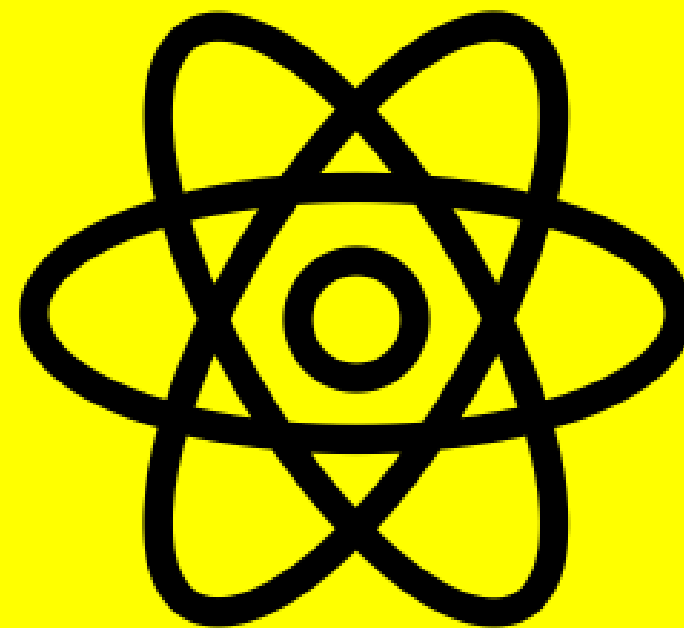
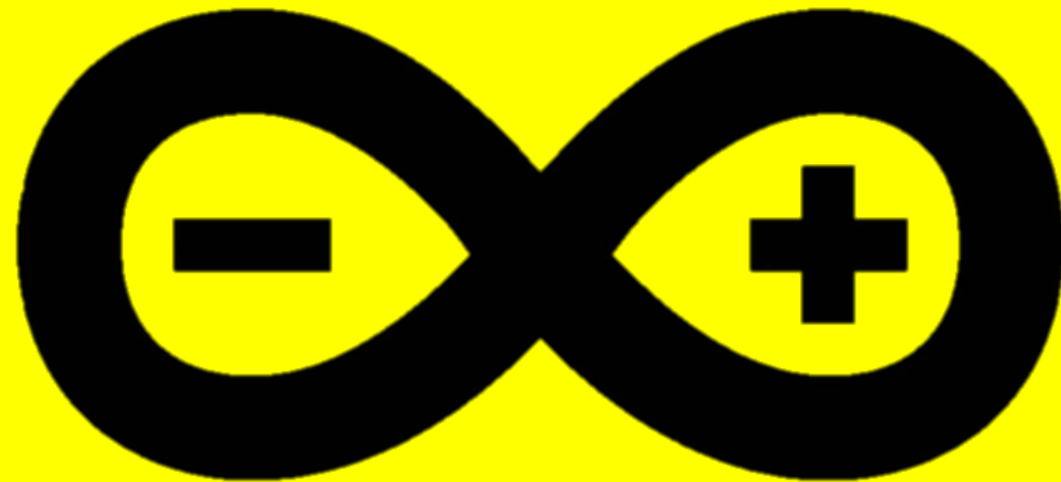




React Native en Aplicativos de Arduino

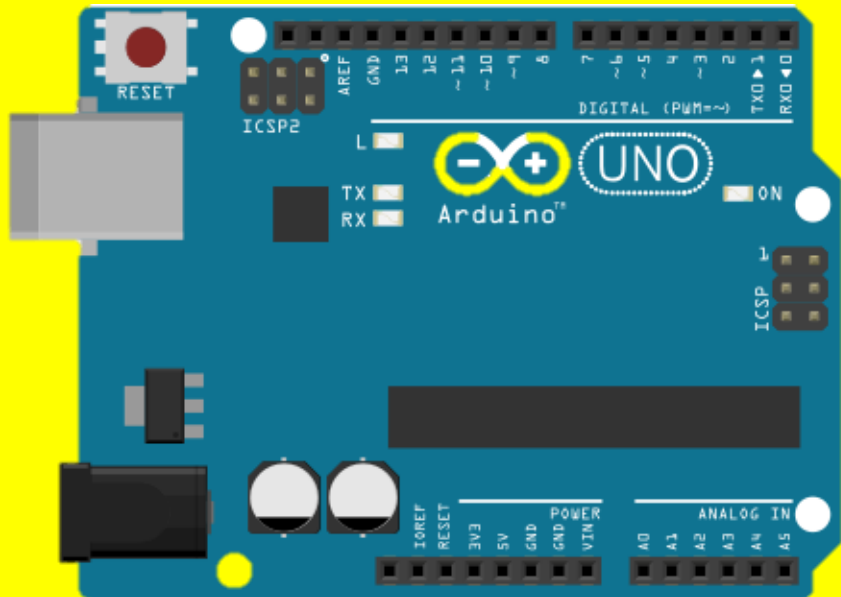


Índice

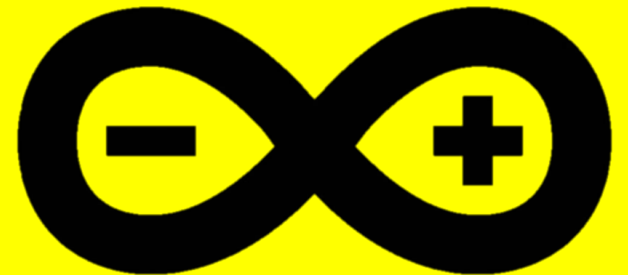
1. Que es Aruino
- 2.Estructura.
- 3.Setup()
- 4.Loop().
- 5.Variables ;
- 6.Proyecto
- 7.Materiales
- 8.Diagraman
- 9.Codio

1.-Que es Arduino

- Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra, los que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.



La “inteligencia de Arduino” se expresa mediante su lenguaje de programación



2.-Estructura

- La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup()  
{  
  setencias;  
}  
void loop()  
{  
  setencias;  
}
```

Setup()

- La función setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar.

```
void setup()
{
  pinMode(pin, OUTPUT); // configura el 'pin' como salida
}
```

Loop()

- Después de llamar a setup(), la función loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la tarjeta.

```
void loop()
{
    digitalWrite(pin, HIGH); // pone en uno (on, 5v) el 'pin'
    delay(1000); // espera un segundo (1000 ms)
    digitalWrite(pin, LOW); // pone en cero (off, 0v.) el
    'pin'
    delay(1000);
}
```

Nota: Olvidarse de poner fin a una línea con un punto y coma se traducirá en un error de compilación. Si se produce un error raro y de difícil detección lo primero que debemos hacer es comprobar que los puntos y comas están colocados al final de las instrucciones.

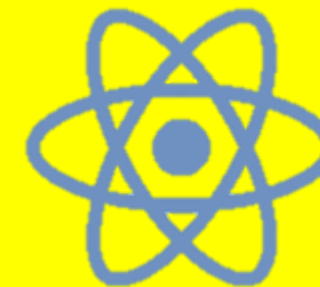
Variables

- Una variable es una manera de nombrar y almacenar un valor numérico para su uso posterior por el programa. Como su nombre indica, las variables son números que se pueden variar continuamente en contra de lo que ocurre con las constantes cuyo valor nunca cambia.

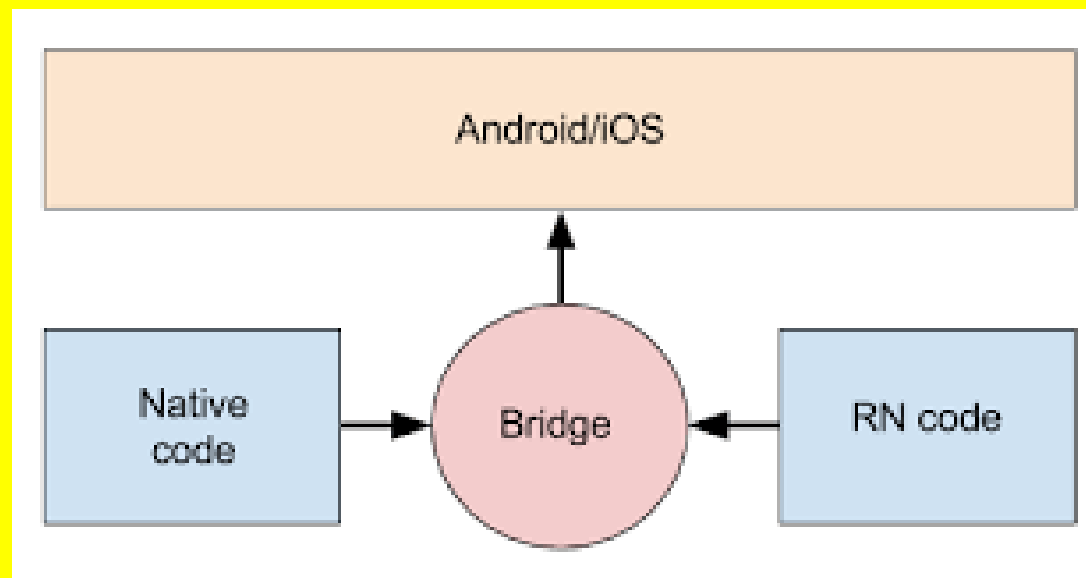
```
int variableEntrada = 0; // declara una variable y le asigna el  
valor 0
```

```
variableEntrada =analogRead(2);// la variable recoge el valor  
analógico del PIN2
```

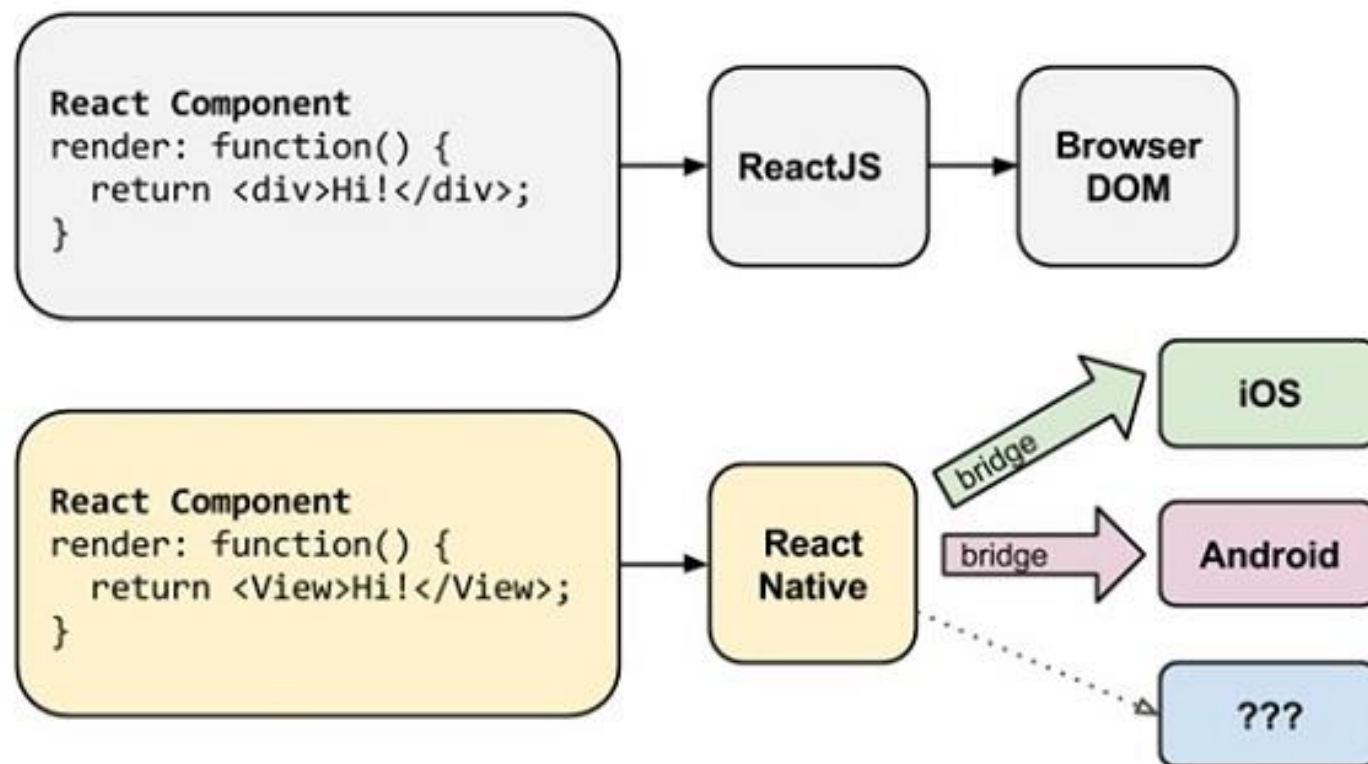
React Native



- Plataforma de código abierto para el desarrollo nativo de aplicaciones móviles desarrollada por Facebook.
- React Native trabaja como puente entre iOS y Android, lo que hace que tu aplicación sea completamente nativa.



Principio básico





Puntos clave

- React Native es una librería de JavaScript, que permite crear aplicaciones iOS y Android.
- Aprenda una vez, escriba en cualquier lugar (no escriba una vez, corra en cualquier lugar)
- FlexBox layout

React Native



Tiene

Componentes

PROYECTO

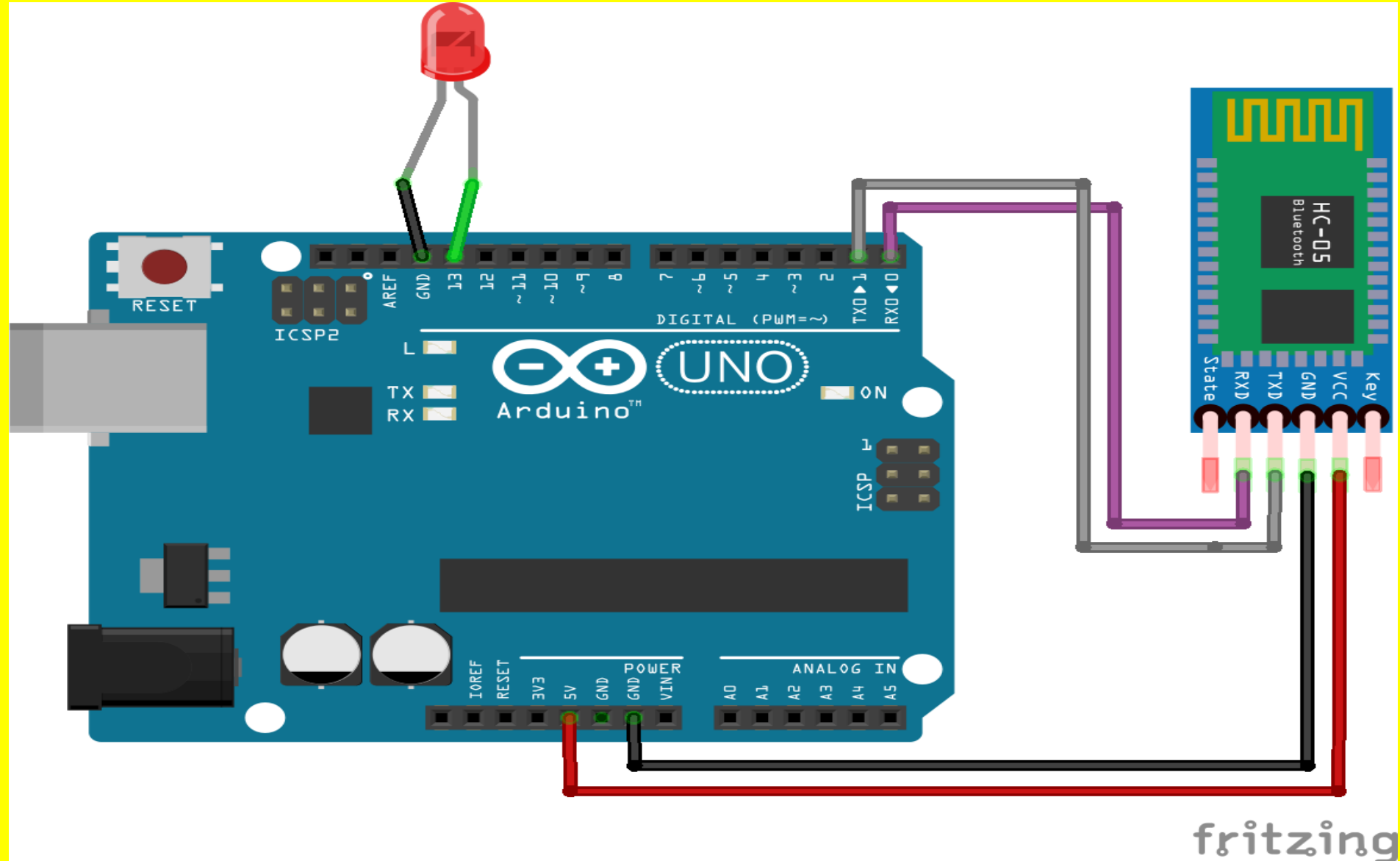
- El objetivo de este proyecto es generar una aplicación que sea capaz de comunicarse con un modulo de bluetooth que estará conectado a un Arduino y a través de la app mandar una alerta al módulo de bluetooth preniendo los led de alerta.

MATERIALES

Los materiales necesarios para la construcción del siguiente proyecto son los siguientes:

- 1 Arduino Genuino Uno
- Jumpers macho-macho
- Jumpers hembra-macho
- 2 leds los cuales serán como interpretaremos las alertas.
- 1 módulo bluetooth hc-05 el cual será nuestro medio de comunicación entre nuestra app y el Arduino.
- Nuestro teléfono con la app instalada

DIAGRAM



Código Arduino

```
#include <SoftwareSerial.h>
SoftwareSerial BT(1,0);

void setup() {
  pinMode(13, OUTPUT);
  BT.begin(9600);
}
char led;
void loop() {
  led=(BT.read());
  switch(led){
    case 'a':
      digitalWrite(13,HIGH);
      delay(2000);
      digitalWrite(13,LOW);
      break;
    case 'b':
      digitalWrite(12,HIGH);
      delay(2000);
      digitalWrite(12,LOW);
      break;
    default:
      digitalWrite(11,HIGH);
      delay(2000);
      digitalWrite(11,LOW);
      break;
  }
}
```