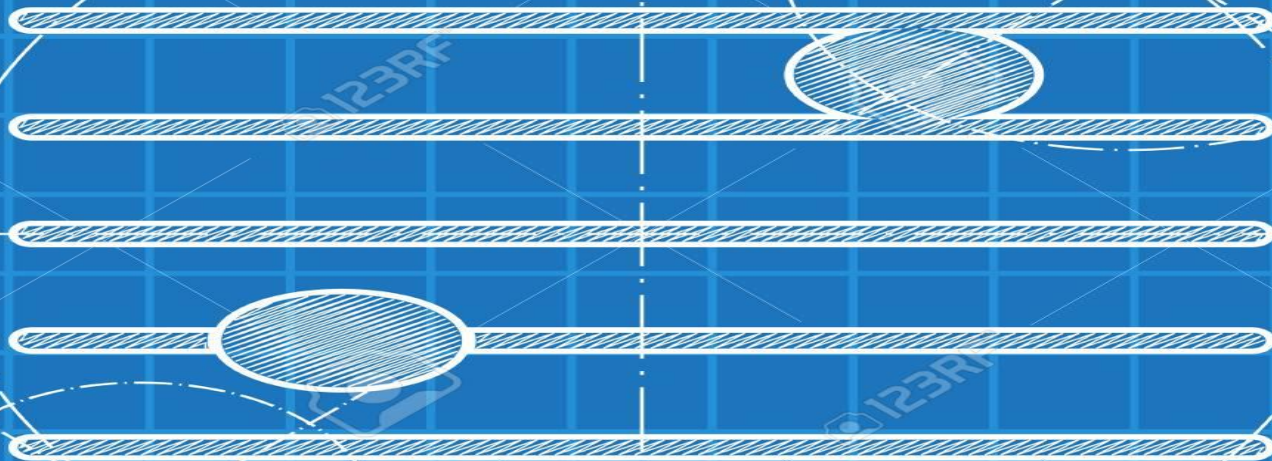


Projeto Sabiá



ICON

VECTOR
EPS 10

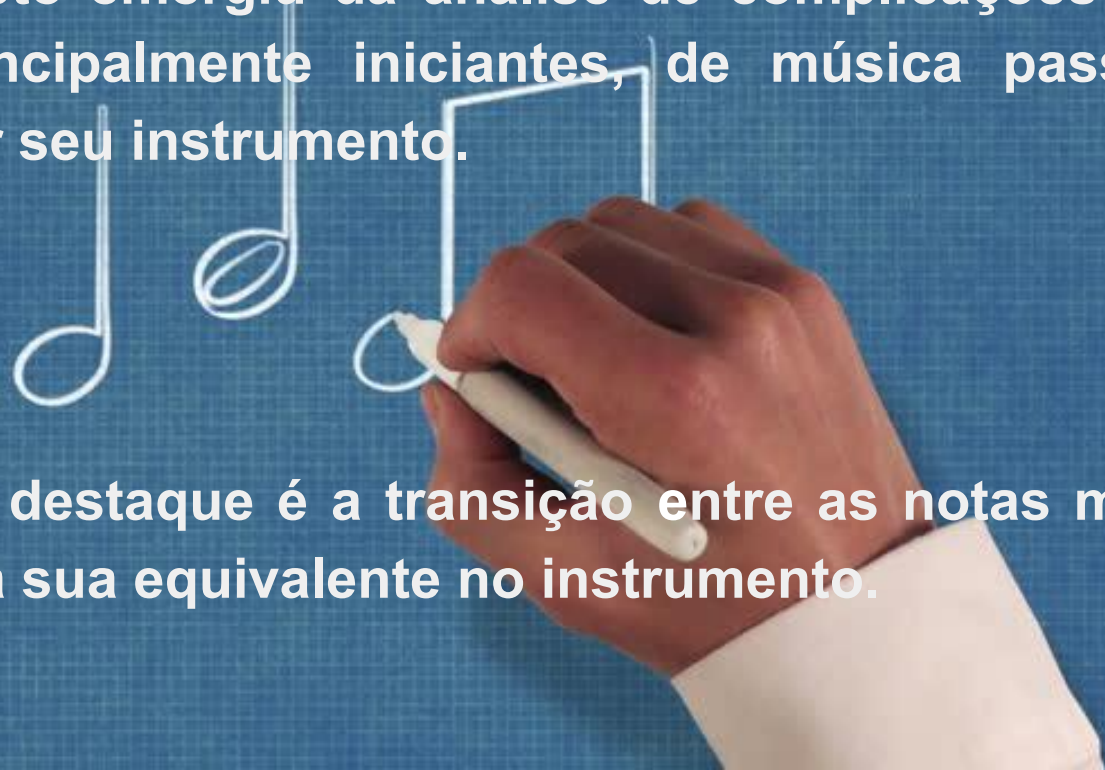
BLUEPRINT STYLE

1000 x 1000

Como surgiu ?

A ideia do projeto emergiu da análise de complicações que os estudantes ,principalmente iniciantes, de música passam ao aprender a tocar seu instrumento.

O problema em destaque é a transição entre as notas musicais na partitura para sua equivalente no instrumento.



May 8, 1928.

J. C. SWANSON

1,668,832

O que é ?

VIOLIN
Filed Nov. 29, 1921

2 Sheets-Sheet 1

É um indicador e tocador de notas musicais, que, a partir de soluções de hardware e software, faz a transição entre a nota na partitura e posição/som da nota no instrumento.

No protótipo o instrumento é o violino.

Fig. 1.



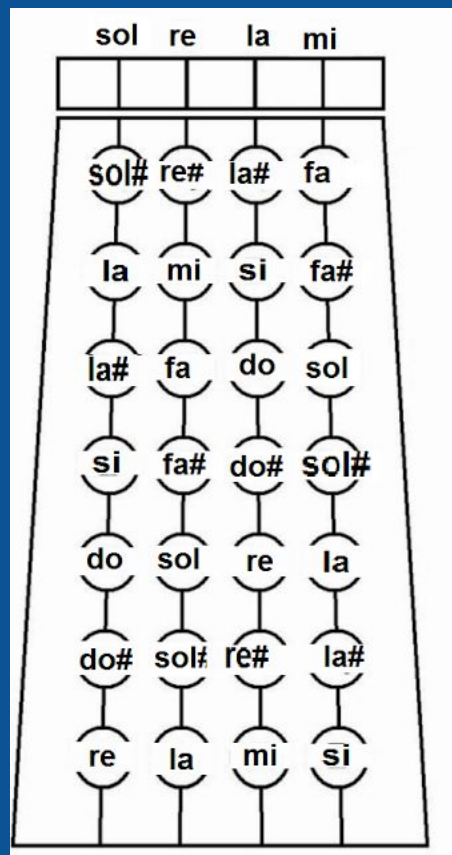
Fig. 2.



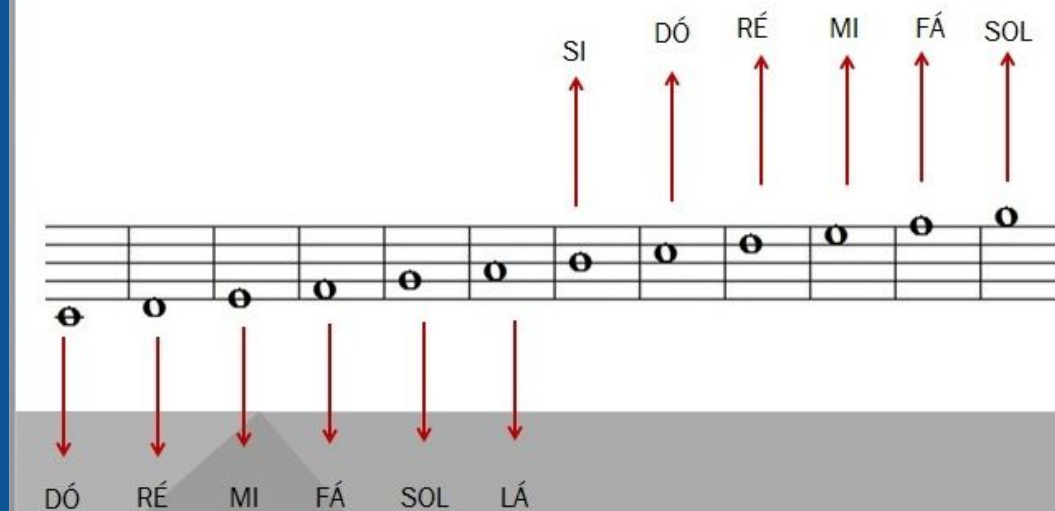
Fig. 3.



Inventor:
John C. Swanson.
By: Fred W. Chapman
Attorney.



Algunas notas

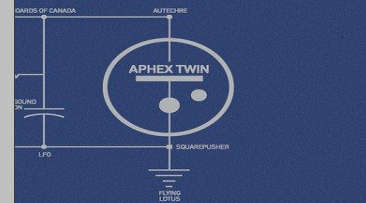
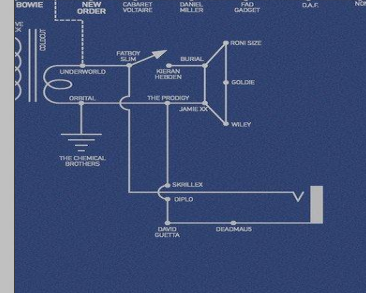
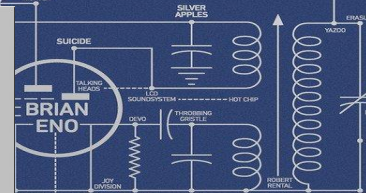
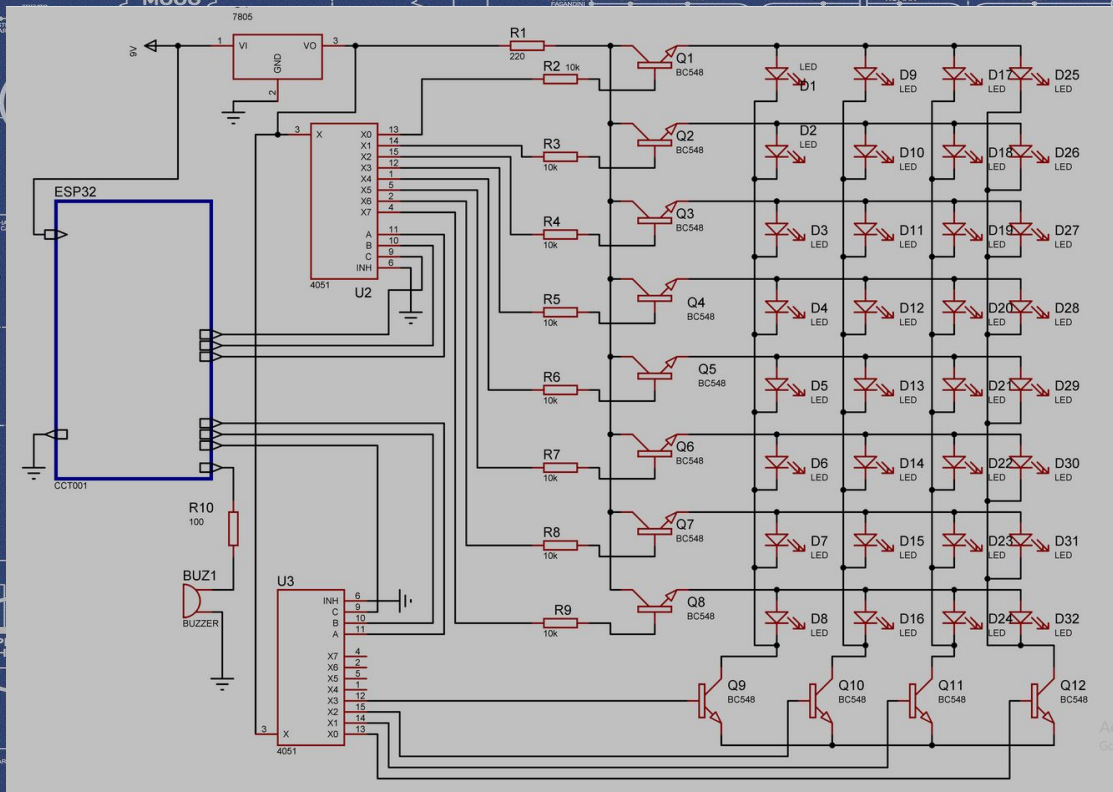


Do que é feito ?

- Leds
- Transistores
- Resistores
- Multiplexadores
- Resistores
- LM7805
- Buzzer
- Esp32
- Pelo preço de



CIRCUITO

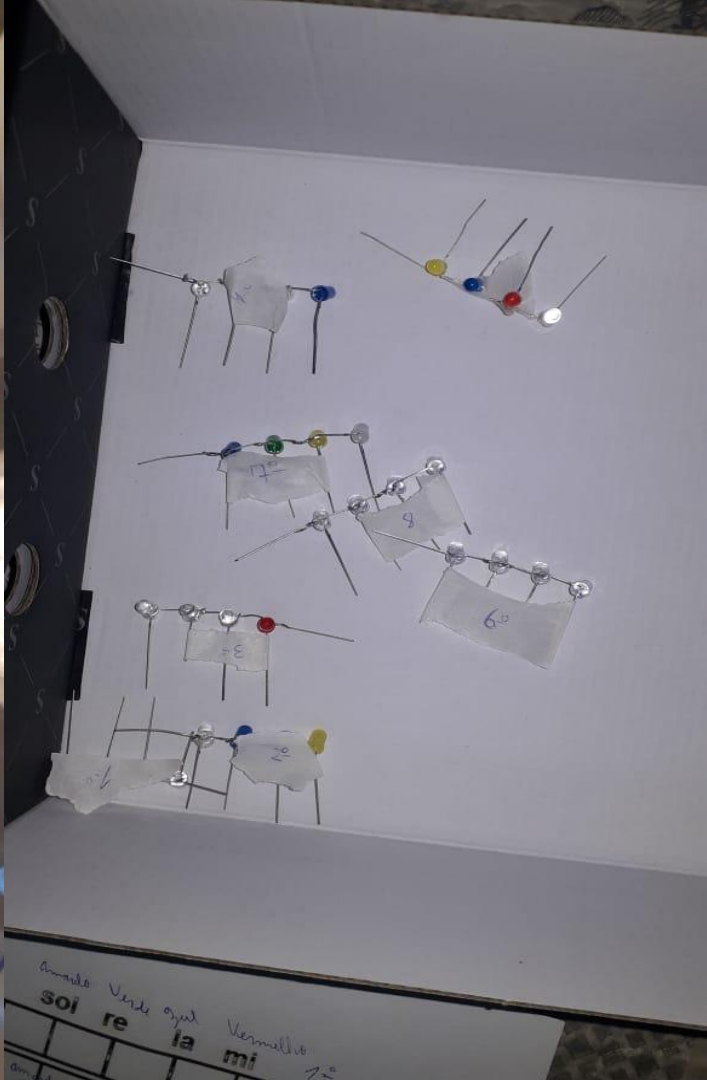
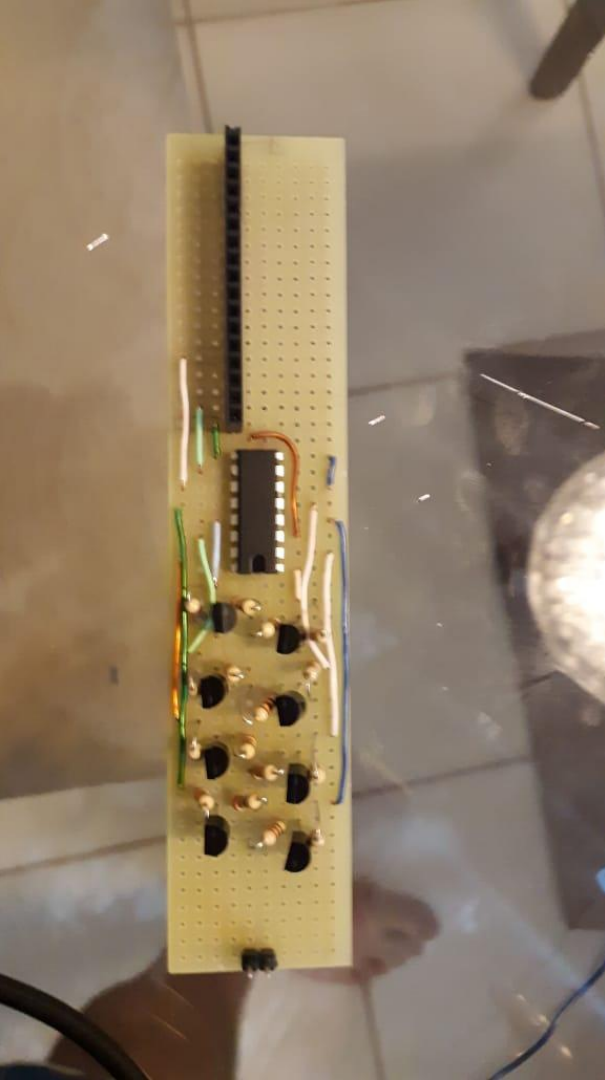


ELECTRIC LOVE
A BRIEF HISTORY OF ELECTRONIC MUSIC

SCALE: A brief history of Electronic music seen in a circuit diagram of a transistor.

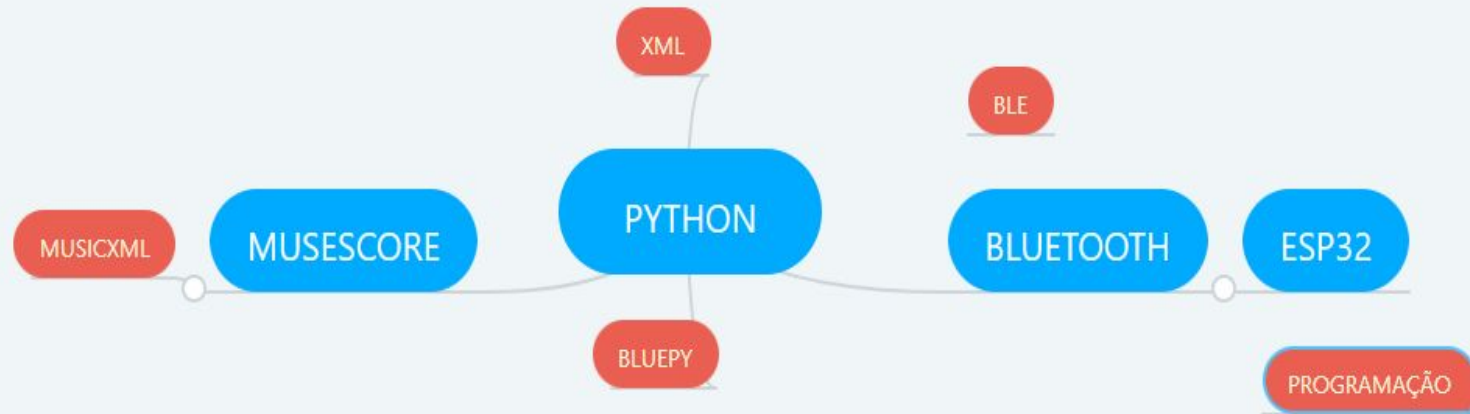
DATE: 19 - 1 - 1997 INCLUDES: Introduction, Components, Materials, & Power

104





E seu Software ?



Musescore/MUSICXML

The screenshot displays the Musescore 3.10.2 software interface. The main window shows a piano score for 'Præludium 10 BWV 855' in G major, 4/4 time. The score is written for two staves (treble and bass clef). The left sidebar contains various toolbars for editing, including a piano keyboard at the bottom. The right sidebar shows the 'Inspector' panel with settings for the selected element, which is a note. The status bar at the bottom indicates 'Note: Pitch: E5, Duration: 140, Voice: 1, Measure: 1, Beat: 1, Staff: 1'.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Representation of middle C on the treble clef created through MusicXML code.

Uso extenso de python

```
import xml.etree.ElementTree as ET

tree = ET.parse('test4.musicxml')
root = tree.getroot()
supp = ['C', 'D', 'E', 'F', 'G', 'A', 'B']
notas = []
bType = 0
refTime = 0
nTpos = ['1024th', '512th', '256th', '128th', '64th', '32nd', '16th', 'eighth', 'quarter', 'half', 'double']
nMultiplos = [1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8, 16, 32]
nTempos = [1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8, 16, 32]
refViolino = ['d4', 'C#4', 'C4', 'B3', 'A#3', 'A3', 'G#3', 'G3', 'A#4', 'A', 'G#4', 'G4', 'F#4', 'F4']
bpm = int(input('Escolha as BPM: '))
print("")

for measure in root[-1]:
    for note in measure:
        nota = []
        if note.tag == 'barline' or note.tag == 'print':
            continue
        if note.tag == 'attributes':
            bType = 4*(1/float(note[2][1].text))
            refTime = (bType*60)/bpm
            continue
        a = 0
        noteItems = [(x.tag, x.text) for x in note]
        pitchItems = [(x.tag, x.text) for x in note[0]]

        for i in noteItems:
            if i[0] == 'accidental':
                print(i[1])
                nota.append(i[1])
                continue
            if i[0] == 'type':
                print(i[1])
                nota.append(i[1])
                continue
            if i[0] == 'rest':
                nota.append('rest')
                continue

        for i in pitchItems:
            if i[0] == 'octave':
                print(i[1])
                nota.append(i[1])
                continue
            if i[0] == 'step':
```

```
                </pitch>
                <duration>1</duration>
                <voice>1</voice>
                <type>quarter</type>
                <accidental>sharp</accidental>
                <stem>up</stem>
            </note>
            <note default-x="184.38" default-y="-15.00">
                <pitch>
                    <step>C</step>
                    <alter>-1</alter>
                    <octave>5</octave>
                </pitch>
                <duration>1</duration>
                <voice>1</voice>
                <type>quarter</type>
                <accidental>flat</accidental>
                <stem>down</stem>
            </note>
            <note default-x="276.27" default-y="-25.00">
                <pitch>
                    <step>A</step>
                    <octave>4</octave>
                </pitch>
                <duration>1</duration>
                <voice>1</voice>
                <type>quarter</type>
                <accidental>natural</accidental>
                <stem>up</stem>
            </note>
            <note default-x="368.16" default-y="-40.00">
                <pitch>
                    <step>E</step>
                    <alter>1</alter>
                    <octave>4</octave>
                </pitch>
                <duration>1</duration>
                <voice>1</voice>
                <type>quarter</type>
                <accidental>sharp</accidental>
                <stem>up</stem>
            </note>
            <barline location="right">
                <bar-style>light-heavy</bar-style>
            </barline>
        </measure>
    </part>
</score-partwise>
```

BLE(Bluetooth Low Energy)

Bluetooth Low Energy for Smart Applications

Key Features

- Ultra low power consumption
- Low cost
- Small size
- Faster connection
- Secure

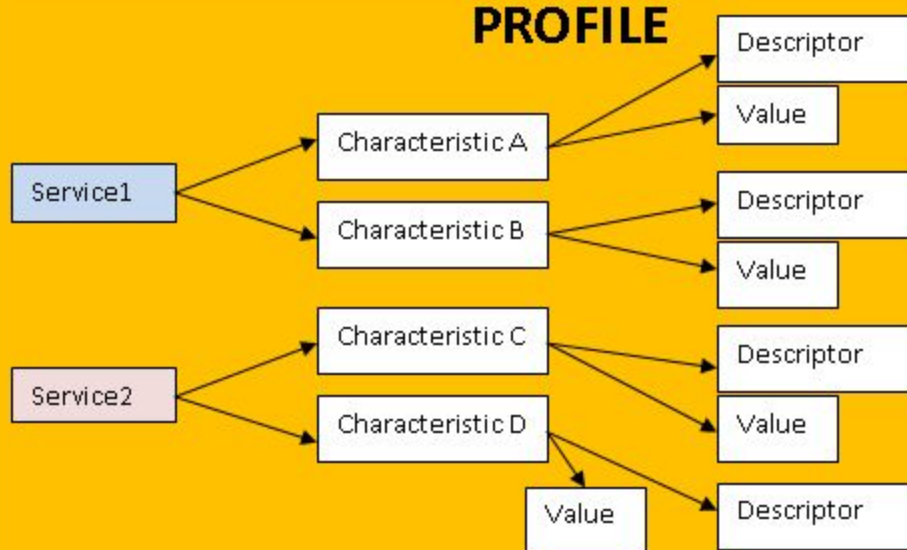


Variety of BLE Applications

- IoT
- Health care
- Smart Home Automation
- Smart Energy
- Advertisement



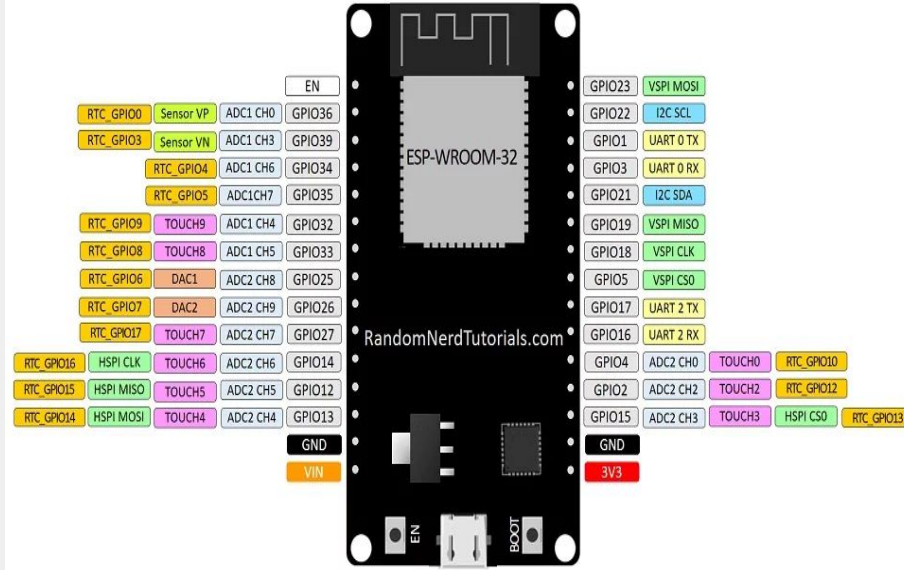
PROFILE



ESP32

ESP32 DEVKIT V1 - DOIT

version with 30 GPIOs



Que fim tomou ?

Após 3 semanas estudando, soldando, testando e debugando obteve-se um projeto de grande potencial educacional. Com uma ideia base que pode ser estendida para inúmeros outros instrumentos.

