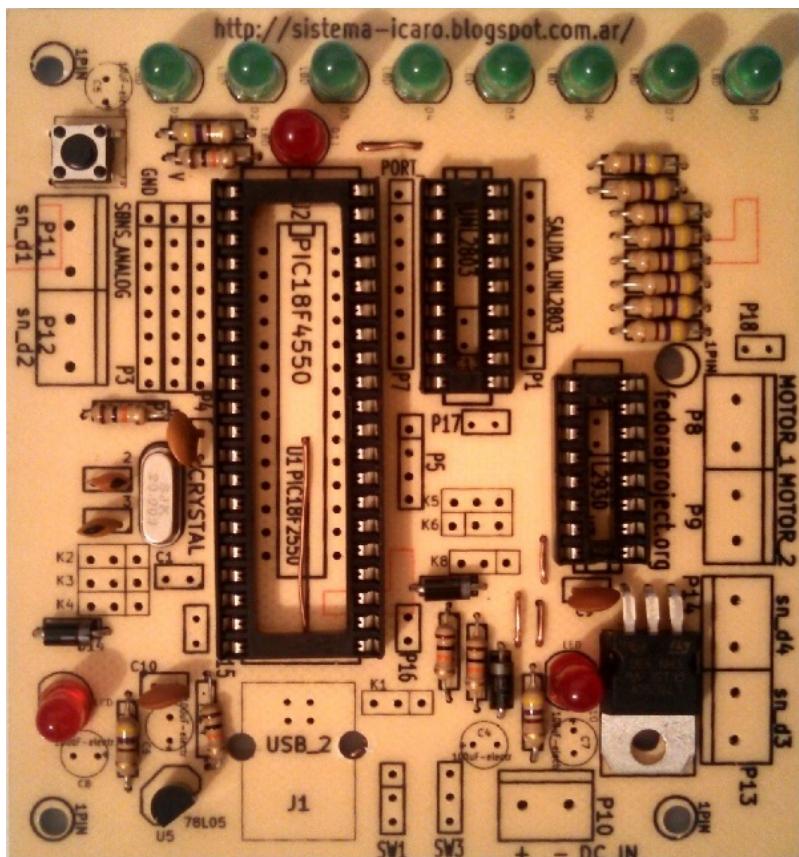


Figure 2.19: Colocar LEDs

#### 2.4.15 paso 14

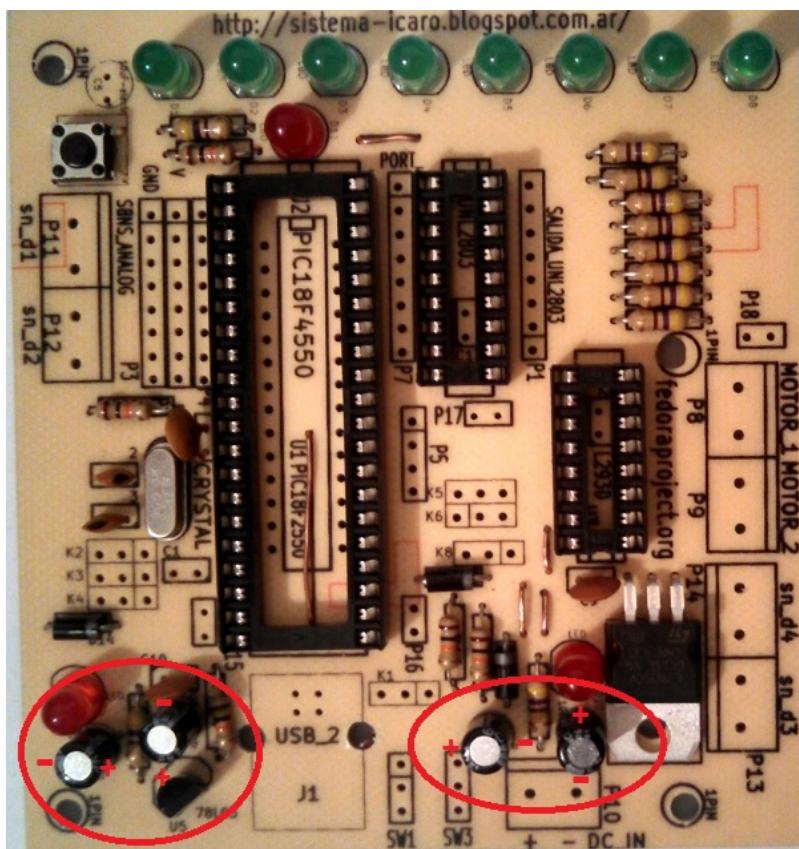


Figure 2.20: Capacitores Electrolíticos 100uF

## 2.4.16 paso 15

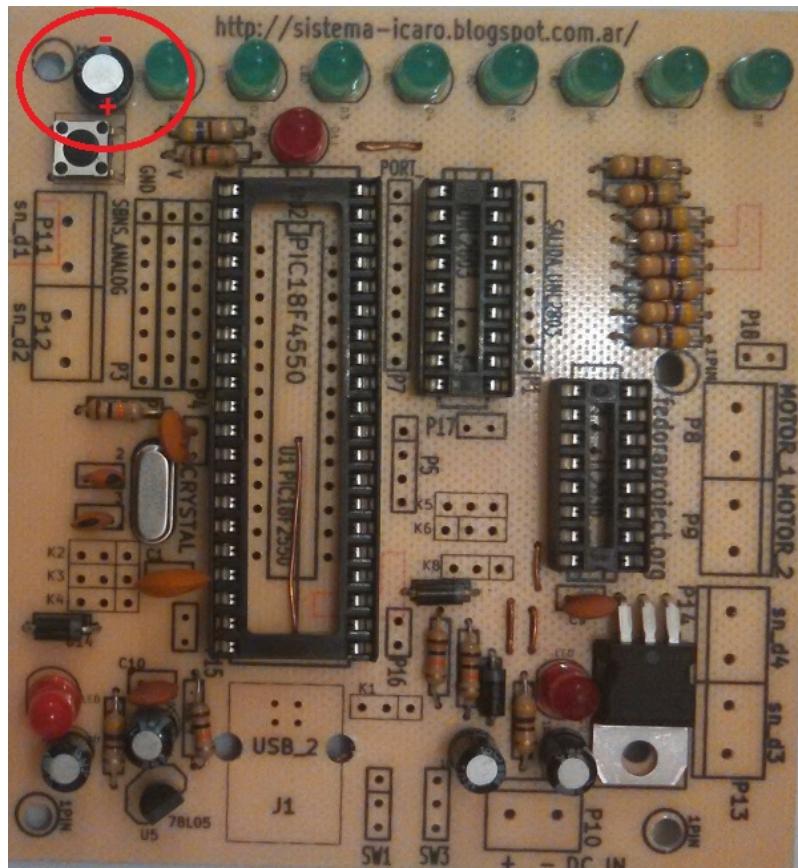


Figure 2.21: Capacitor Electrolítico 10uF

#### 2.4.17 paso 16

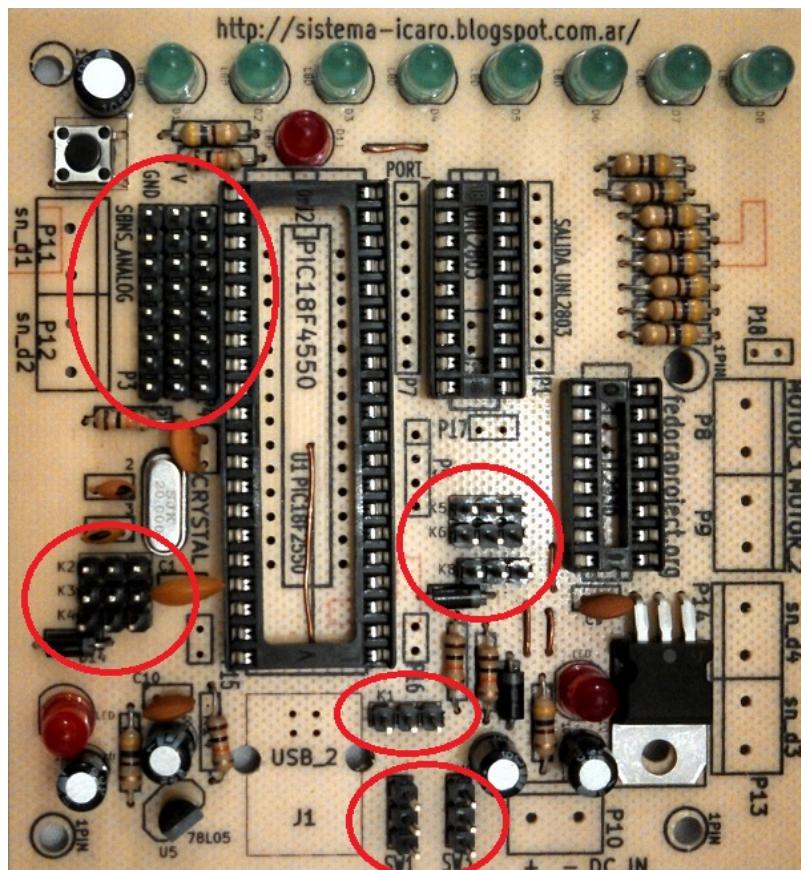


Figure 2.22: Postes Macho

2.4.18 paso 17

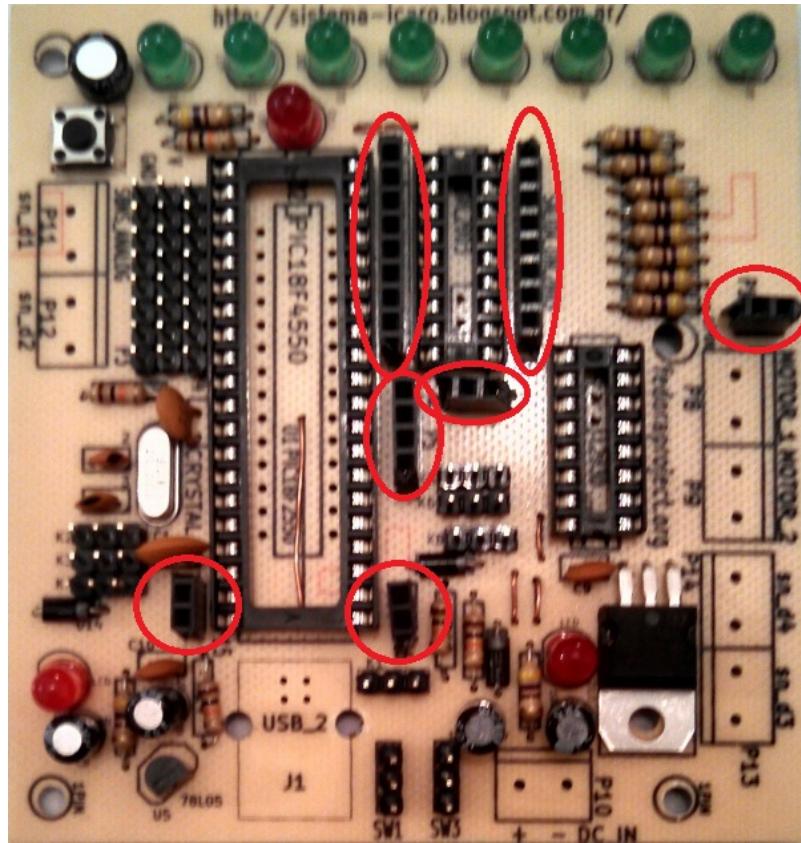


Figure 2.23: Postes Hembra

#### 2.4.19 paso 18

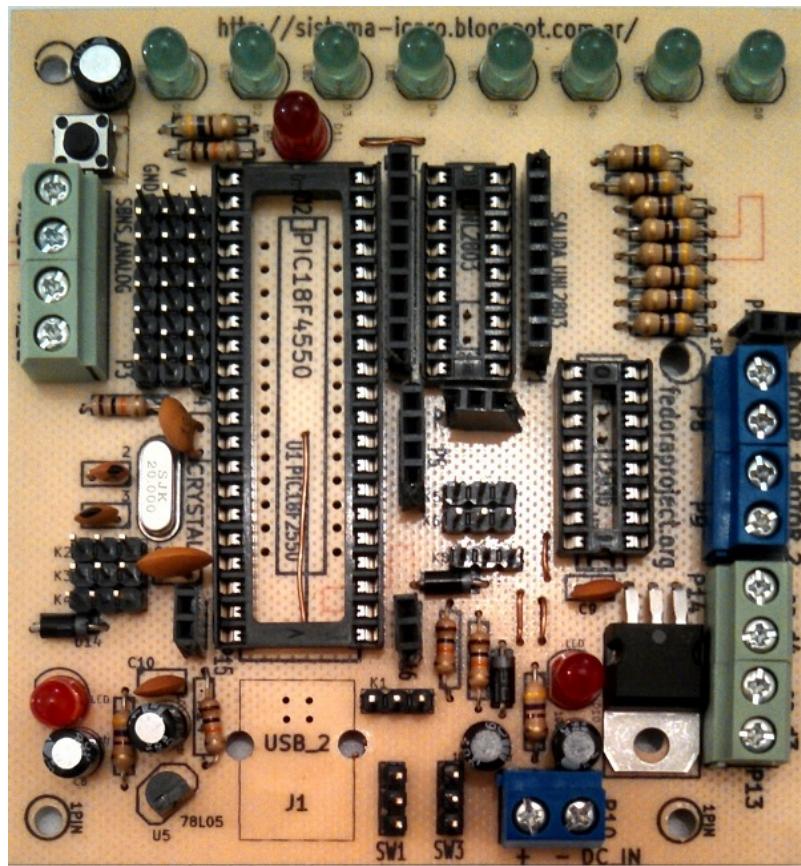


Figure 2.24: Borneras

## 2.4.20 paso 19

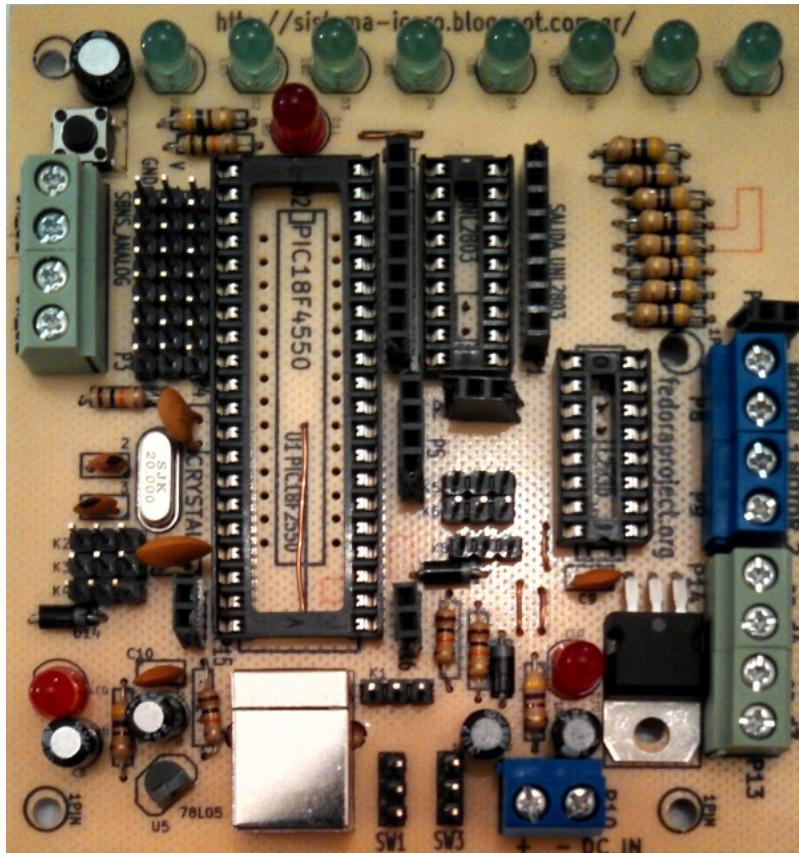


Figure 2.25: Conector USB hembra B

#### 2.4.21 paso 20

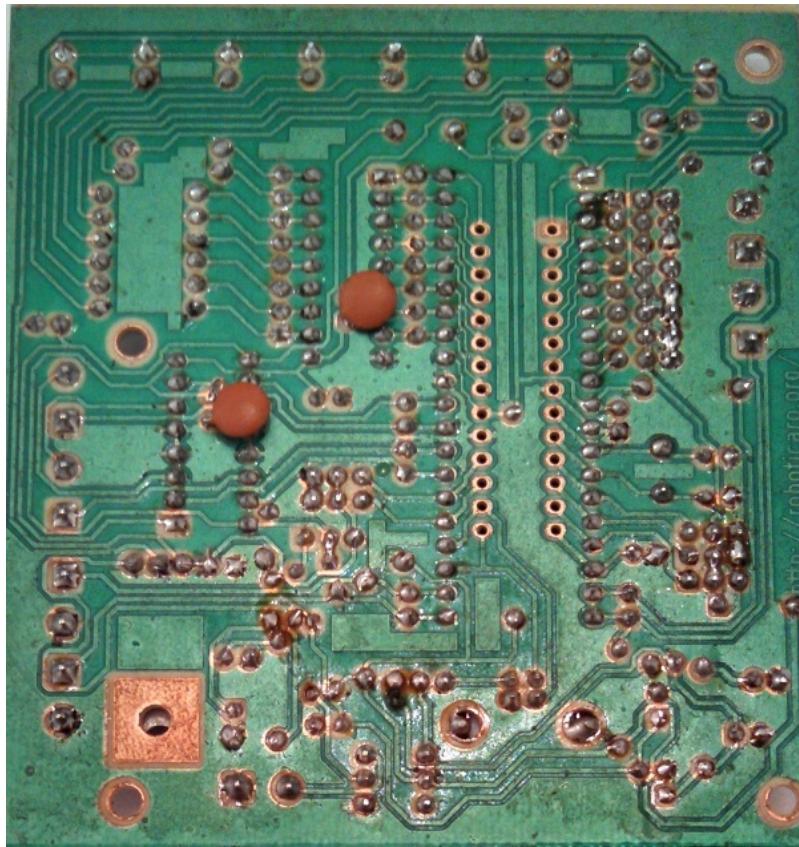


Figure 2.26: Capacitores Cerámicos 0,1uF



## ICARO-BLOQUES

bla

### 3.1 barra de herramientas

bla

### 3.2 barra de componentes

bla

### 3.3 editor de bloques

bla

### 3.4 editor de codigo fuente

bla



## TRABAJANDO CON ICARO-BLOQUES

bla

### 4.1 compilando un proyecto

bla

### 4.2 cargando el firmware a la placa

bla

### 4.3 referencias de los bloques

bla



## GUIA DE EJEMPLOS

bla

### 5.1 usando LEDS

bla

#### 5.1.1 Ejemplo 01

bla

#### 5.1.2 ejemplo 02

bla

#### 5.1.3 Ejemplo 03

bla

#### 5.1.4 ejemplo 04

bla

#### 5.1.5 Ejemplo 05

bla

### 5.2 usando motores CC

bla

#### 5.2.1 hola mundo

bla

## 5.2.2 hola mundo 2

bla

## 5.2.3 Ejemplo 01

bla

## 5.2.4 ejemplo 02

bla

## 5.2.5 Ejemplo 03

bla

# 5.3 usando comunicación CDC

bla

## 5.3.1 hola mundo

bla

## 5.3.2 Ejemplo 01

bla

# 5.4 usando una pantalla LCD

bla

## 5.4.1 Ejemplo 01

bla

## 5.4.2 ejemplo 02

bla

# 5.5 usando un sensor hc-sr04

bla

### 5.5.1 ping

bla

### 5.5.2 ping CDC

bla



---

CHAPTER  
**SIX**

---

**PYTHON**

bla

### **6.1 cargando firmware tortucaro**

bla

### **6.2 apicaro**

bla

### **6.3 referencia apicaro**

bla



---

CHAPTER  
SEVEN

---

**CLEMENTE**

bla

### **7.1 activando servidor clemente**

bla

### **7.2 cargando firmware PILAS**

bla

### **7.3 graficador**

bla