Montain Wolves Airsoft Bomb

Project

Uma imagem com emblema, símbolo, logótipo, Marca registada

Descrição gerada automaticamente

# Introdução

No âmbito da unidade curricular, é necessário desenvolver e entregar dois projetos eletrônicos que atendam a diferentes requisitos mínimos. Para cumprir essas exigências, o tema escolhido foi um projeto que permite expansões e atende a todos os critérios propostos.

Como praticante de Airsoft e membro da equipe Montain Wolves, decidi criar um projeto pessoal que pudesse ser integrado na disciplina. Este projeto consiste em uma Fake Bomb para ser utilizada em ambientes de jogo de Airsoft.

O projeto possui múltiplas possibilidades de atualização, tais como:

Operar como um sistema simples de efeito de bomba,

Integrar-se a um protocolo de comunicação para interação com outros dispositivos durante o jogo,

Conectar-se a um servidor, permitindo, através de sensores de geolocalização, o monitoramento em tempo real de sua posição, criando assim uma espécie de "streaming" do jogo.

O dispositivo, em sua versão final, poderá apresentar as seguintes funcionalidades:

Configuração via Bluetooth,

Configuração manual,

Geolocalização do dispositivo,

Integração em uma rede LoRa.

Para a unidade curricular é necessário entregar dois projetos eletrônicos em que os mesmos têm de apresentar requisitos mínimos diferentes, contudo o tema escolhido é um projeto que permite o upgrade e que abrange todos os requisitos pedidos.

Sendo eu um jogador de Airsoft na equipa dos Montain Wolves decidi criar um projeto pessoal em que poderia enquadrar na unidade curricular. Projeto este que é uma Fake Bomb para ser utilizada em ambiente de jogo de Airsoft.

Este projeto contém muitas vertentes de atualização como:

- Utilizar o seu simples sistema de efeito bomba

- Utilizar a mesma para enquadrar num protocolo de comunicação e comunicar com outros dispositivos em jogo

- Enquadrar a mesma num servidor e assim através dos seus sensores de glocalização saber sempre em tempo real onde a mesma se encontra e assim criar uma espécie de stream do jogo em questão.

O dispositivo na sua entrega final poderá contar com as seguintes funcionalidades:

- Configuração através de Bluetooth,

- Configuração através de manual,

- Geolocalização do dispositivo,

- Inserção numa rede Lora.

Índice

[Introdução 3](#_Toc182936372)

[Tema do projeto 7](#_Toc182936373)

[Linguagem de programação 8](#_Toc182936374)

[C++ 8](#_Toc182936375)

[Java 8](#_Toc182936376)

[Microcontroladores 9](#_Toc182936377)

[Arduino 9](#_Toc182936378)

[Arduino Nano 9](#_Toc182936379)

[Arduino Uno 9](#_Toc182936380)

[Arduino Mega 10](#_Toc182936381)

[ESP82 11](#_Toc182936382)

[ESP32 11](#_Toc182936383)

[Shields 12](#_Toc182936384)

[Softwares utilizados 13](#_Toc182936385)

[Arduino IDE 13](#_Toc182936386)

[Android Studio 14](#_Toc182936387)

[Git 14](#_Toc182936388)

[Repositório GitHub 14](#_Toc182936389)

[Componentes Utilizados no Projeto 15](#_Toc182936390)

[Esp32 Dev Kit V4 16](#_Toc182936391)

[Protoboard 16](#_Toc182936392)

[Placa PCB perfurada 17](#_Toc182936393)

[Display OLED 17](#_Toc182936394)

[Display Liquid Crystal I2C 17](#_Toc182936395)

[Módulo GPS 18](#_Toc182936396)

[Teclado 4x4 18](#_Toc182936397)

[Buzzer 19](#_Toc182936398)

[Atomizador piezoelétrico 19](#_Toc182936399)

[Esquemas e layouts 20](#_Toc182936400)

[Esquema elétrico 20](#_Toc182936401)

[Layout para PCB 20](#_Toc182936402)

[Modelo 3D 21](#_Toc182936403)

[Tampa 21](#_Toc182936404)

[Original 21](#_Toc182936405)

[Modelado 21](#_Toc182936406)

[Corpo 21](#_Toc182936407)

[Original 21](#_Toc182936408)

[Modelado 22](#_Toc182936409)

[Aplicação Android 22](#_Toc182936410)

[Estrutura e design 23](#_Toc182936411)

[Código Java 27](#_Toc182936412)

[Código Arduino 32](#_Toc182936413)

[Bibliografia 33](#_Toc182936414)

[Figura 1 - Protejo Inicial 7](#_Toc182936415)

[Figura 2 - Arduino Nano 9](#_Toc182936416)

[Figura 3 - Arduino Uno R4 10](#_Toc182936417)

[Figura 4 - Arduino Mega 10](#_Toc182936418)

# Tema do projeto

Para a construção física do projeto, foi utilizado como base o modelo descrito a seguir. Os links presentes na bibliografia apresentam todos os componentes do projeto original, incluindo peças 3D e código. O código do projeto original é amplamente utilizado em diversos projetos semelhantes no universo do Airsoft. No entanto, a ideia central deste trabalho foi reutilizar a estrutura física com o propósito de adaptá-la às nossas necessidades específicas. Assim, a estrutura serviu apenas como um ponto de partida e precisou ser redesenhada. Para esse processo, foi utilizado o software de modelagem 3D SolidWorks.

Para a construção do projeto fisicamente foi utilizado o modelo abaixo como base. Os links na bibliografia iram apresentar todos os componentes do projeto original desde peças 3D como código. O projeto original o seu código é um código muito utilizado em amplos projetos do mesmo género no Airsoft, contudo a ideia era reutilizar a estrutura física para assim a mesma ser alterada consoante a nossa necessidade, logo o objetivo é apenas servir de base, logo as mesmas terão de ser redesenhadas, para isso será utilizado o software de modelagem 3D SolidWorks.



Figura 1 - Protejo Inicial

# Linguagem de programação

## No desenvolvimento do projeto, foram empregadas diferentes linguagens de programação, de acordo com as necessidades específicas de cada ambiente. O C++ foi utilizado para a programação do microcontrolador, garantindo eficiência e controle detalhado dos recursos embarcados. Já a linguagem Java foi aplicada no desenvolvimento do aplicativo Android, proporcionando uma interface amigável e uma integração robusta com o sistema móvel.

## C++

C++ é uma linguagem de programação de propósito geral conhecida por ser poderosa e versátil, derivada do C. Ela oferece suporte à programação orientada a objetos, programação genérica e programação procedural. Seu principal diferencial é a combinação de eficiência de baixo nível (próxima ao hardware) com recursos de alto nível, o que a torna ideal para sistemas de software complexos, como sistemas operacionais, drivers de dispositivos, aplicações em tempo real, jogos e aplicações de alta performance.

C++ oferece suporte a encapsulamento, herança e polimorfismo, permitindo modularidade e reuso de código, além de controle detalhado de recursos, como memória, por meio de ponteiros e gerenciamento manual. Sua flexibilidade também se estende ao desenvolvimento de sistemas embarcados, como o ESP32, onde é amplamente utilizado para criar aplicações eficientes e com recursos avançados, tirando proveito das capacidades de conectividade Wi-Fi e Bluetooth do microcontrolador.

## Java

Java é uma linguagem de programação orientada a objetos, conhecida por sua portabilidade, robustez e facilidade de uso. Desenvolvida para ser independente de plataforma através da Máquina Virtual Java (JVM), ela é amplamente utilizada no desenvolvimento de aplicativos, especialmente para dispositivos Android.

No contexto do projeto, Java está sendo utilizado para o desenvolvimento de um aplicativo Android. Essa escolha permite a criação de interfaces interativas e a integração de funcionalidades específicas para o controle e monitoramento do dispositivo desenvolvido. A flexibilidade da linguagem, aliada a um vasto ecossistema de bibliotecas e ferramentas, facilita o desenvolvimento de aplicações móveis robustas e escaláveis, garantindo uma experiência de usuário fluida.

# Microcontroladores

Os microcontroladores desempenham o papel de "cérebro" neste projeto, pois são responsáveis, por meio da programação em C++, pelo controle da lógica do sistema. Eles permitem a leitura de sinais de entrada e a gestão das saídas, viabilizando o funcionamento desejado do dispositivo. Atualmente, há diversos tipos de microcontroladores disponíveis no mercado, dos quais alguns dos principais serão mencionados mais adiante neste trabalho.

Os microcontroladores serão o cérebro neste projeto pois é o mesmo que é responsável de através da linguagem de programação C++ controlar a parte lógica e assim podermos ler sinais de entrada e gerir as nossas saídas para assim obtermos o pretendido.

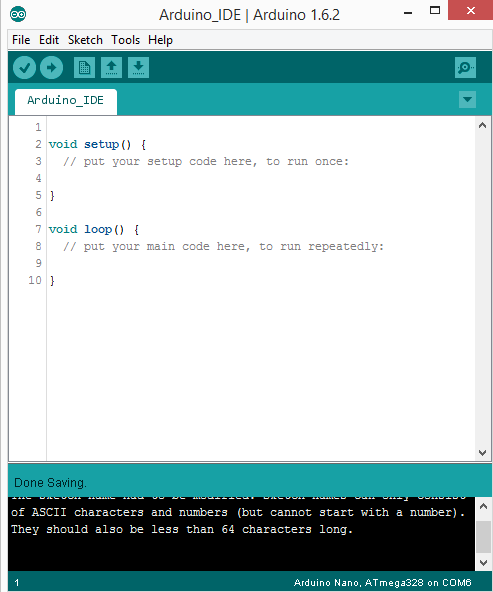
Existem muitos tipos de microcontroladores no mercado ao qual irei mencionar alguns principais mais abaixo.

# Softwares utilizados

## Arduino IDE

O Arduino IDE (Integrated Development Environment) é um ambiente de desenvolvimento integrado utilizado para programar placas Arduino e outros microcontroladores compatíveis. Ele oferece uma interface simples e acessível, projetada para facilitar a programação e a prototipagem de dispositivos eletrônicos, mesmo para iniciantes. O Arduino IDE suporta a linguagem de programação baseada em C/C++, tornando o desenvolvimento prático e intuitivo por meio de funções predefinidas e bibliotecas amplamente utilizadas pela comunidade.

Com o Arduino IDE, é possível escrever, compilar e carregar código para microcontroladores através de uma conexão USB. A ferramenta oferece recursos como detecção automática de portas seriais, um monitor serial para depuração em tempo real, além de um vasto repositório de exemplos e bibliotecas que ajudam a acelerar o desenvolvimento de projetos. Sua versatilidade e compatibilidade com uma ampla gama de placas, incluindo módulos baseados no ESP32, tornam o Arduino IDE uma escolha popular para prototipagem de sistemas embarcados, projetos IoT e automação em geral.



## Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android. Baseado no IntelliJ IDEA, o Android Studio foi projetado pela Google para oferecer uma experiência de desenvolvimento poderosa e flexível, fornecendo todas as ferramentas necessárias para criar, depurar e testar aplicativos móveis para dispositivos Android.

O Android Studio oferece um editor de código inteligente com sugestões e autocompletar, suporte integrado ao desenvolvimento com a linguagem Java, Kotlin e C++, além de ferramentas gráficas de design para construir interfaces de usuário de forma visual. Possui também emuladores para testar aplicativos em uma ampla gama de dispositivos e configurações, bem como um sistema de compilação baseado no Gradle, que facilita a automação de tarefas e gerência de dependências.

O ambiente ainda traz ferramentas para otimização de desempenho, depuração de código, testes automatizados, suporte a desenvolvimento de aplicações para diferentes tamanhos de tela e dispositivos, além de integração com o Firebase para fornecer funcionalidades como autenticação, armazenamento de dados em tempo real, entre outros. Tudo isso faz do Android Studio a solução mais robusta e completa para o desenvolvimento de aplicativos Android.

## Git

Git é um sistema de controle de versão distribuído amplamente utilizado no desenvolvimento de software. Ele permite que desenvolvedores rastreiem alterações em seu código-fonte, colaborem de maneira eficiente em equipes e revertam para versões anteriores do projeto quando necessário. Criado por Linus Torvalds em 2005, o Git é conhecido por sua rapidez, confiabilidade e flexibilidade.

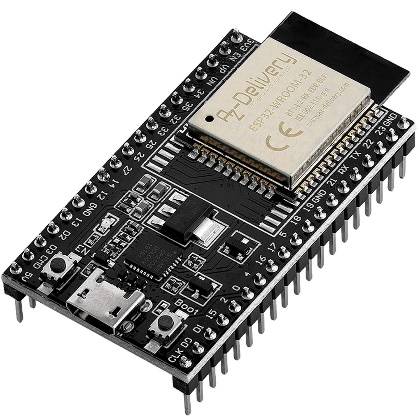
Com Git, as mudanças são registradas em um repositório que armazena todo o histórico de modificações, possibilitando o controle detalhado das versões do código. Cada desenvolvedor possui uma cópia local do repositório, permitindo o trabalho offline e a realização de experimentos sem impactar a versão principal do projeto. Ele utiliza conceitos como branches (ramificações) para que múltiplos desenvolvedores possam trabalhar em diferentes funcionalidades ou correções de forma isolada antes de mesclá-las ao projeto principal.

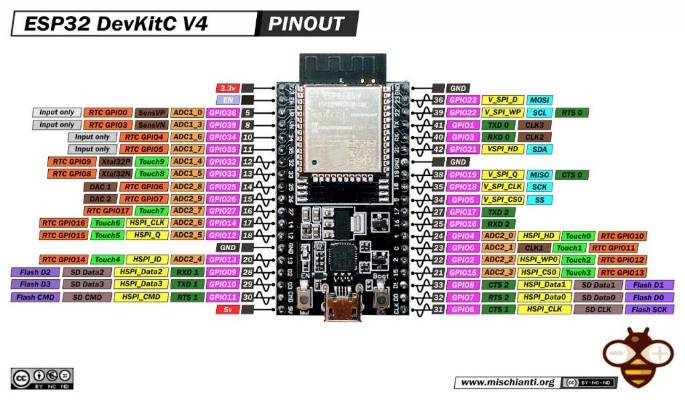
Além disso, Git facilita a colaboração por meio de plataformas de hospedagem como GitHub, GitLab e Bitbucket, que oferecem recursos adicionais como gerenciamento de issues, revisões de código, e integração contínua (CI/CD). Essas funcionalidades tornam Git uma ferramenta essencial para equipes de desenvolvimento modernas, ajudando a gerenciar o ciclo de vida dos projetos de forma eficiente e segura.



# Componentes Utilizados no Projeto

### Esp32 Dev Kit V4

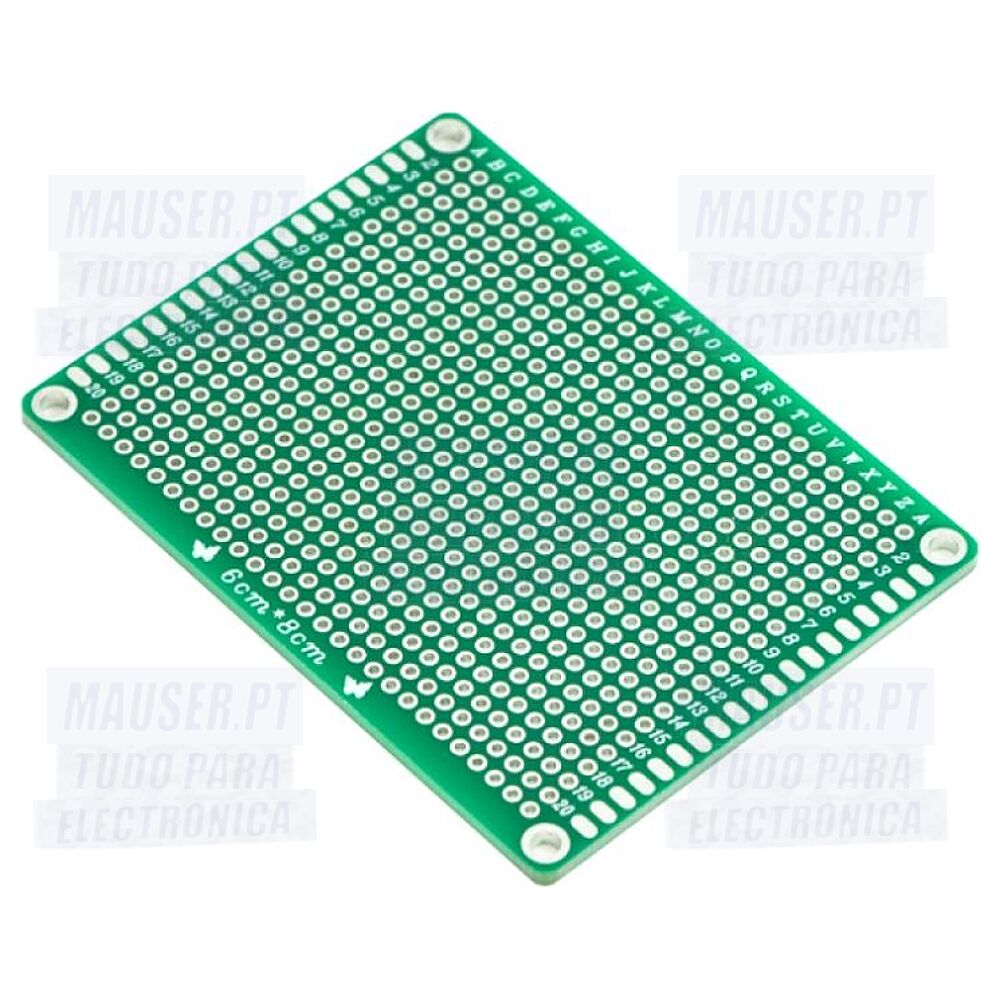




### Protoboard



### Placa PCB perfurada



### Display OLED



### Display Liquid Crystal I2C



### Módulo GPS



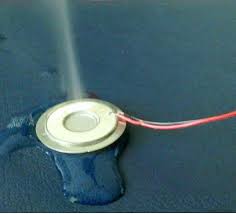
### Teclado 4x4



### Buzzer



### Atomizador piezoelétrico





### Leds WS2812b

# Esquemas e layouts

## Esquema elétrico

Uma imagem com texto, diagrama, Esquema, file

Descrição gerada automaticamente

## Layout para PCB

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, eletrónica, Engenharia eletrónica, circuito

Descrição gerada automaticamente

Uma imagem com captura de ecrã, Retângulo, Azul elétrico, Azul majorelle

Descrição gerada automaticamente

# Modelo 3D

## Tampa

### Original

Uma imagem com dados

Descrição gerada automaticamente com confiança baixa

### Modelado

Uma imagem com Retângulo, entrada, design

Descrição gerada automaticamente

Uma imagem com círculo, Retângulo, design

Descrição gerada automaticamente

## Corpo

### Original

### Modelado

# Aplicação Android

## Estrutura e design

Uma imagem com texto, captura de ecrã, póster, logótipo

Descrição gerada automaticamente

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/teamLogo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/montain\_wolves\_logo"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <ImageButton  
 android:id="@+id/play\_pause\_intro"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="5dp"  
 android:background="@null"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_pause" />  
  
 <TextView  
 android:id="@+id/editText\_inOut"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="90dp"  
 android:text="@string/in\_out\_layout"  
 android:textColor="@color/green"  
 android:textSize="30dp"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
 <Switch  
 android:id="@+id/sw\_gps"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="50dp"  
 android:background="@color/white"  
 android:text="@string/switch\_gps\_off"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 android:textColor="@color/blue"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Switch  
 android:id="@+id/sw\_sound"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="36dp"  
 android:background="@color/white"  
 android:text="@string/switch\_sound\_off"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 android:textColor="@color/blue"  
 app:layout\_constraintBottom\_toBottomOf="@+id/sw\_gps"  
 app:layout\_constraintStart\_toEndOf="@+id/sw\_gps"  
 app:layout\_constraintTop\_toTopOf="@+id/sw\_gps"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Switch  
 android:id="@+id/sw\_leds"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/switch\_leds\_off"  
 android:background="@color/white"  
 android:textSize="25dp"  
 android:textColor="@color/blue"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="@+id/sw\_gps"  
 app:layout\_constraintStart\_toStartOf="@+id/sw\_gps"  
 app:layout\_constraintTop\_toBottomOf="@+id/sw\_gps" />  
  
 <Switch  
 android:id="@+id/sw\_smoke"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="25dp"  
 android:textColor="@color/blue"  
 android:background="@color/white"  
 android:text="@string/switch\_smoke\_off"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="@+id/sw\_sound"  
 app:layout\_constraintStart\_toStartOf="@+id/sw\_sound"  
 app:layout\_constraintTop\_toBottomOf="@+id/sw\_sound" />  
  
 <TextView  
 android:id="@+id/editText\_dataGame"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="128dp"  
 android:layout\_marginTop="20dp"  
 android:text="@string/game\_data\_layout"  
 android:textColor="@color/green"  
 android:textSize="30dp"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/sw\_smoke" />  
  
 <EditText  
 android:id="@+id/edit\_text\_time\_game"  
 android:layout\_width="150dp"  
 android:layout\_height="50dp"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="30dp"  
 android:background="@color/white"  
 android:ems="10"  
 android:hint="@string/time\_of\_game"  
 android:inputType="number"  
 android:textAlignment="center"  
 android:textSize="30dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText\_dataGame" />  
  
 <EditText  
 android:id="@+id/edit\_text\_players"  
 android:layout\_width="150dp"  
 android:layout\_height="50dp"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="30dp"  
 android:background="@color/white"  
 android:ems="10"  
 android:hint="@string/num\_of\_players"  
 android:textSize="30dp"  
 android:inputType="number"  
 android:textAlignment="center"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/edit\_text\_time\_game" />  
  
 <EditText  
 android:id="@+id/edit\_text\_bomb\_Code"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="30dp"  
 android:background="@color/white"  
 android:ems="10"  
 android:hint="@string/bomb\_code"  
 android:textSize="30dp"  
 android:inputType="number"  
 android:textAlignment="center"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/edit\_text\_players" />  
  
 <Button  
 android:id="@+id/btn\_send\_config"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="40dp"  
 android:background="@color/red"  
 android:text="@string/send\_config"  
 android:textSize="30dp"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/edit\_text\_bomb\_Code" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## Código Java

package com.example.montainwolvesapp;  
  
import static android.content.ContentValues.*TAG*;  
  
import android.bluetooth.BluetoothAdapter;  
import android.bluetooth.BluetoothDevice;  
import android.bluetooth.BluetoothSocket;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.content.pm.PackageManager;  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
import android.widget.Switch;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import java.io.IOException;  
import java.io.OutputStream;  
import java.util.UUID;  
  
public class MainActivity extends AppCompatActivity {  
  
 Switch gps, leds, sound, smoke;  
 Button sendConfig;  
 ImageButton playPause;  
 ImageView teamLogo;  
 MediaPlayer mediaPlayer = new MediaPlayer();  
 EditText timeGame, numPlayers, bombCode;  
 boolean musicSound;  
  
 private BluetoothAdapter bluetoothAdapter;  
 private BluetoothSocket bluetoothSocket;  
 private BluetoothDevice bluetoothDevice;  
 private OutputStream outputStream;  
  
 private final String DEVICE\_ADDRESS = "08:A6:F7:20:B3:1E"; // Coloque o endereço MAC do seu módulo Bluetooth  
 private final UUID MY\_UUID = UUID.*fromString*("00001101-0000-1000-8000-00805F9B34FB");  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 bluetoothAdapter = BluetoothAdapter.*getDefaultAdapter*();  
 bluetoothDevice = bluetoothAdapter.getRemoteDevice(DEVICE\_ADDRESS);  
  
 mediaPlayer = MediaPlayer.*create*(this, R.raw.*intro*);  
 mediaPlayer.start();  
 playPause = (ImageButton) findViewById(R.id.*play\_pause\_intro*);  
 gps = (Switch) findViewById(R.id.*sw\_gps*);  
 leds = (Switch) findViewById(R.id.*sw\_leds*);  
 smoke = (Switch) findViewById(R.id.*sw\_smoke*);  
 sound = (Switch) findViewById(R.id.*sw\_sound*);  
 sendConfig = (Button) findViewById(R.id.*btn\_send\_config*);  
 timeGame = (EditText) findViewById(R.id.*edit\_text\_time\_game*);  
 numPlayers = (EditText) findViewById(R.id.*edit\_text\_players*);  
 bombCode = (EditText) findViewById(R.id.*edit\_text\_bomb\_Code*);  
  
 timeGame.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message = "TIME" + timeGame.getText().toString() + "\n";  
 sendData(message);  
 }  
 });  
  
 numPlayers.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message = "TEAM" + numPlayers.getText().toString() + "\n";  
 sendData(message);  
 }  
 });  
  
 bombCode.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (bombCode.length() == Integer.*parseInt*(numPlayers.getText().toString())){  
 String message = "CODE" + bombCode.getText().toString() + "\n";  
 sendData(message);  
 }else{  
 Toast.*makeText*(MainActivity.this,  
 "Bomb Code Size let be the same of Players Number",  
 Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 });  
  
  
 playPause.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 introSound();  
 }  
 });  
  
 gps.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (gps.isChecked()) {  
 String message = "gpsON\n";  
 sendData(message);  
 } else {  
 String message = "gpsOFF\n";  
 sendData(message);  
 }  
 }  
 });  
  
 leds.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message;  
 if (leds.isChecked()) {  
 leds.setText(R.string.*switch\_leds\_on*);  
 message = "ledsON\n";  
 } else {  
 leds.setText(R.string.*switch\_leds\_off*);  
 message = "ledsOFF\n";  
  
 }  
 sendData(message);  
 }  
 });  
  
 smoke.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message;  
 if (smoke.isChecked()) {  
 smoke.setText(R.string.*switch\_smoke\_on*);  
 message = "smokeON\n";  
 } else {  
 smoke.setText(R.string.*switch\_smoke\_off*);  
 message = "smokeOFF\n";  
 }  
 sendData(message);  
 }  
 });  
  
 sound.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message;  
 if (sound.isChecked()) {  
 sound.setText(R.string.*switch\_sound\_on*);  
 message = "soundON\n";  
 } else {  
 sound.setText(R.string.*switch\_sound\_off*);  
 message = "soundOFF\n";  
 }  
 sendData(message);  
 }  
 });  
  
 sendConfig.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String message = "sendConfig";  
 sendData(message);  
 }  
 });  
  
 connectBluetooth();  
  
 }  
  
 public void introSound(){  
 if(musicSound){  
 musicSound = false;  
 playPause.setImageResource(R.drawable.*ic\_play*);  
 }else{  
 musicSound = true;  
 playPause.setImageResource(R.drawable.*ic\_pause*);  
 }  
  
 if(musicSound) {  
 mediaPlayer.start();  
 }else{  
 mediaPlayer.pause();  
 }  
 }  
  
 private void connectBluetooth() {  
 try {  
 bluetoothSocket = bluetoothDevice.createRfcommSocketToServiceRecord(MY\_UUID);  
 bluetoothSocket.connect();  
 outputStream = bluetoothSocket.getOutputStream();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void sendData(String message) {  
 try {  
 outputStream.write(message.getBytes());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 try {  
 bluetoothSocket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
}

# Código Arduino

# Bibliografia

https://www.printables.com/model/146410-airsoft-arduino-bomb-replica-kajaki-airsoft

<https://github.com/yinbot/Airsoft-BombPro>

https://github.com/JorgeFilipePinto/Airsoft\_Bomb\_Montain\_Wolves\_Team.git