

The logo for Fritzing, featuring the word "fritzing" in a white, lowercase, sans-serif font with a dot over the 'i', centered within a red rectangular background.

fritzing

TP_4

**ESTRUCTURAS DE CONTROL REPETITIVAS (BUCLES).
INSTALACIÓN DEL SIMULADOR FRITZING.**

AUTOR: AHMED CHATATOU BOUGDADI

ÍNDICE

1. ESTRUCTURAS DE CONTROL REPETIVAS (BUCLES) (“do...while”; “for”; “while”	2
1.2. EJEMPLOS MODIFICANDO EL CODIGO CON “do...while”; “while” Y “for”	3
2. INSTALACION DEL PROGRAMA FRITZING.....	6
2.2. FUNCIONALIDADES QUE TIENE EL PRGRAMA FRITZING.....	11

1. ESTRUCTURAS DE CONTROL REPETIVAS (BUCLES) (“do...while”; “for”;
“while” 2

Sintaxis

```
while (condición) {  
    //sentencia a ejecutar  
}
```

```
do {  
    //sentencia a ejecutar  
}
```

```
while (condición)
```

```
for (<inicio>;<condición>;<incremento>) {  
    <sentencia>  
}
```

1.2. EJEMPLOS MODIFICANDO EL CODIGO

While...

Antirrobo_Mejorado_while_Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

```
Antirrobo_Mejorado_while_
8 //Conexiones
9 const int ledSalonPin=8;
10 const int ledCocinaPin=5;
11 const int ledEntradaPin=2;
12 const int ledDormitorioPin=11;
13 const int numeroLed=12;
14
15 //tiempo entre pasos
16 int duracionTarde=6000;
17 int duracionMadrugada=6000;
18 int duracionNoche=6000;
19 int duracionSalonCocina=2000;
20 int duracionDormitorio=1000;
21 int duracionEntrada=6000;
22
23 void setup() {
24   int numeroLed=2;
25   while (numeroLed<12) {pinMode (numeroLed, OUTPUT);
26                       numeroLed=numeroLed+3;}
27
28 }
29
30 void loop() {
31   // Estado Tarde
32   digitalWrite (ledSalonPin,HIGH); //encendemos salon.
33   digitalWrite (ledCocinaPin,HIGH); //encendemos cocina.
34   delay (duracionSalonCocina); //esperamos duracion SalonCocina
35   digitalWrite (ledSalonPin,LOW); //apagamos cocina
```

Compilado

```
Archiving built core (caching) in: C:\Users\JUANDA~1\AppData\Local\Temp
El Sketch usa 1156 bytes (3%) del espacio de almacenamiento de programa
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejand
```

do.....while

Antirrobo_Mejorado_for_Arduino 1.8.5

Archivo Editar Programa Herramientas Ayuda

```
Antirrobo_Mejorado_for_ $
24  /*pinMode (ledSalonPin, OUTPUT);
25  pinMode (ledCocinaPin, OUTPUT);
26  pinMode (ledEntradaPin, OUTPUT);
27  pinMode (ledDormitorioPin, OUTPUT); */
28
29  /*for (int numeroLed=2;numeroLed<12;numeroLed=numeroLed+3)
30
31      {pinMode (numeroLed, OUTPUT);}*/
32
33  do
34      {
35          pinMode (numeroLed, OUTPUT);
36          numeroLed=numeroLed+3;
37      }
38      while (numeroLed<12);
39
40 }
41
42 void loop() {
43     // Estado Tarde
44     digitalWrite (ledSalonPin, HIGH); //encendemos salon.
45     digitalWrite (ledCocinaPin, HIGH); //encendemos cocina.
46     delay (duracionSalonCocina); //esperamos duracion SalonCocina
47     digitalWrite (ledSalonPin, LOW); //apagamos cocina
48     delay (duracionTarde); //esperamos duracionNoche. hasta que
49
50     //Estado Noche
51     digitalWrite (ledEntradaPin, HIGH); //encendemos entrada
```

Compilado

El Sketch usa 1156 bytes (3%) del espacio de almacenamiento de programa. El espacio de memoria dinámica disponible es de 1024 bytes. Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 1015 bytes disponibles para las variables locales. El espacio de memoria dinámica disponible es de 1024 bytes.

42

for

Antirrobo_Mejorado_for_Arduino 1.8.5

Archivo Editar Programa Herramientas Ayuda

```
Antirrobo_Mejorado_for_
4  debe activarse la alarma de cada sector de la vivienda.
5  */
6
7  //Declaracion de variable
8  //Conexiones
9  const int ledSalonPin=8;
10 const int ledCocinaPin=5;
11 const int ledEntradaPin=2;
12 const int ledDormitorioPin=11;
13
14 //tiempo entre pasos
15 int duracionTarde=6000;
16 int duracionMadrugada=6000;
17 int duracionNoche=6000;
18 int duracionSalonCocina=2000;
19 int duracionDormitorio=1000;
20 int duracionEntrada=6000;
21
22 void setup() {
23   /*pinMode (ledSalonPin, OUTPUT);
24   pinMode (ledCocinaPin, OUTPUT);
25   pinMode (ledEntradaPin, OUTPUT);
26   pinMode (ledDormitorioPin, OUTPUT); */
27
28   for(int numeroLed=2;numeroLed<12;numeroLed=numeroLed+3)
29
30     {pinMode (numeroLed, OUTPUT) ;}
31
```

Compilado

```
Archiving built core (caching) in: C:\Users\JUANDA~1\AppData\Loc
El Sketch usa 1156 bytes (3%) del espacio de almacenamiento de p
Las variables Globales usan 9 bytes (0%) de la memoria dinámica,
```

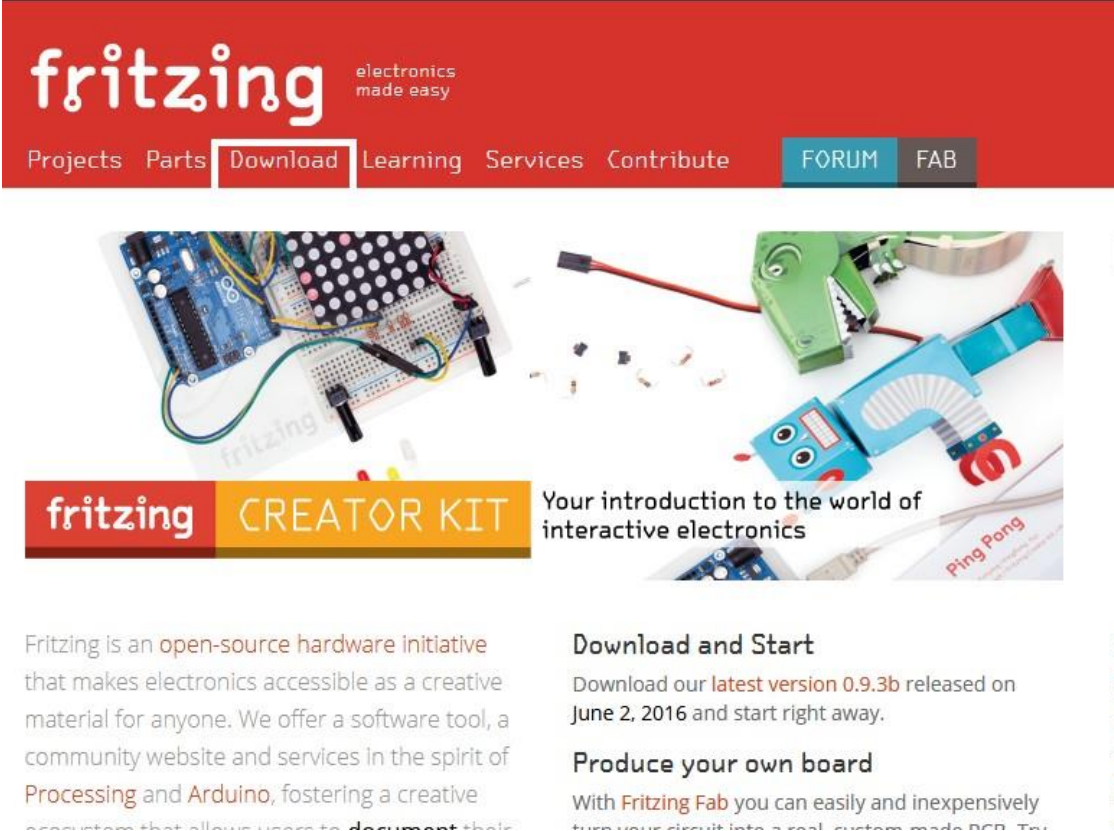
28

2. INSTALACION DE FRITZING.

Para descargar el fritzing, nos dirigimos a su página oficial.

<http://fritzing.org/home/>.

Una vez dentro de la página procedemos a descargarlo en la pestaña “DOWNLOAD” y pinchamos en Download.



fritzing electronics made easy

Projects Parts **Download** Learning Services Contribute FORUM FAB

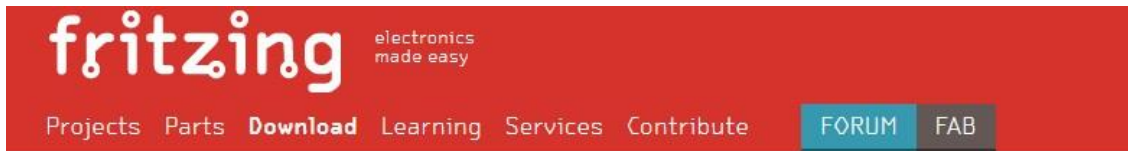
fritzing CREATOR KIT

Your introduction to the world of interactive electronics

Download and Start
Download our **latest version 0.9.3b** released on June 2, 2016 and start right away.

Produce your own board
With **Fritzing Fab** you can easily and inexpensively turn your circuit into a real, custom made PCB. Try

Después nos saldrá una nueva ventana donde tiene la opción de descargarlo para diferentes sistemas operativos, seleccionamos en Windows y empezamos la descarga.



Fritzing is open source, free software. Be aware that the development of it depends on the active support of the community. Select the download for your platform below.

Version **0.9.3b** was released on **June 2, 2016**.

[Windows](#) 32 bit

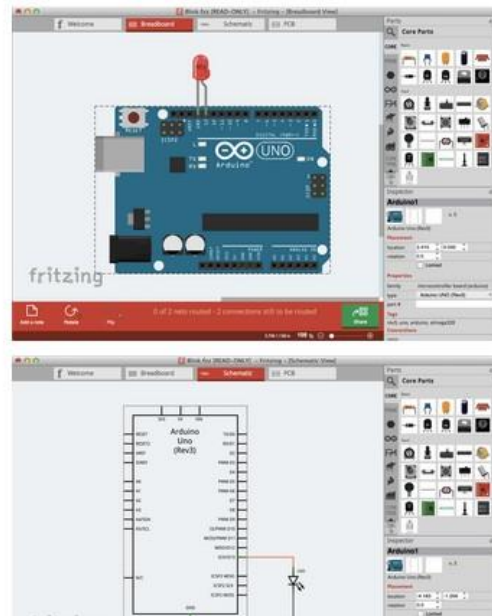
[Windows](#) 64 bit

[Mac OS X](#) 10.7 and up

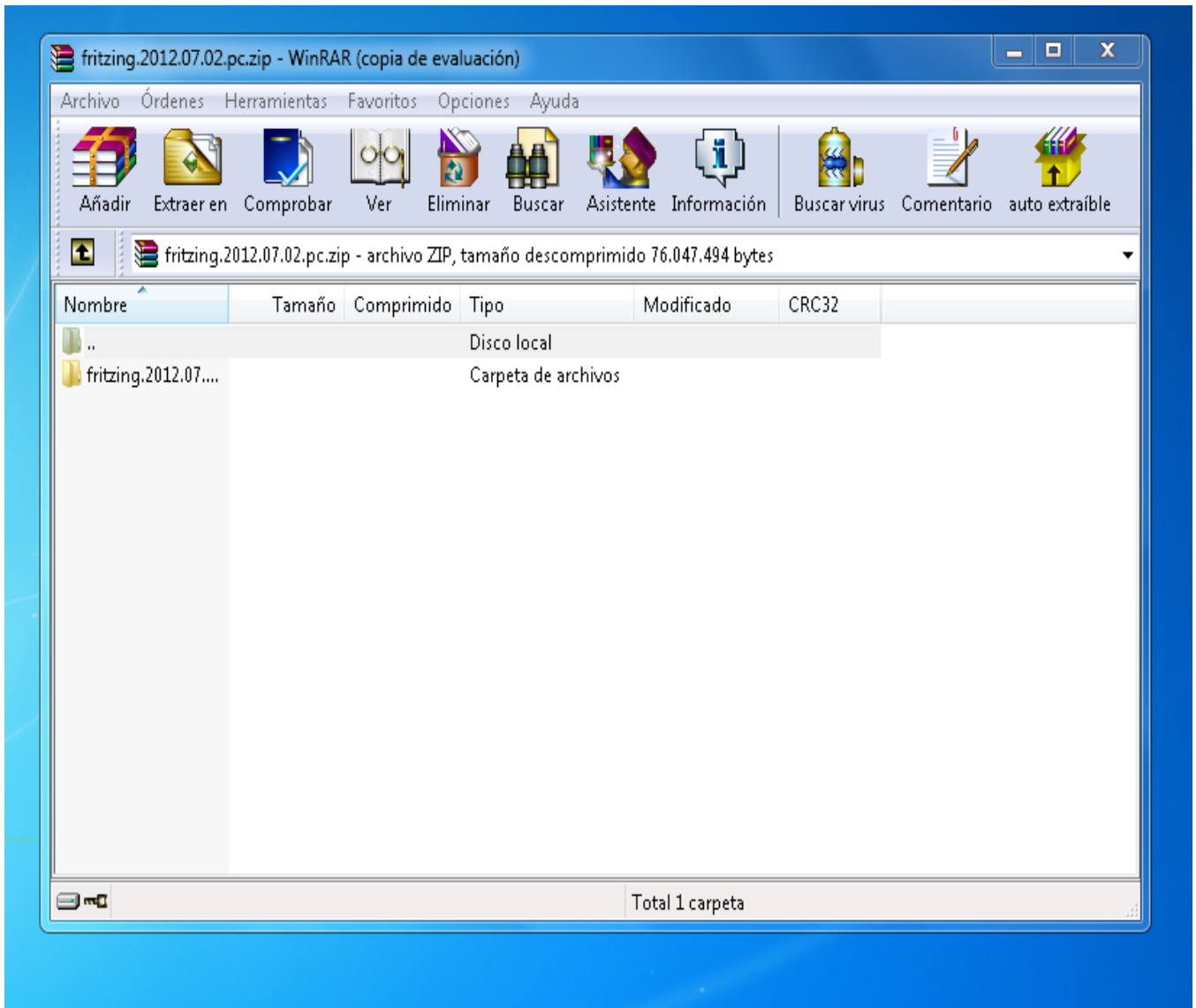
[Linux](#) 32 bit

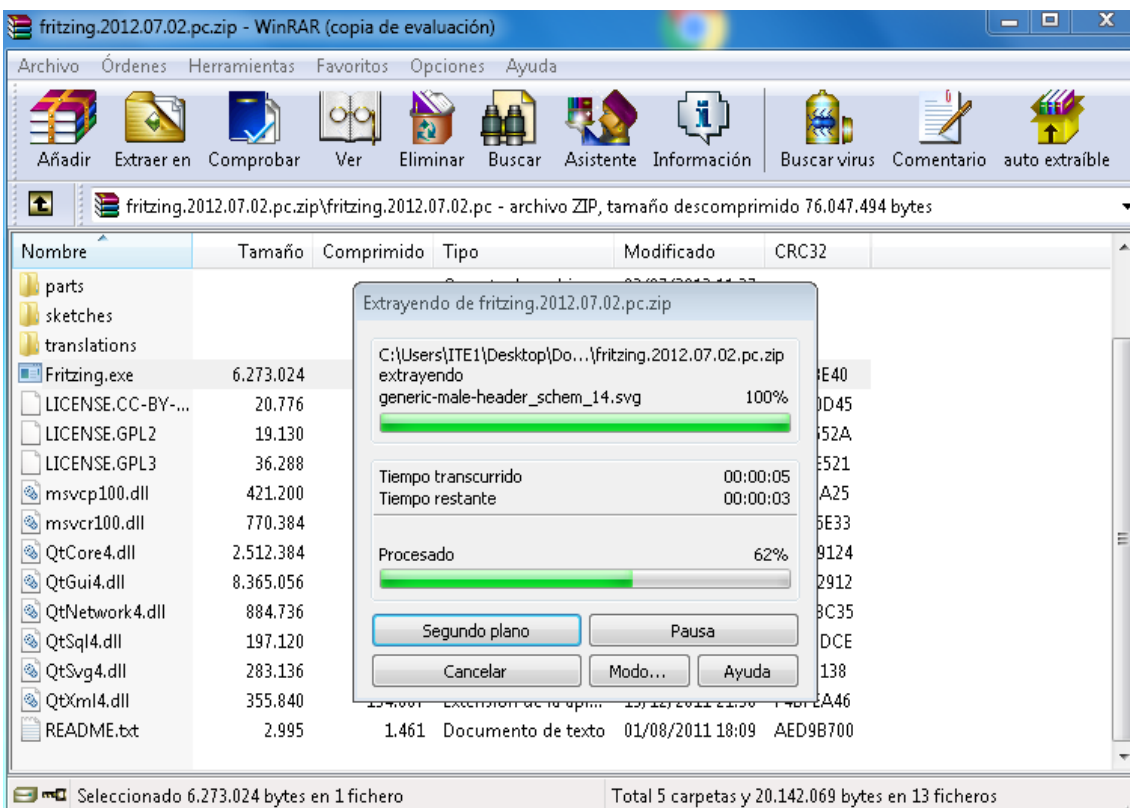
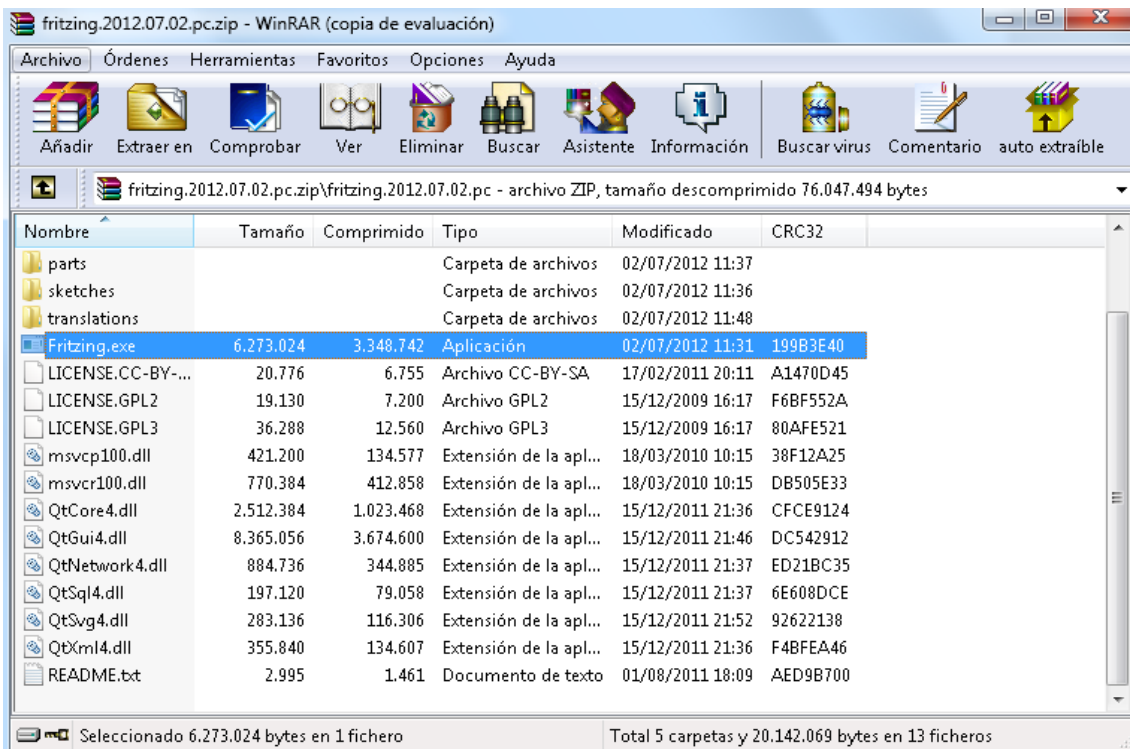
[Linux](#) 64 bit

[Source](#) Github



Una vez descargado, vemos que el programa viene comprimido en un formato WinRAR el cual tenemos que descomprimir y empezara el proceso, y nos quedara luego la carpeta del programa con el ejecutable.



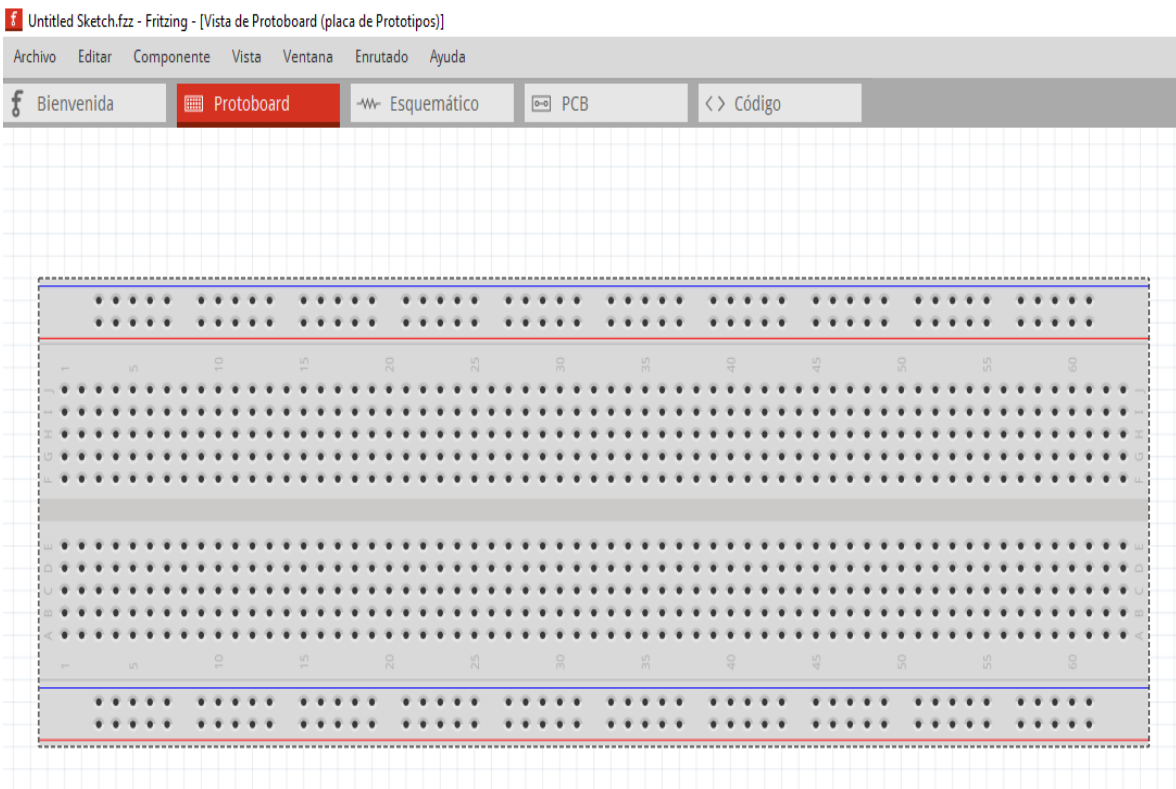


Una vez terminado el proceso, clicamos en el ejecutable para iniciar el programa y ya podemos empezar a trabajar.

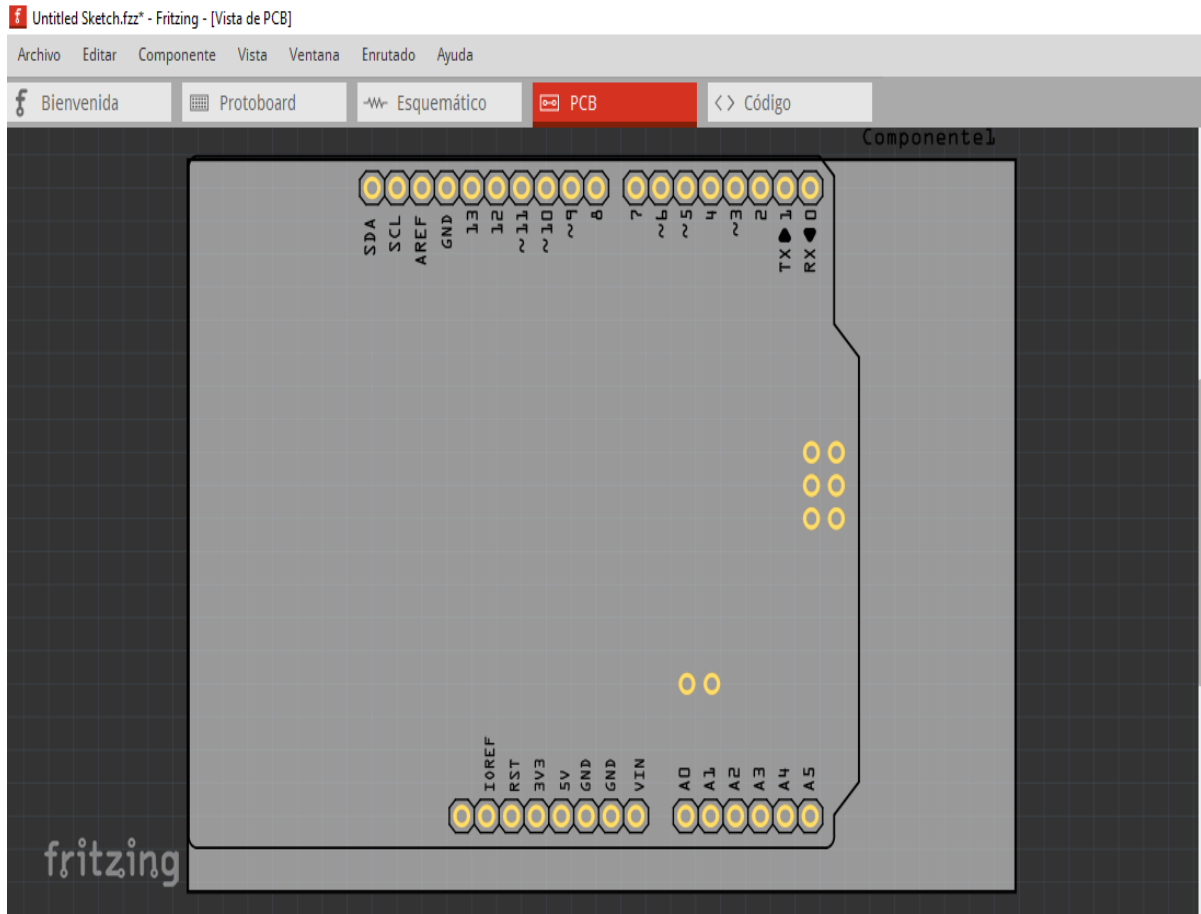
2.2. FUNCIONALIDADES DE FRITZING.

Como podemos ver el programa nos ofrece el poder montar circuitos con muchos componentes y nos vendrá muy bien para usarlo con ARDUINO, ya que cuenta con un editor de código que es muy práctico a la hora de crear programas.

Simulador Protoboard.



PCB



Editor de código de Fritzing

