



AUTOMAÇÃO RESIDENCIAL



OFICINA
MAKER



APRESENTAÇÃO INICIAL

• OBJETIVO DO WORKSHOP

- Conhecer um projeto de automação residencial tipo “Faça você mesmo”
- Entender a arquitetura de uma casa automatizada
- Integrar as tecnologias Arduino – Bluetooth – Android
- Ser capaz de automatizar a própria casa

• PÚBLICO ALVO

- Makers que queiram automatizar a própria casa
- Estudantes e profissionais que desejam entrar no mercado de automação residencial
- Estudantes e profissionais que desejam entender a integração Arduino – Bluetooth - Android



FLUXOGRAMA DO WORKSHOP



O MERCADO – BRASIL / PORTO VELHO

- Nichos de atuação da Automação Residencial
 - Áudio e Vídeo
 - Segurança (alarme, monitoramento e controle de acesso)
 - Controle de iluminação
 - Telefonia
 - Rede de dados
 - Ar condicionado e aquecimento
 - Persianas e cortinas
 - Eletrodoméstico
 - Utilidades (irrigação, bomba, aspiração e gás)
 - Gerenciamento de energia

Média de
crescimento anual
nos últimos 10 anos

11%

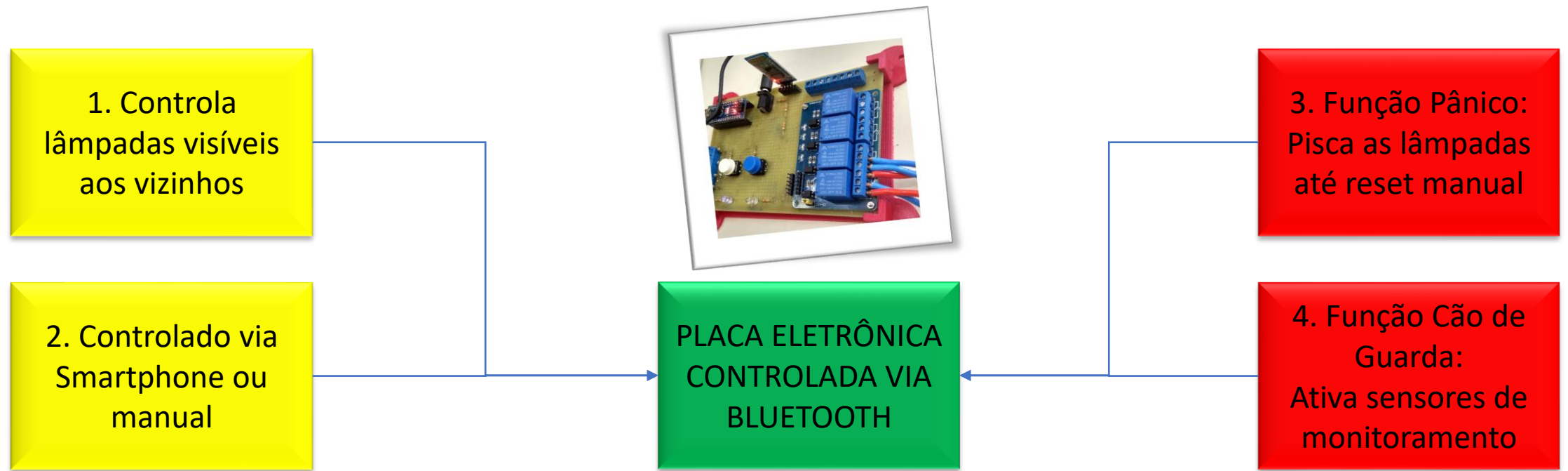
Residências aptas a
receber automação

- 1,5 MILHÃO Brasil
- 40 MIL Porto Velho

Qte de empresas de
Automação
Integrada:
1 em PVH

PROJETO VIZINHO POLÍCIA

- É um projeto vertical (hardware e software) de **código aberto** para o controle de **iluminação estratégica** de uma residência.



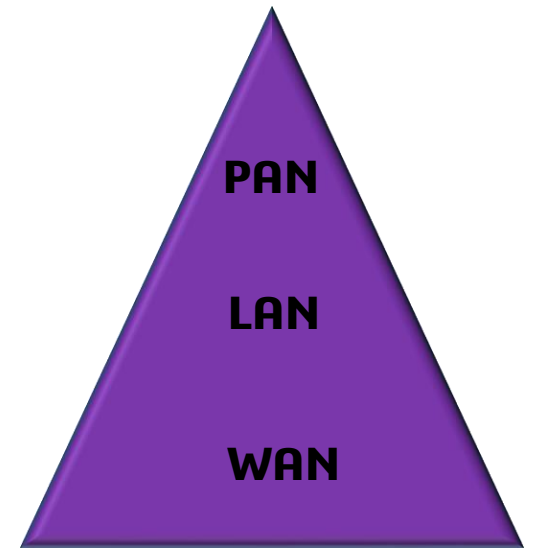
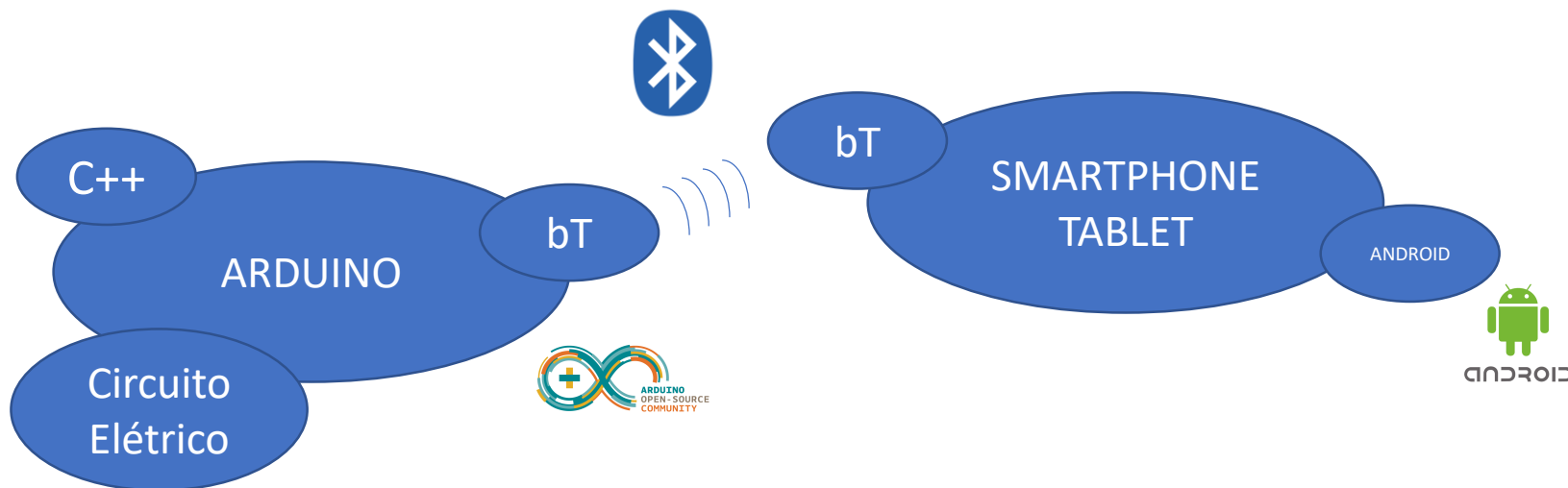
CÓDIGO ABERTO DISPONIBILIZADO NO GITHUB

github.com/oficinamaker/automacaoresidencial/vizinhopolicia

PROJETO – ABORDAGEM TOP DOWN

- ARQUITETURA DE REDE TIPO INTERNET DAS COISAS (IOT)

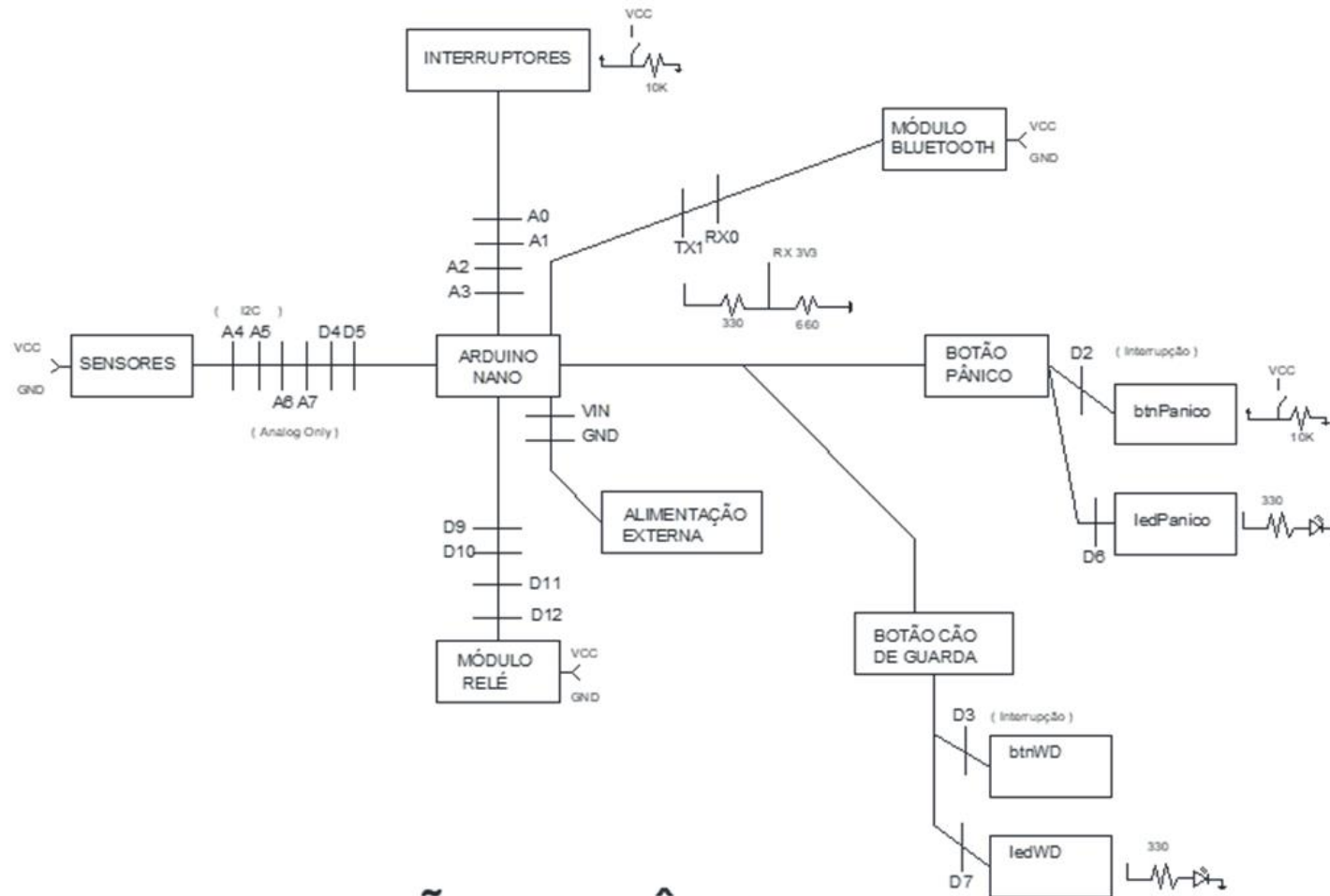
- Camada de aplicação -> Android
- Camada de rede - > Bluetooth (PAN)
- Camada de dispositivo -> Arduino



Projeto Eletrônico

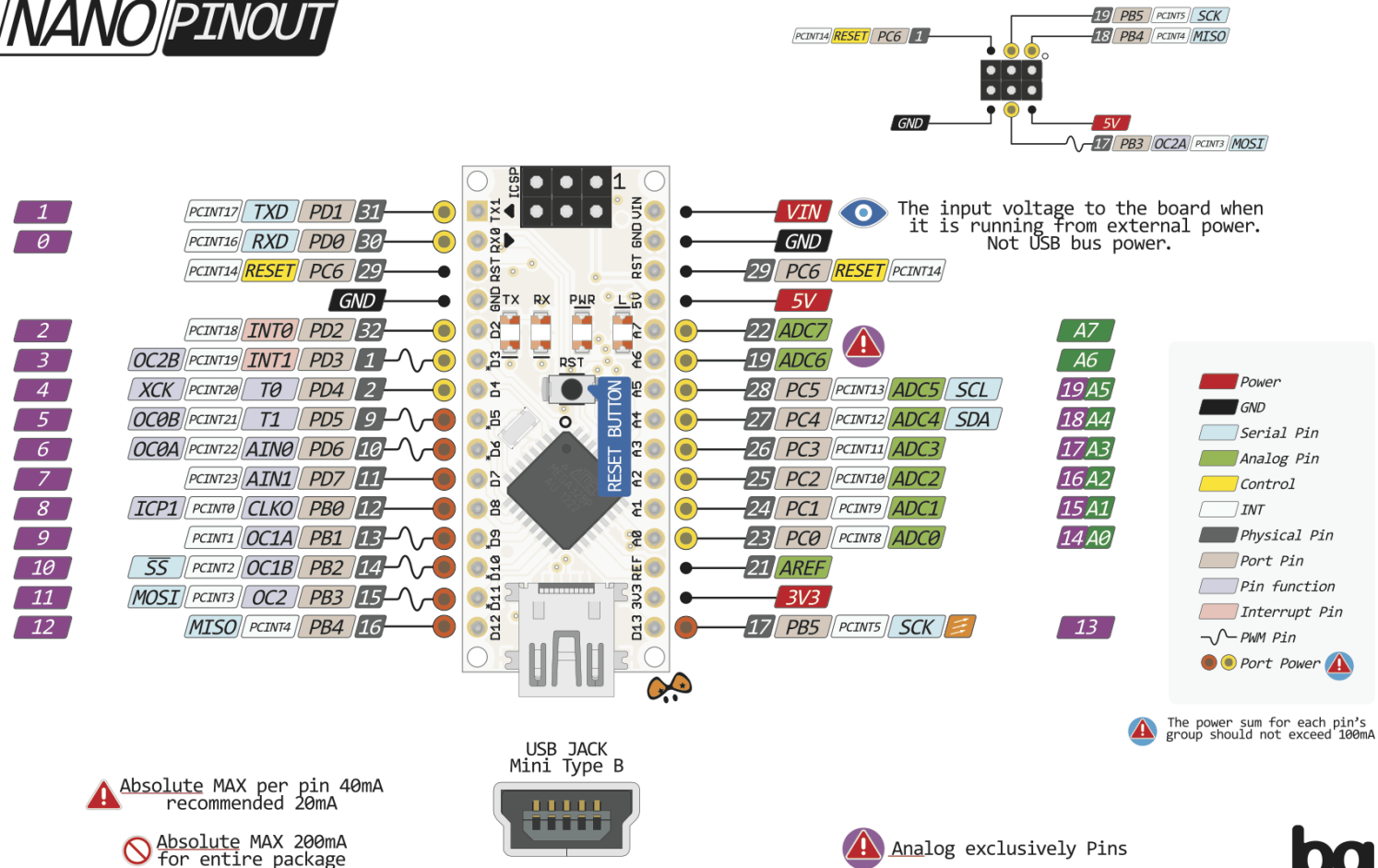
- Divisor de tensão
- Resistência do led
- Resistor pull down
- Alimentação dos módulos

DIAGRAMA - VISÃO SISTÊMICA



PROJETO ELETRÔNICO - CONTROLADOR

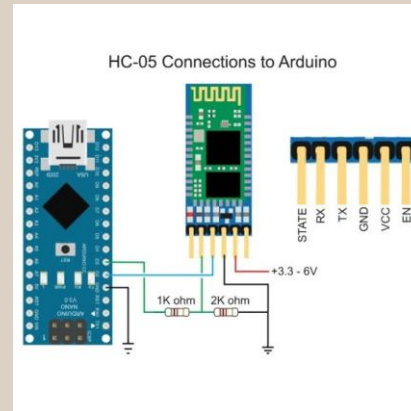
NANO PINOUT



PROJETO ELETRÔNICO - MÓDULOS

BLUETOOTH E O DIVISOR DE TENSÃO

TX – Transmissor
RX – Receptor (3V3)
Vcc – Alimentação
Gnd – Alimentação
HC-05 / HC -06

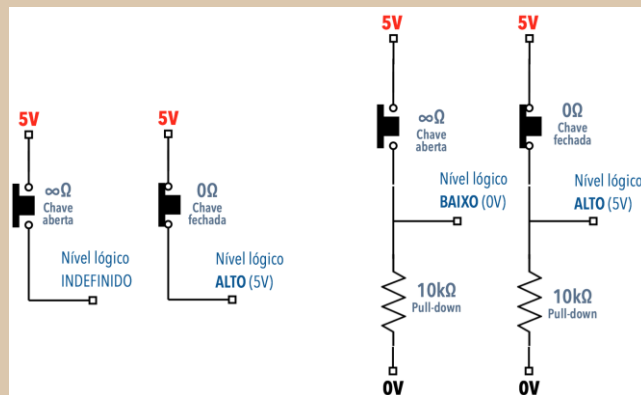


RÉLE E OS OPTOACOPLADORES

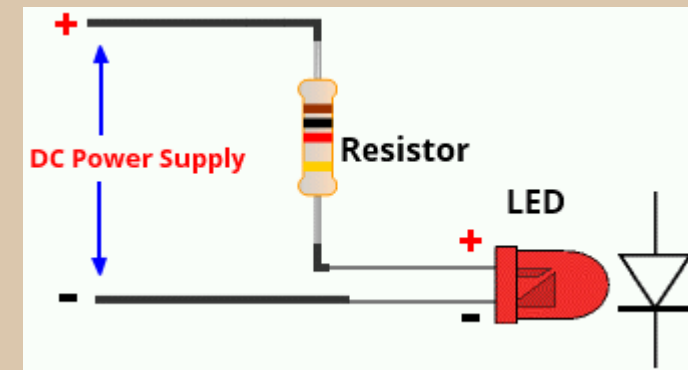
IN1 – Liga/Desliga
Vcc – Alimentação
Gnd – Alimentação
c/ opto – 2mA
s/ opto – 20mA



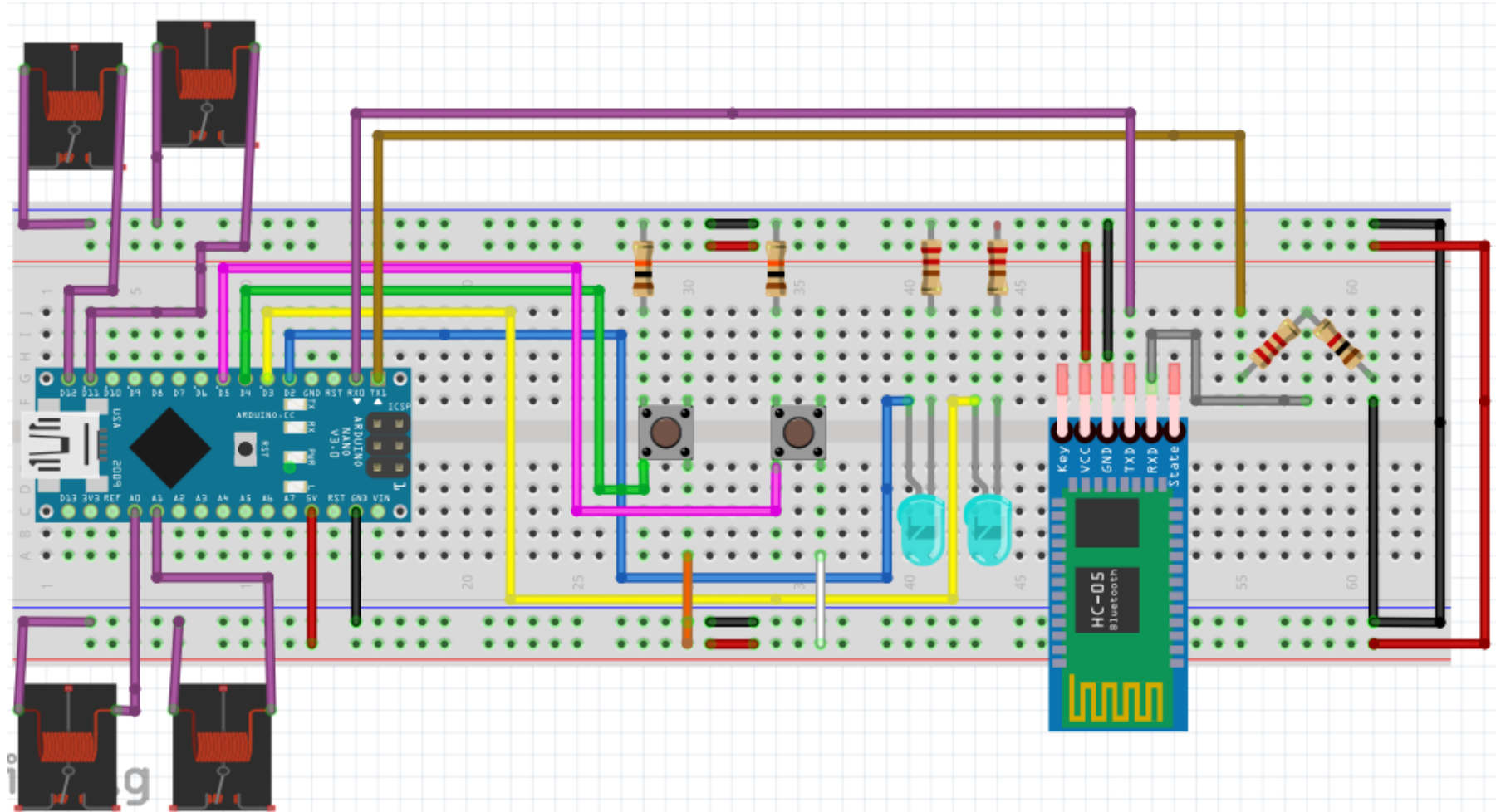
INTERRUPTORES E OS RESISTORES PULL DOWN



LEDs E OS RESISTORES LIMITADOR DE CORRENTE



PROJETO ELETRÔNICO - PROTOBOARD



PROJETO DO FIRMWARE – SKETCH ARDUINO

IDE DO ARDUINO

DECLARAÇÕES INICIAIS

VOID
SETUP

VOID LOOP

PINOS

VARIÁVEIS

INICIA
COMUNICAÇÃO
BLUETOOTH

RECEBE OS
COMANDOS DO
ANDROID

PROJETO DO FIRMWARE – SKETCH ARDUINO

Nessa primeira parte do Sketch da programação serão feitos três procedimentos, tais como:

- Definir os pinos do relé;
- Definir o pino analógico do botão pulsador;
- Incluir a biblioteca do Bluetooth.

Já na segunda parte do Sketch o **void setup** também serão vários procedimentos, tais como:

- Declarar os pinos de saídas do relé;
- Declarar o pino de entrada do botão pulsador;
- Colocar os pinos digitais do relé ligados.

Essa parte antes do **void loop** serão feitos vários procedimentos, tais como:

- Declarar a variável do estado do botão;
- Declarar os status do rele;
- Declarar o botão de comando do bluetooth.

```
#define pinRele 11
#define pinrele2 10
#define button A0

#include <SoftwareSerial.h>
SoftwareSerial btSerial(0, 1); // RX, TX
```

```
void setup() {

    pinMode(button, INPUT);

    pinMode(pinRele, OUTPUT);
    pinMode(pinrele2, OUTPUT);
    digitalWrite(pinRele, HIGH);
    digitalWrite(pinrele2, HIGH);
    btSerial.begin(9600);
}

boolean estadobotao;
boolean statusRele1 = LOW;
boolean statusRele2 = LOW;
String btComando;
```

PROJETO DO FIRMWARE – SKETCH ARDUINO

Essa parte já dentro do **void loop**, serão feitos alguns procedimentos como:

- **Leitura da do pino do botão e coloca-lo na variável estadodoboato;**
- **Verificação se o bluetooth está conectado;**
- **Se não estiver que possa mostra no monitor serial;**

Continuando no **void loop** agora dentro do **while** os procedimentos de comando serão:

- **Mostrar no monitor serial que o bluetooth esta conectado e dentro de while;**

```
void loop(){

    estadobotao = digitalRead(button);
    Serial.println (estadobotao);

    if (estadobotao == 0){
        digitalWrite(pinrele2, !statusRele2);
        statusRele2=!statusRele2;
        delay(500);
        Serial.println (" BluetoothConectadoForaDoWhile");
        Serial.println ("-----");
        Serial.println (estadobotao);
        Serial.println ("*****");
    }

    while (btSerial.available()) {
        Serial.println ("BluetoothConectadoDentroDoWhile");
        Serial.println ("-----");
        Serial.println (estadobotao);
    }
}
```

PARTE 1

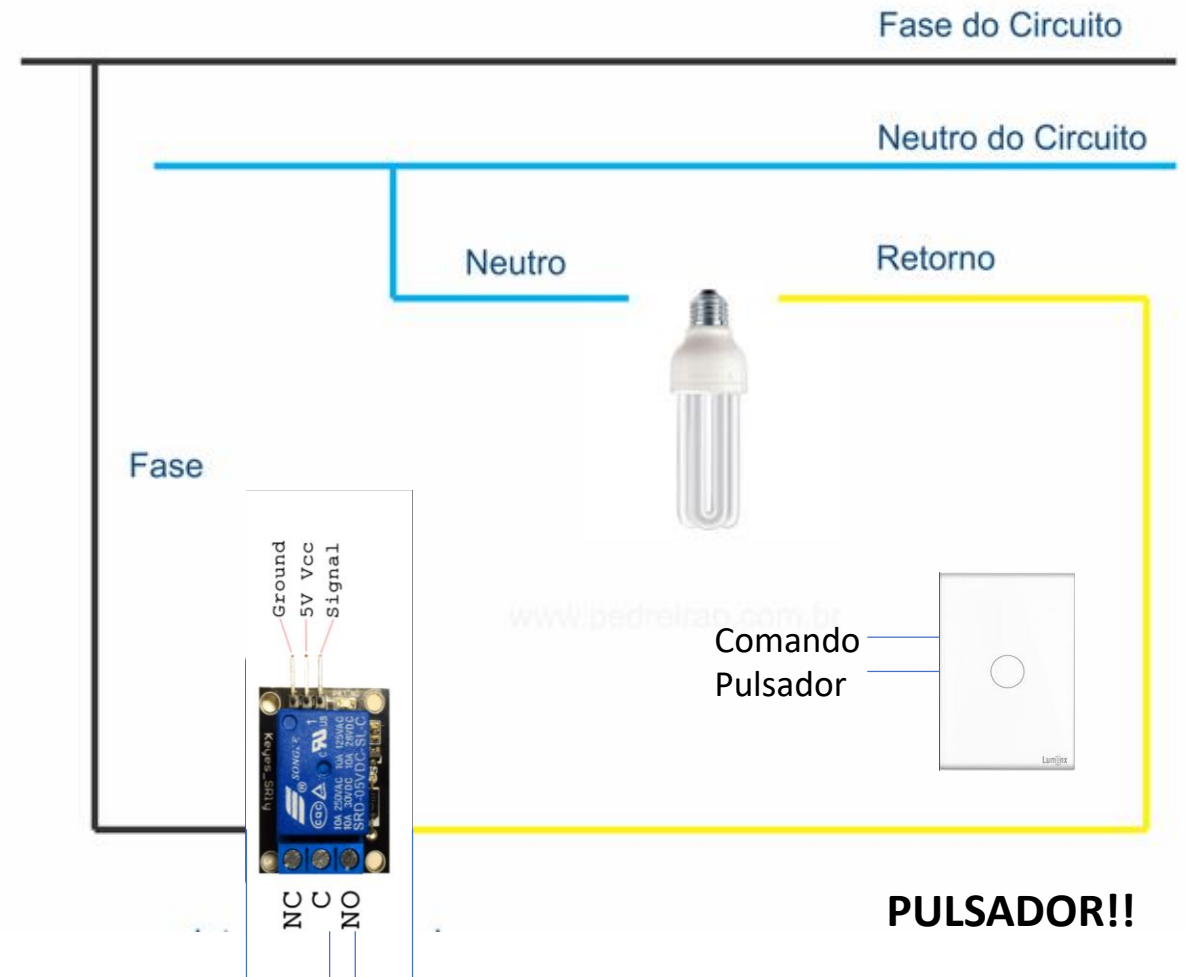
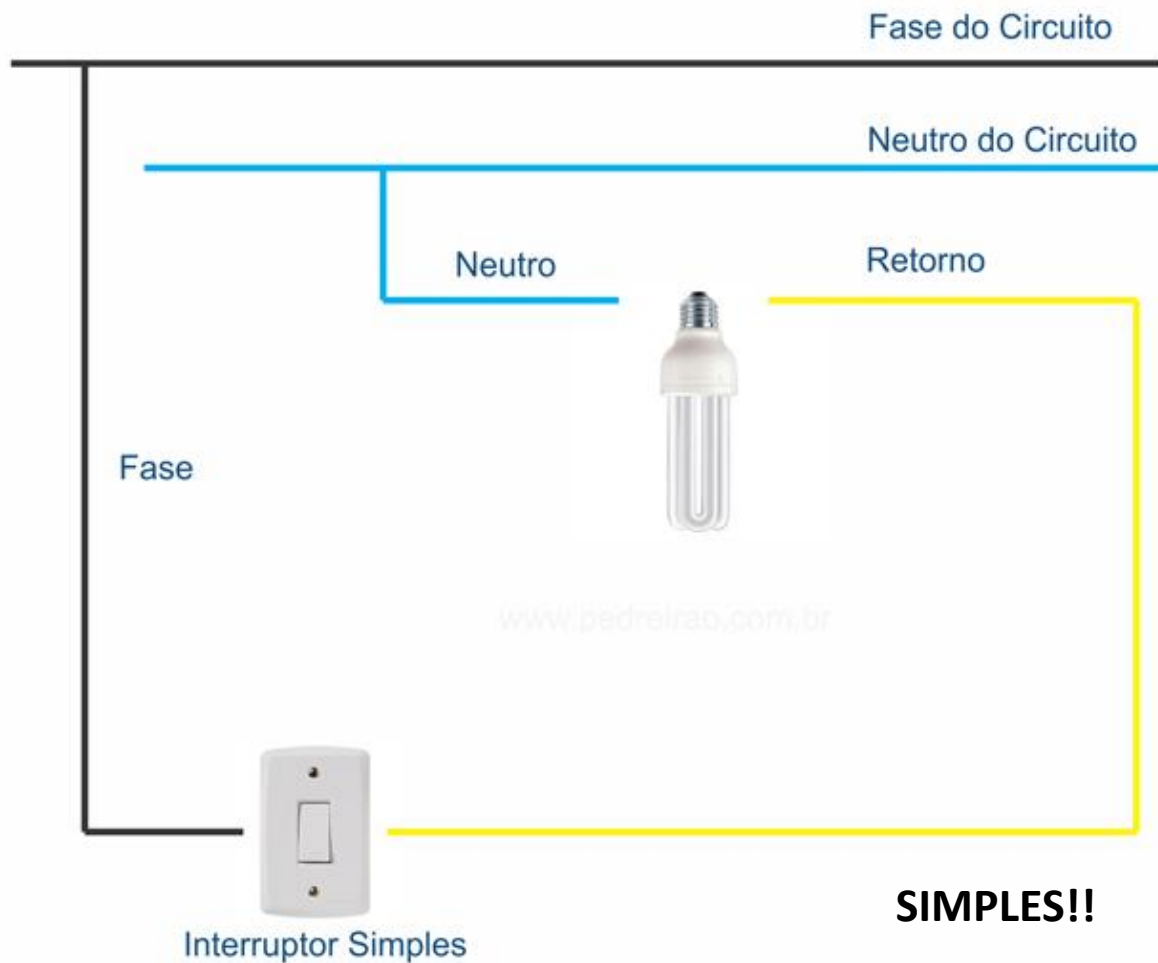
PROJETO DO FIRMWARE – SKETCH ARDUINO

- Declarar uma variável de comando de recepção (comandoRec) que receba o a leitura do serial do bluetooth;
- Executar a variável botão do bluetooth(btComando) e exibir no monitor serial o comando de recepção;
- Verificar e executar os comandos “comandoRec e do btComando”;
- E conforme os comandos dados atualizar os status dos pinos dos relés.

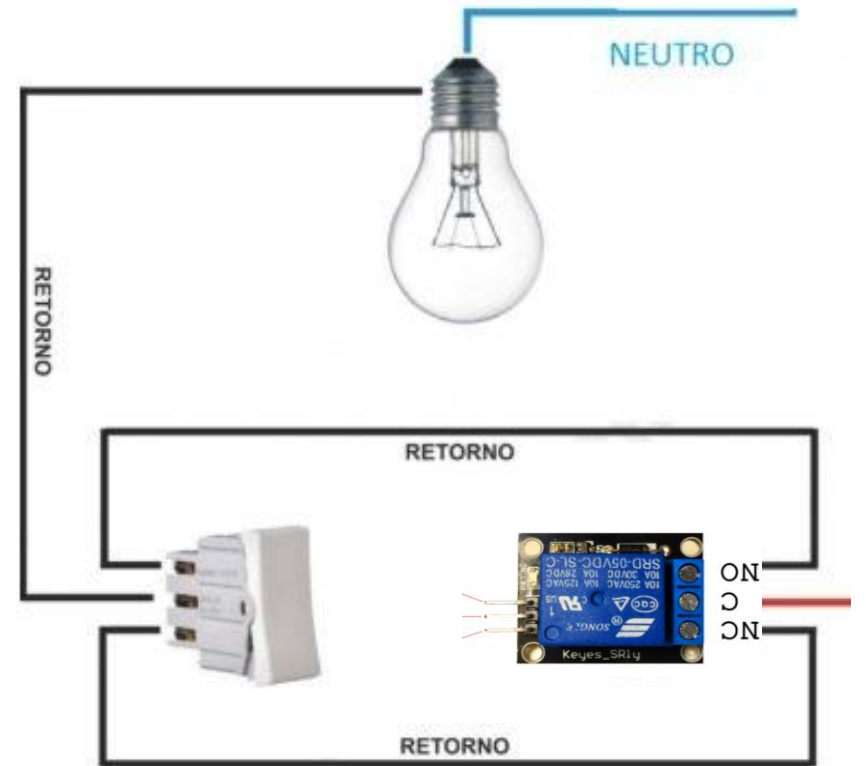
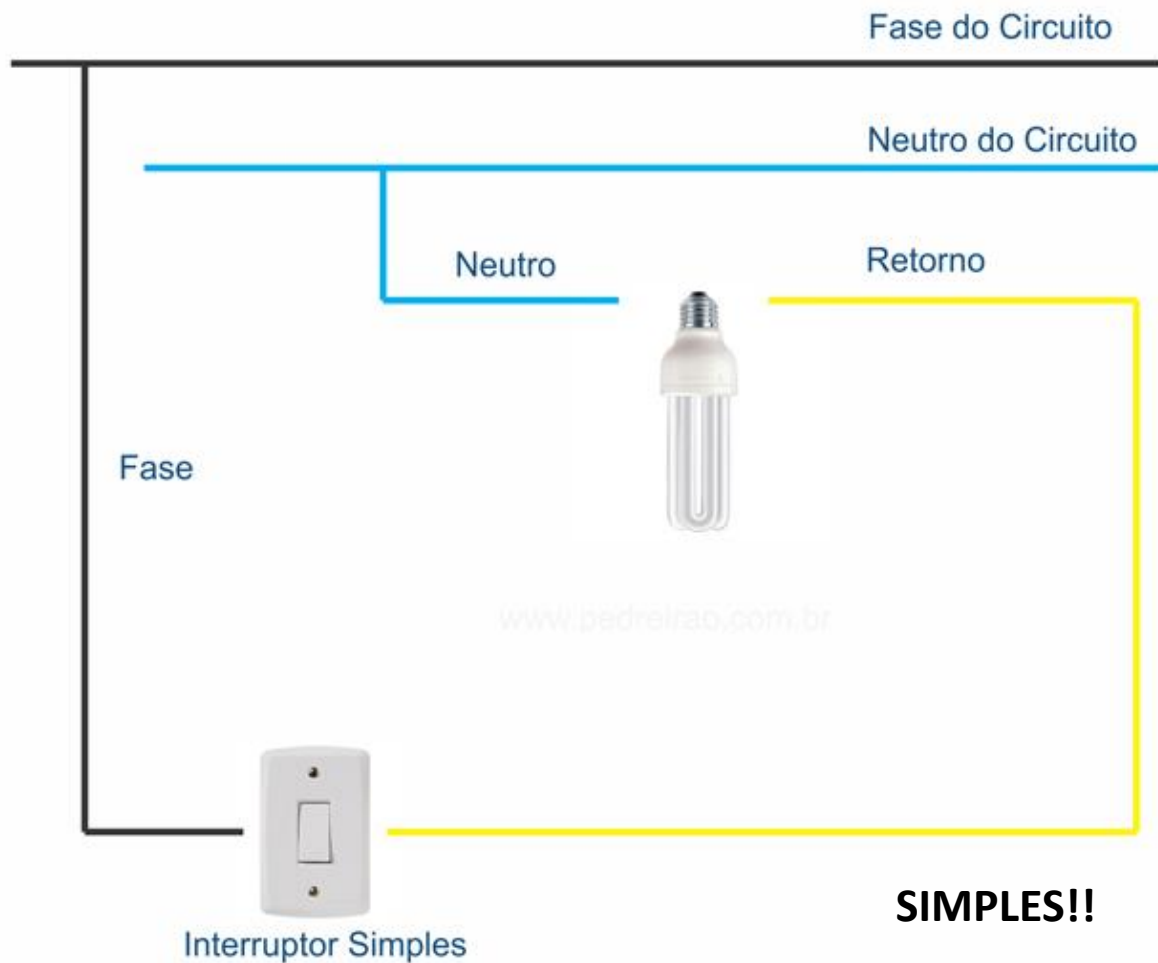
```
char comandoRec = btSerial.read();  
btComando += char( comandoRec );  
Serial.print(comandoRec);  
  
if (comandoRec == ';') {  
    if (btComando == "B1;") {  
        statusRele1=!statusRele1;  
        digitalWrite(pinRele, statusRele1);  
    }  
    if (btComando == "B2;") {  
        statusRele2=!statusRele2;  
        digitalWrite(pinrele2, statusRele2);  
    }  
    btComando = "";
```

PARTE 2

PROJETO ELETROTÉCNICO



PROJETO ELETROTÉCNICO



DESENVOLVIMENTO DE SOFTWARE

O **Android** é um sistema operacional para dispositivos móveis com telas sensíveis ao toque.

O **Android** é baseado no kernel Linux e possui licença código fonte aberto

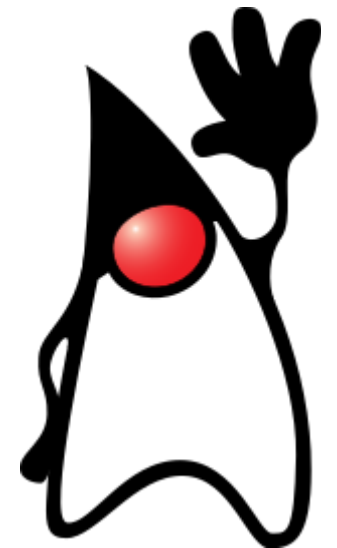
Possui **80%** do mercado de dispositivos móveis do mundo



O **App Inventor** é uma aplicação de código aberto para desenvolver aplicativos para o SO Android.

Utiliza uma linguagem baseada em **JAVA** com artifícios gráficos para facilitar a programação para iniciantes.

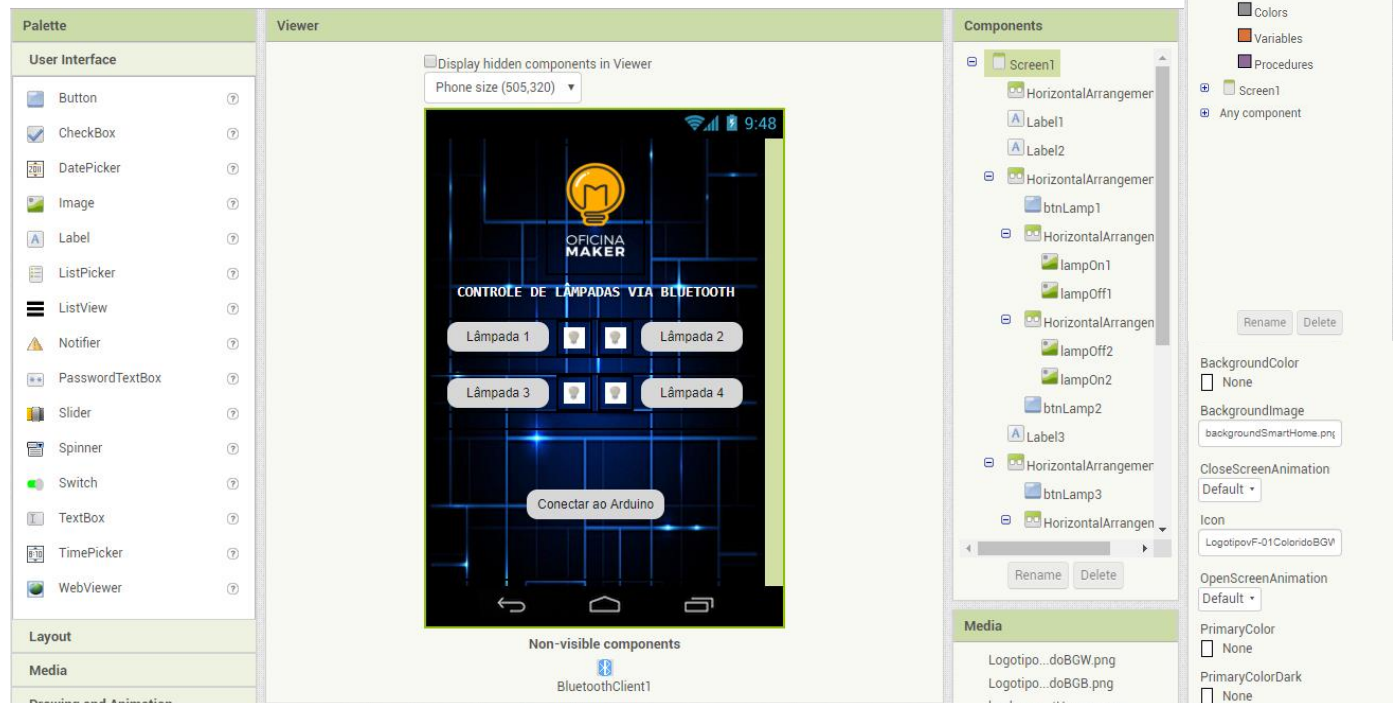
Desenvolvido por meio de uma parceria entre o **Google** e o **MIT**.



DESENVOLVIMENTO DE SOFTWARE

ai2.appinventor.mit.edu/

MODO DESIGN – FRONT END



MODO PROGRAMAÇÃO– BACK END

DESENVOLVIMENTO DE SOFTWARE

