

MANUAL DO DISPOSITIVO DE GEOLOCALIZAÇÃO PARA AUXÍLIO DE IDOSOS COM PROBLEMA DE MEMÓRIA

Histórico de Revisões

Data	Versão	Descrição	Autor
26/11/2018	1.0	Criação do manual	Gabriel Tutia Leonardo Centenaro Ramos
xx/12/2018	1.1	Mudanças recomendadas	Gabriel Tutia Leonardo Centenaro Ramos

Índice

1. OBJETIVO	4
2. ARQUIVOS E LINKS	4
3. ESTRUTURA DO PROJETO	4
3.1. INSTALAR O ESP32 PARA ARDUINO.....	4
3.2. INSTALAR O CÓDIGO DO GITHUB PARA ESP32	8
4. COMPLEMENTO	10
5. RODAR O PROJETO	10
5.1. LIGAR O ESP32 E SIM808.....	10
5.2. APLICATIVO DO USUÁRIO.....	18
5.3. APLICATIVO DE MONITORAMENTO	23

1. OBJETIVO

Este documento é um manual para reprodução do projeto: dispositivo de auxílio a idosos com problema de memória. A seguir se encontram os links para baixar os arquivos necessários, bem como os programas necessários para a reprodução dos projetos. Na seção seguinte está o passo a passo para a instalação de todos os sistemas do projeto.

2. ARQUIVOS E LINKS

Se for necessário a instalação do código no microcontrolador ESP32, a última versão do programa deve ser baixada no GitHub <https://git.io/fpcXW>. Também será utilizado o IDE do Arduino, que pode ser baixado no link: <https://www.arduino.cc/en/Main/Software>. Para os aplicativos temos os dois GitHub: <https://github.com/GTutia/TCCV1> e <https://github.com/GTutia/TesteHeatMap>. No primeiro GitHub é possível baixar o aplicativo do projeto sem a função de monitoramento, em uma sessão posterior estão as instruções para a configuração da função de monitoramento. Já no segundo GitHub, está o aplicativo de mapa de calor que é uma ferramenta de visualização da função de monitoramento, que também deverá ser configurado.

3. ESTRUTURA DO PROJETO

3.1. **INSTALAR O ESP32 PARA ARDUINO**

1. Instalar a IDE do Arduino no link <https://www.arduino.cc/en/Main/Software>
2. Instalar a ferramenta Git pelo link <https://git-scm.com/> sem alterar nenhuma opção durante a instalação. Ao final, selecione a opção **Launch Git Bash** e clique em **Finish**.

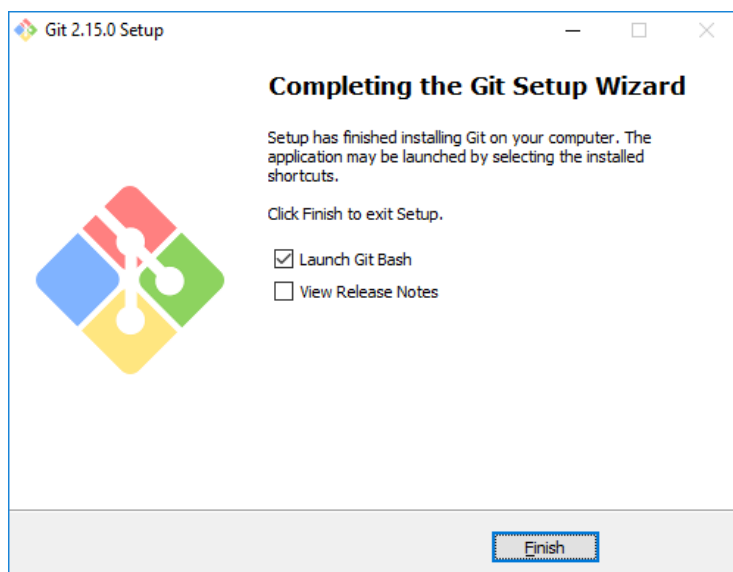


Figura 1.1 – Janela final da ferramenta Git

3. Na janela exibida automaticamente como abaixo, digite **"git gui"** e pressione ENTER



Figura 1.2 – Janela exibida e comando necessário

4. Na janela aberta Git GUI, selecione **Clone Existing Repository**:

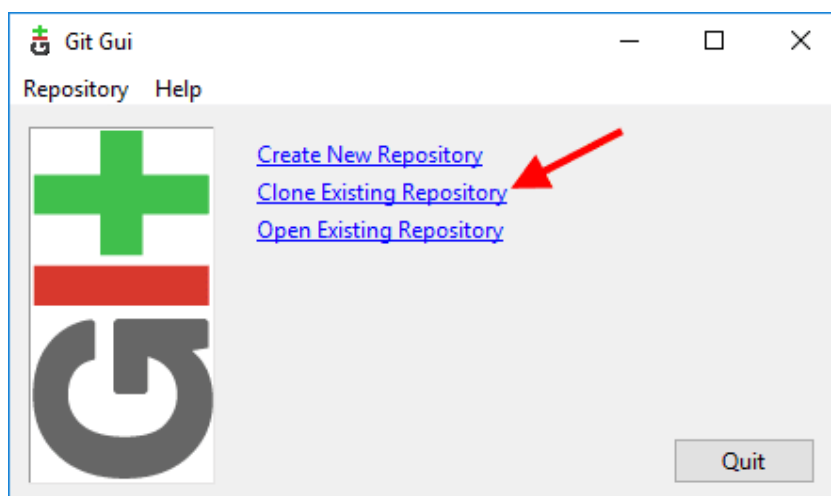


Figura 1.3 – Janela Git GUI

5. Na janela seguinte, preencha os campos **Source Location** e **Target Directory**:

Source Location: <https://github.com/espressif/arduino-esp32.git>

Target Directory é o caminho e pode variar, dependendo de como foi configurada a sua IDE, mas geralmente a pasta é C:\users\[usuario]\Documents\Arduino. Você pode encontrar essa informação dentro da IDE do Arduino, no menu **Arquivo => Preferências**:

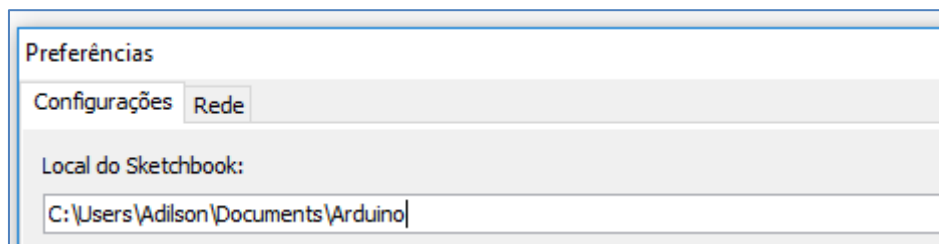


Figura 1.4 – Caminho que será utilizado no menu Preferências do IDE do Arduino

O campo **Target Directory** é formado pelo caminho mostrado em "Local do Sketchbook" mais \hardware\espressif\esp32. No exemplo, o campo ficaria:

Target Directory: C:\Users\Adilson\Documents\Arduino\hardware\espressif\esp32

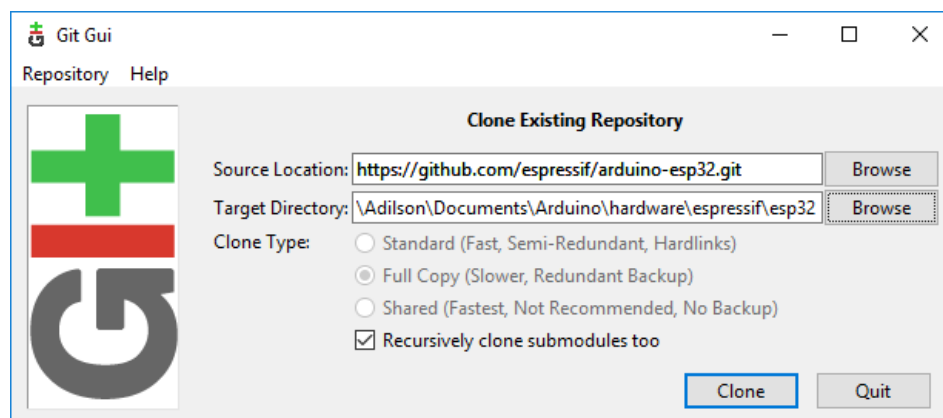


Figura 1.5 – Campos preenchidos como no exemplo

Após preencher os campos, clique em **Clone** e aguarde o download e instalação dos arquivos.

6. Vá até a pasta **[Local do Sketchbook]\hardware\espressif\esp32\tools** e execute o arquivo **get.exe**. Aguarde o download e descompactação dos arquivos:

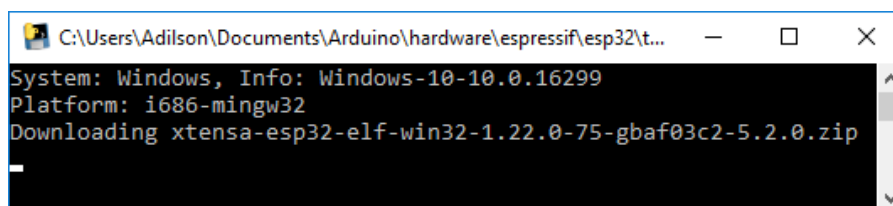


Figura 1.6 – Arquivos sendo descompactados

7. Inicie a IDE do Arduino e no menu **Ferramentas => Placa** selecione **ESP32 Dev Module**:

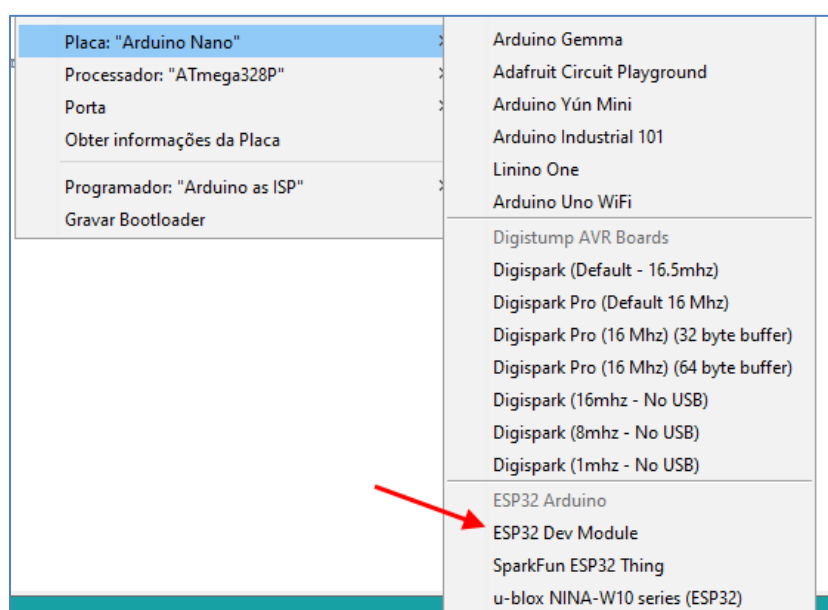


Figura 1.7 – Escolha da placa ESP32 no IDE do Arduino

3.2. INSTALAR O CÓDIGO DO GITHUB PARA ESP32

1. Acessar o GitHub pelo link <https://git.io/fpcXW>. Na página, clique para baixar os arquivos como .zip clicando em “**Clone or download**” e em seguida em “**Download ZIP**”:

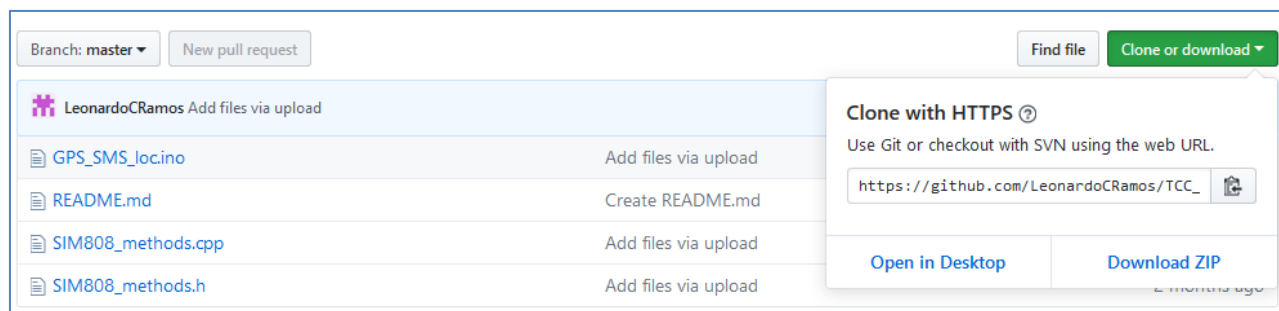


Figura 2.1 – Baixar arquivos em ZIP do código para ESP32

2. Salve o ZIP em uma pasta e extraia os arquivos, clicando em “**Extrair arquivos...**” e, em seguida, clique em **OK**.

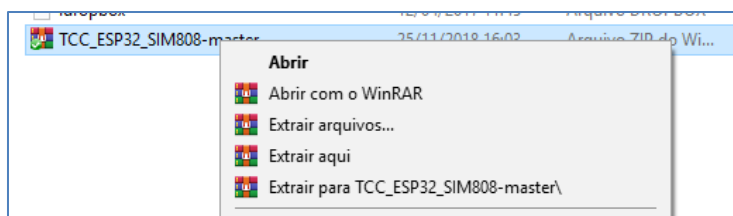


Figura 2.2 – Extrair arquivos da pasta baixada

3. Abra a pasta criada e o arquivo **GPS_SMS_loc**. O arquivo deve ser aberto pelo IDE do Arduino. Na janela que aparecer pedindo para criar uma pasta, clique em **OK**.

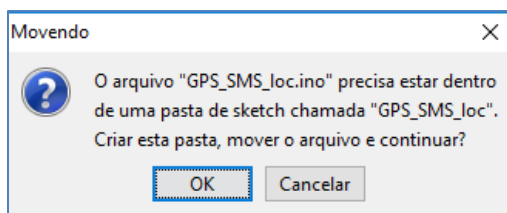


Figura 2.3 – Janela pedindo a criação de uma pasta

- Na pasta anterior, transfira os arquivos **SIM808_methods.cpp** e **SIM808_methods.h** para dentro da nova pasta criada **GPS_SMS_loc**

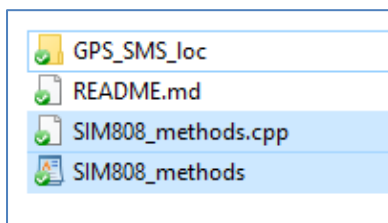


Figura 2.4 – Mover os arquivos selecionados para a nova pasta

- Na IDE do Arduino aberta, vá em **Ferramentas -> Placa** e escolha a placa **ESP32 Dev Module**

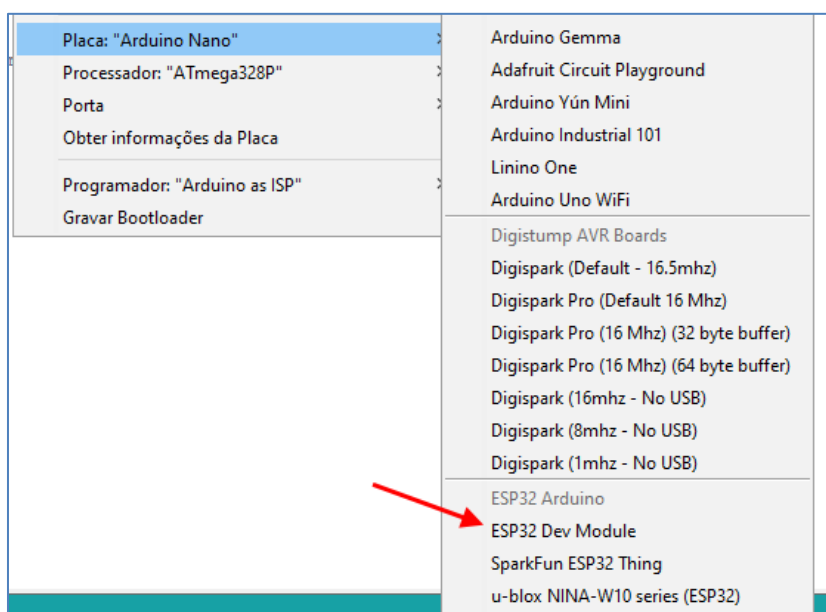


Figura 2.5 – Selecionar a placa ESP32 Dev Module

- Conecte o USB-micro no microcontrolador ESP32 e na porta USB do computador. Em **Ferramentas -> Porta**, escolha a porta COM do módulo.

7. Clique na seta para carregar o código no ESP32:

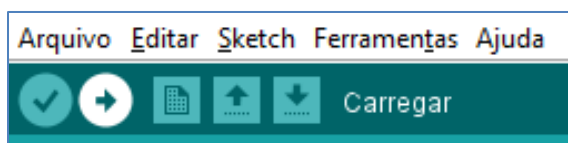


Figura 2.6 – Clicar na seta (Carregar) para fazer o upload do código no ESP32

4. COMPLEMENTO

O projeto foi implementado utilizando a versão 3.1.4 do Android Studio.

5. RODAR O PROJETO

5.1. LIGAR O ESP32 E SIM808

1. No SIM808, rosquear a antena de GPRS/GSM no pino escrito **GSM ANTE**, como mostrado abaixo:



Figura 3.1 – Antena GPRS/GSM

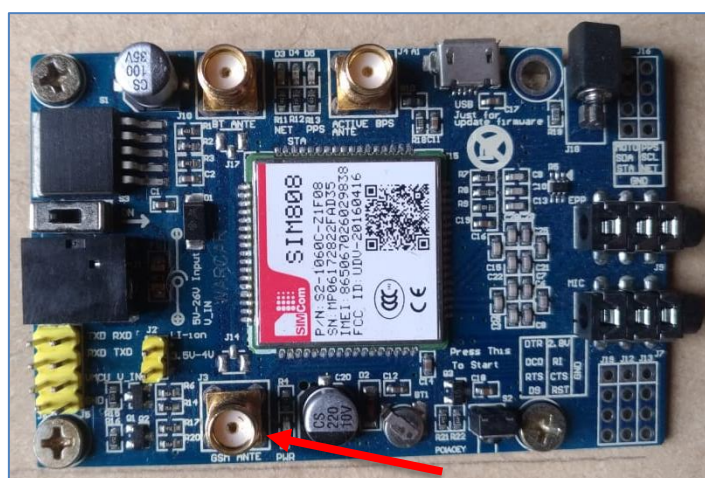


Figura 3.2 – Entrada para antena GPRS/GSM

2. Rosquear o sensor GPS no pino escrito **ACTIVE BPS ANTE** no SIM808:



Figura 3.3 – Sensor GPS



Figura 3.4 – Entrada para sensor GPS

3. As conexões devem ser como abaixo:



Figura 3.5 – Antena GPRS/GSM e GPS conectados

4. Com um cabo jumper, conectar na fileira de oito pinos no pino TXD próximo à borda e entrada da fonte do SIM808, indicado abaixo:

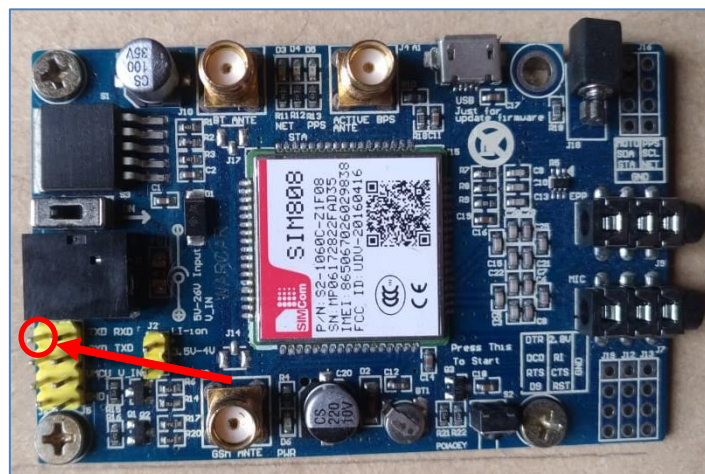


Figura 3.6 – Conexão do primeiro jumper (TXD)

Conectar a outra extremidade do cabo jumper na porta GPIO17 TX2 do ESP32, sendo ela a nona entrada a partir da entrada USB do lado direito, como indicado no esquemático abaixo:

NodeMCU-32S PINOUT

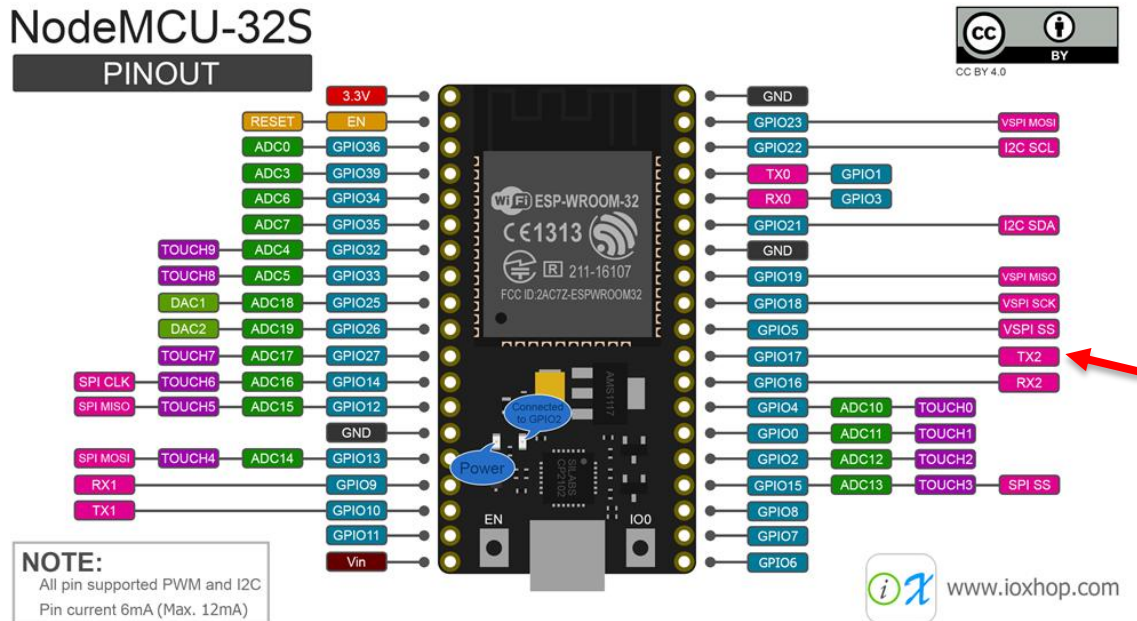


Figura 3.7 – Esquemático mostrando entrada GPIO17 (TX2)

- Com um cabo jumper, conectar na fileira de oito pinos no pino RXD ao lado do pino TXD conectado anteriormente no SIM808, indicado abaixo:



Figura 3.8 – Conexão do segundo jumper (RXD)

Conectar a outra extremidade do cabo jumper na porta GPIO16 RX2 do ESP32, sendo ela a oitava entrada a partir da entrada USB do lado direito, logo abaixo da entrada TX2 conectada anteriormente, como indicado no esquemático abaixo:

NodeMCU-32S PINOUT

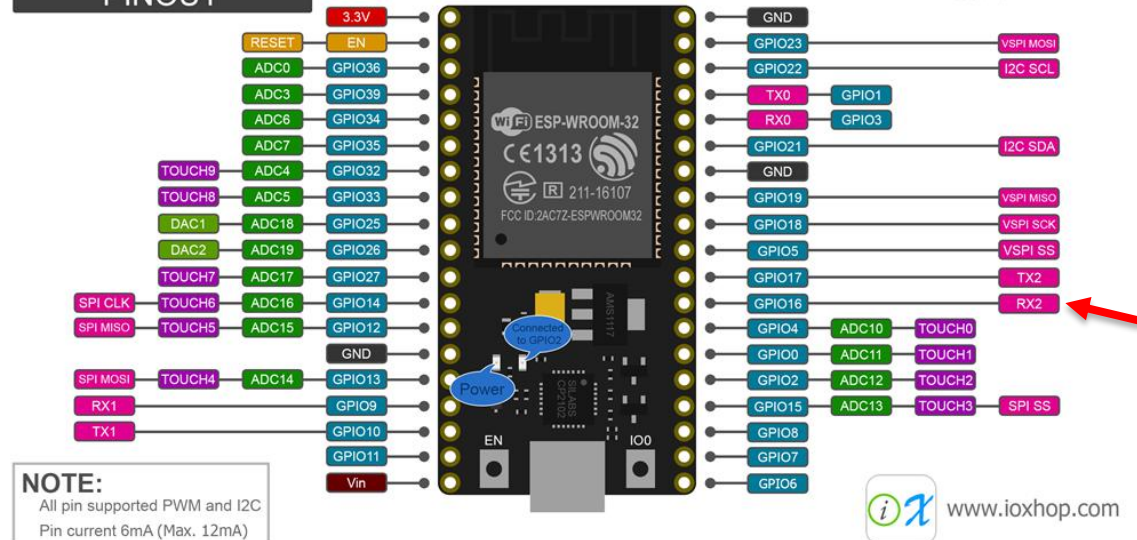


Figura 3.9 – Esquemático mostrando entrada GPIO16 (RX2)

- Com um cabo jumper, conectar na fileira de dois pinos no pino próximo à antena GPRS/GSM do SIM808, indicado abaixo:



Figura 3.10 – Conexão do terceiro jumper (terra)

Conectar a outra extremidade do cabo jumper na porta GND do ESP32, sendo ela a última entrada a partir da entrada USB do lado direito, como indicado no esquemático abaixo:

NodeMCU-32S PINOUT

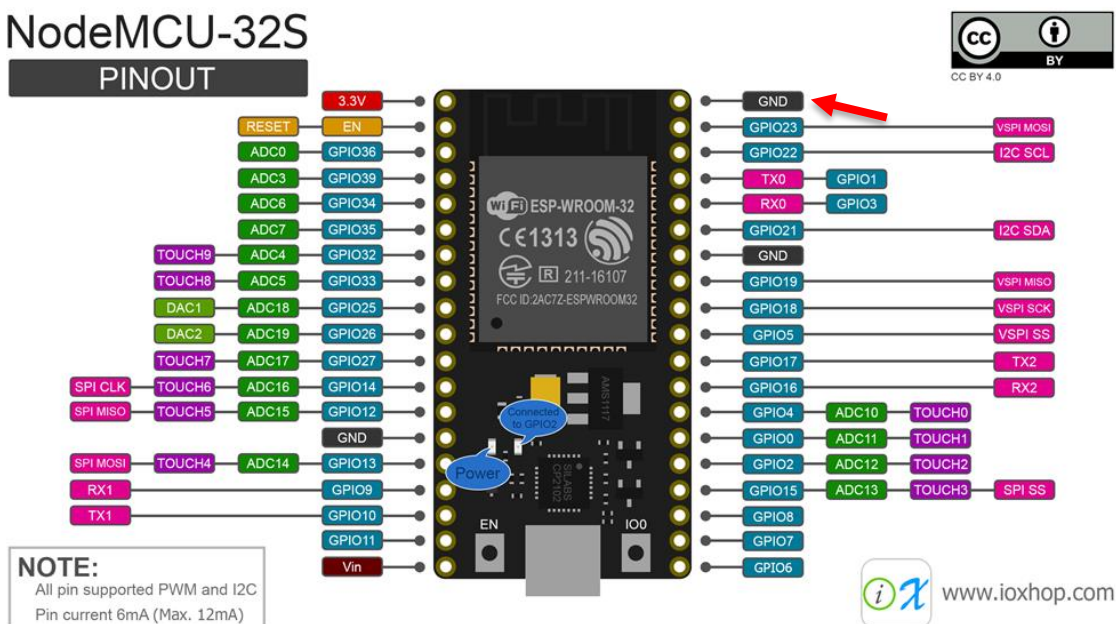


Figura 3.11 – Esquemático mostrando entrada GND

- Com um cabo jumper, conectar na fileira de dois pinos no pino mais longe da antena GPRS/GSM do SIM808, acima do pino conectado anteriormente, indicado abaixo:



Figura 3.12 – Conexão do quarto jumper (tensão)

Conectar a outra extremidade do cabo jumper na porta 3.3V do ESP32, sendo ela a última entrada a partir da entrada USB do lado esquerdo, como indicado no esquemático abaixo:

NodeMCU-32S

PINOUT

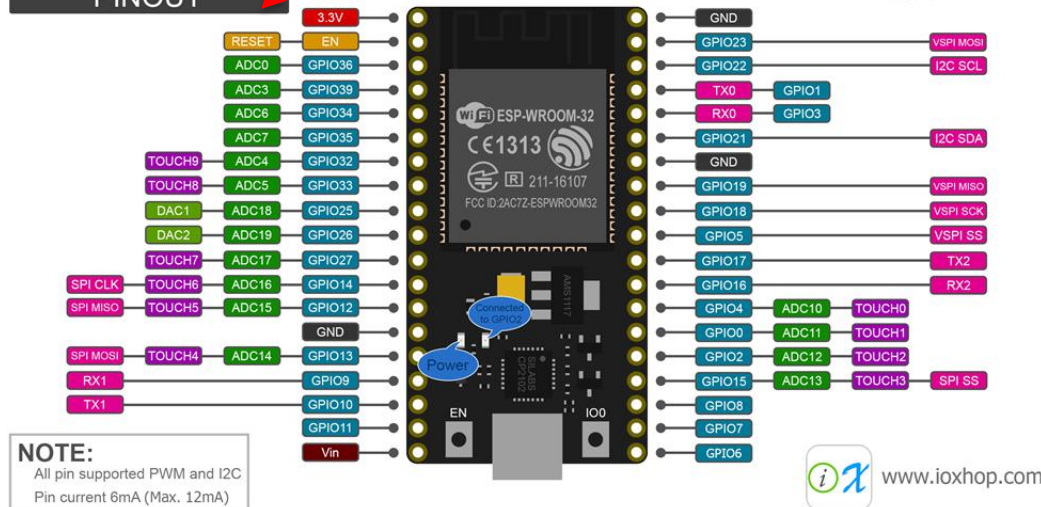


Figura 3.13 – Esquemático mostrando entrada 3.3V

- Conectar uma bateria/pilha ou uma fonte na entrada de alimentação do SIM808, indicado abaixo:



Figura 3.14 – Entrada para alimentação do SIM808

Com a energia conectada, ligue a chave de alimentação ao lado da entrada no sentido da borda para o centro do módulo, seguindo a direção da seta logo acima da chave. Se a chave foi ligada, um LED vermelho ao lado da antena GPRS/GSM deve indicar que o SIM808 está ligado, e um LED vermelho no ESP32 indica que o microcontrolador também está ligado. A chave em posição ligada é como indicada na imagem abaixo:

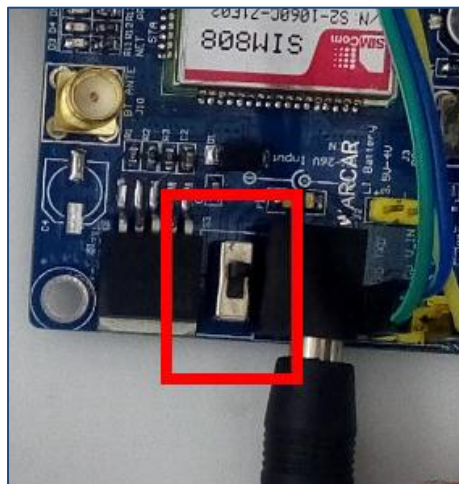


Figura 3.15 – Chave na posição ligada

9. Segurar o botão de ligar no SIM808 por 3s, indicado na imagem abaixo:



Figura 3.16 – Botão para ligar o SIM808

Se o botão foi pressionado tempo suficiente, dois LEDs vermelhos devem acender ao lado do sensor de GPS. Um deles deve ficar ligado sempre, enquanto o outro deve piscar. Para o funcionamento adequado, o LED deve estar piscando a uma frequência de aproximadamente uma piscada a cada três segundos.

10. Para desligar, basta virar a chave no sentido oposto do item 8.

5.2. APLICATIVO DO USUÁRIO

1. Baixar o apk presente em <https://github.com/GTutia/TCCV1> e instale o aplicativo.
2. Após a instalação, na primeira vez que o aplicativo for aberto serão pedidas algumas permissões necessárias ao funcionamento. Selecione a opção **"PERMITIR"** para que essas permissões sejam concedidas.

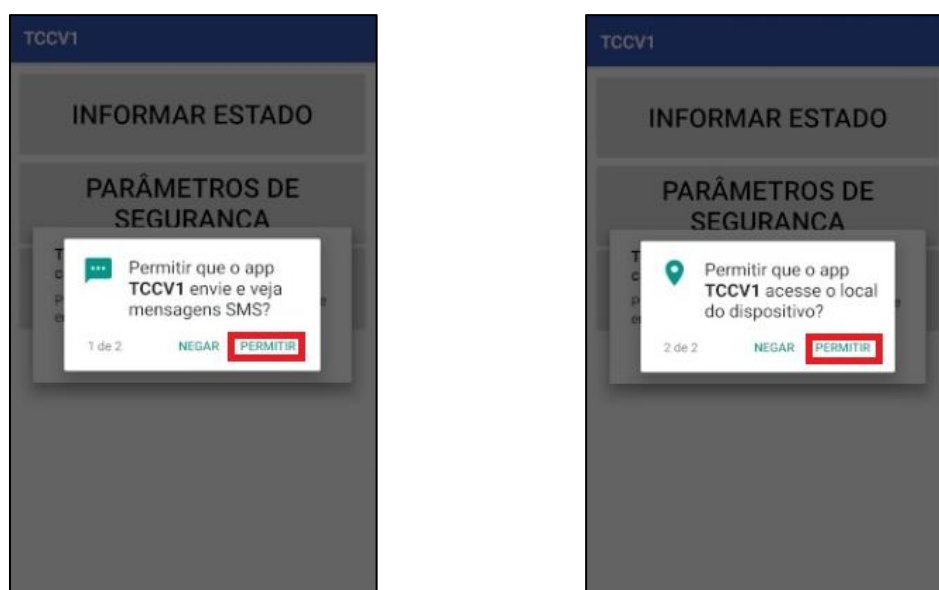


Figura 4.1 – Permissões na instalação do aplicativo

- Em seguida, uma caixa de texto aparecerá requisitando o cadastro do contato de emergência. Clique no botão “**OK**” e prossiga para a tela de cadastro.



Figura 4.2 – Notificação para cadastrar um contato de emergência

- Preencha o formulário de cadastro com o nome e o telefone do contato de segurança conforme o exemplo e clique no botão “**CADASTRAR**” para prosseguir.



Figura 4.3 – Cadastro do contato de emergência

- Após o cadastro do telefone de contato, uma segunda caixa de texto aparecerá para cadastrar o endereço da casa do paciente. Clique em “**OK**” e preencha o formulário de cadastro conforme o exemplo e clique novamente em “**CADASTRAR**”.

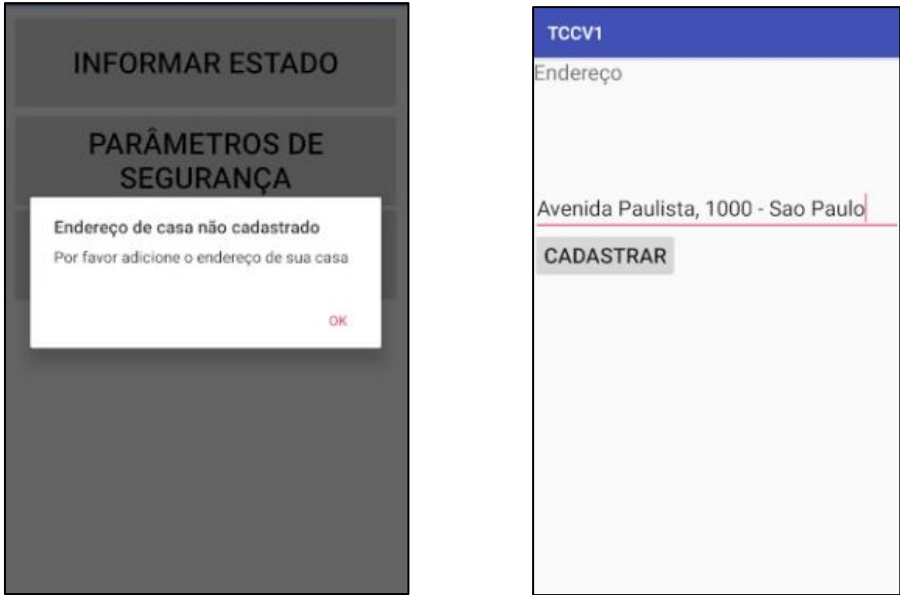


Figura 4.4 – Cadastro da casa do paciente

- Todos os parâmetros de segurança foram configurados e o aplicativo deve retornar a sua tela inicial.

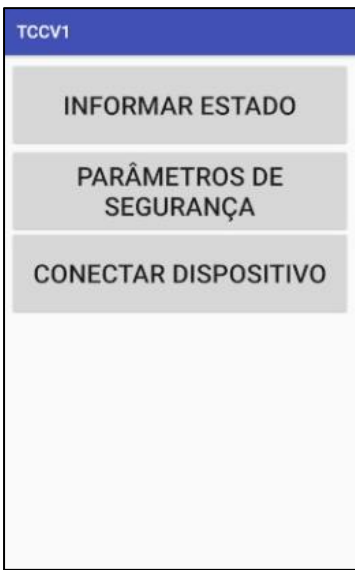


Figura 4.5 – Tela inicial do aplicativo

- Com os parâmetros de segurança configurados, é o momento de conectar o aplicativo ao dispositivo via Bluetooth. Inicialmente, ligue o Bluetooth do celular e procure um dispositivo com o nome "MyESP32". Após encontrar o dispositivo, pareie o celular com ele.



Figura 4.6 – Conectar o Bluetooth ao dispositivo MyESP32

- Com o dispositivo pareado, retorne à tela inicial do aplicativo e clique na opção "**CONECTAR DISPOSITIVO**". Aparecerá uma tela para inserir a chave correspondente ao dispositivo. Coloque no campo a seguinte chave: a49e4056-1260-4b64-a5a9-8b0ae5ba5126. Clique em "**CONECTAR**".

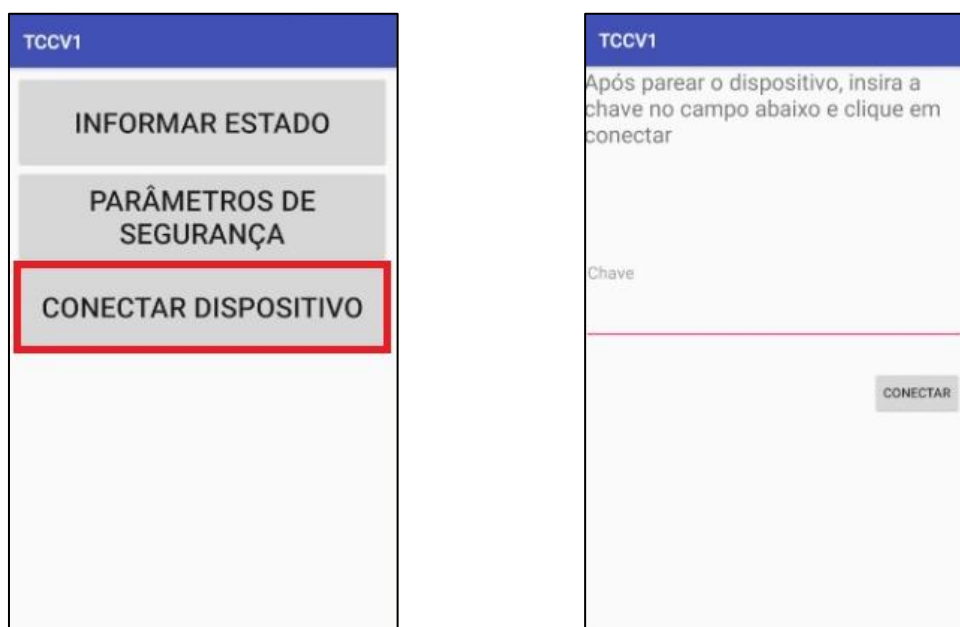


Figura 4.7 – Clicar em **Conectar dispositivo** e inserir chave

9. A partir de agora o dispositivo está conectado e o aplicativo iniciar seu monitoramento normalmente. A qualquer momento o usuário pode voluntariamente informar seu estado. Para isso, na tela inicial, clique na opção **"INFORMAR ESTADO"**. Ao escolher a opção em verde **"SIM"**, será enviado um SMS ao dispositivo informando que o usuário está bem. Caso escolha a opção em vermelho **"NÃO"**, será enviado um SMS ao contato de segurança com um pedido de socorro e a posição atual do usuário.

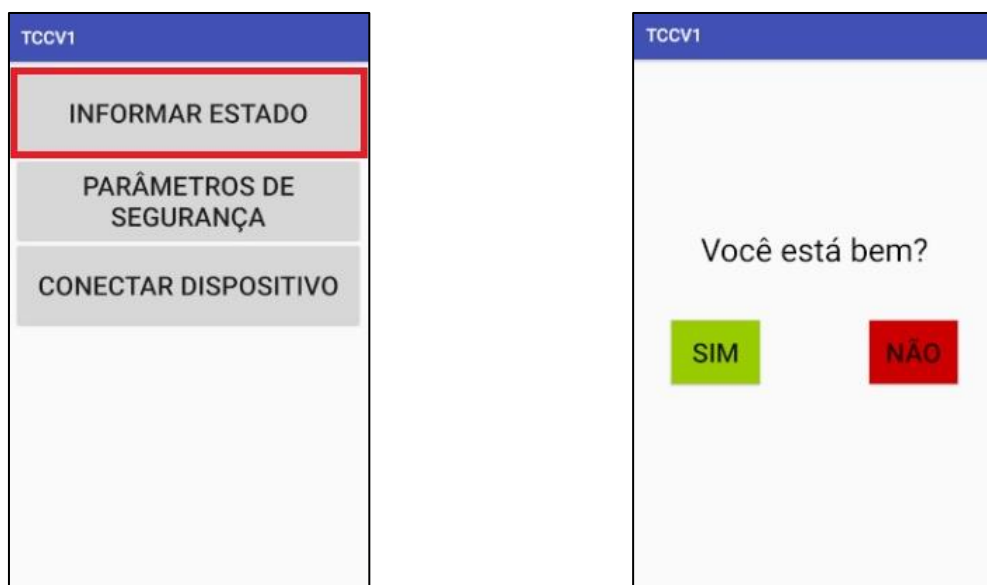


Figura 4.8 – Botão de **Informar estado** e as opções que aparecem

10. A qualquer momento também o usuário pode verificar e/ou alterar os parâmetros de segurança. Para isso basta clicar, na tela inicial, na opção “**PARÂMETROS DE SEGURANÇA**”. Então o usuário poderá visualizar os parâmetros e, se desejar alterá-los, poderá selecionar as opções “**EDITAR CONTATO DE SEGURANÇA**” e “**EDITAR ENDEREÇO DE CASA**”. **IMPORTANTE:** toda vez que o usuário alterar qualquer um desses parâmetros será preciso realizar novamente a operação de conexão com o dispositivo descrita nos itens 7 e 8.



Figura 4.9 – Botão para visualização e edição dos **Parâmetros de segurança**

5.3. APLICATIVO DE MONITORAMENTO

1. Inicialmente é preciso configurar o aplicativo inicial para habilitar a função de monitoramento. Acesse o link <https://github.com/GTutia/TCCV1> e baixe o projeto em zip. Descompacte o projeto numa pasta conhecida. Abra o Android Studio e selecione a opção “File -> Open” e abra o projeto baixado.

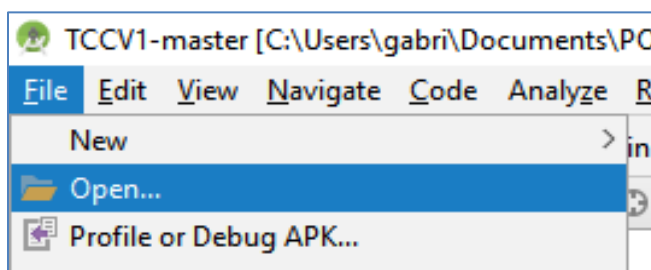


Figura 5.1 – Abrir o projeto

2. No arquivo MainActivity.java, tire os comentários das linhas logo abaixo do aviso sobre Monitoramento.

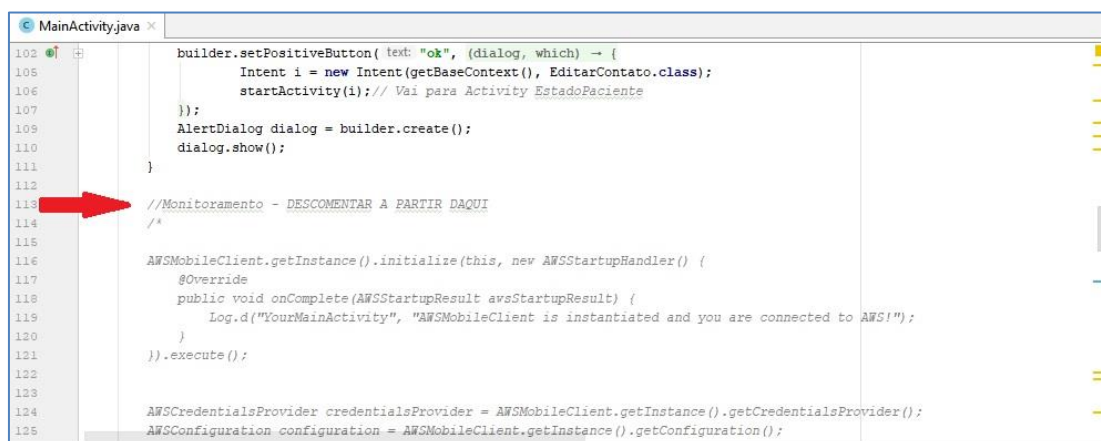


Figura 5.2 – Tirar os comentários na posição indicada

3. Crie uma conta na Amazon Web Services (<https://aws.amazon.com/pt/>).
4. Após criar a conta e estar devidamente logado, selecione o produto AWS Mobile Hub e crie um novo projeto. Nomeie o projeto da forma que preferir, selecione a plataforma Android como sendo a plataforma do aplicativo e clique em **"Add"**.

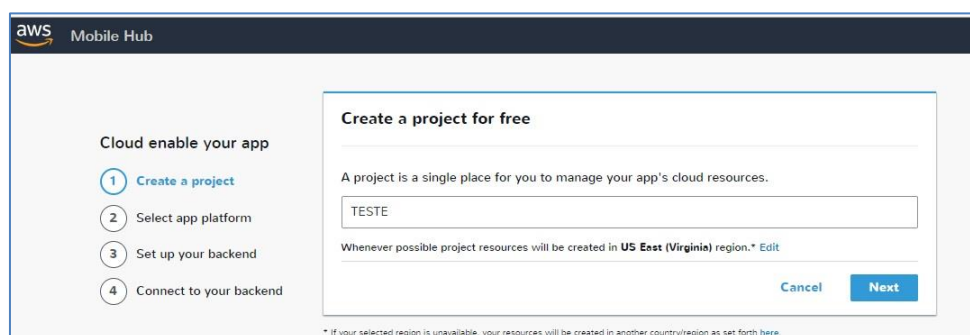


Figura 5.3 – Criar um novo projeto

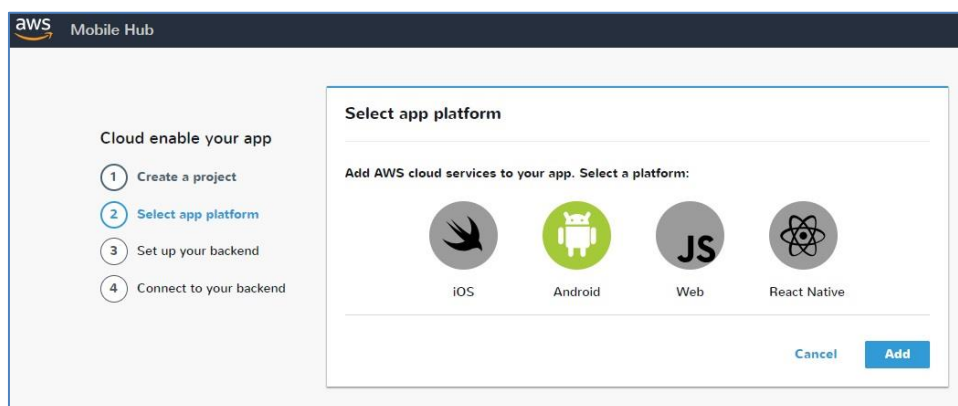


Figura 5.4 – Selecione a plataforma Android

5. Em seguida clique nos botões “**Next**” e “**Done**”. **IMPORTANTE:** Não baixe neste momento o arquivo de configuração.

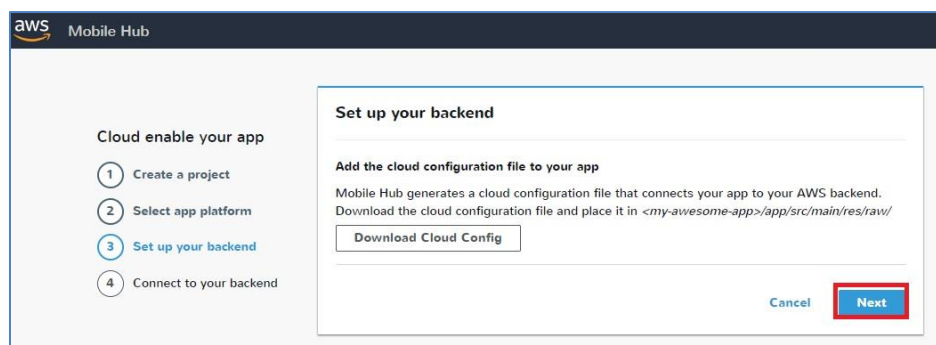


Figura 5.5 – Clicar em **Next**

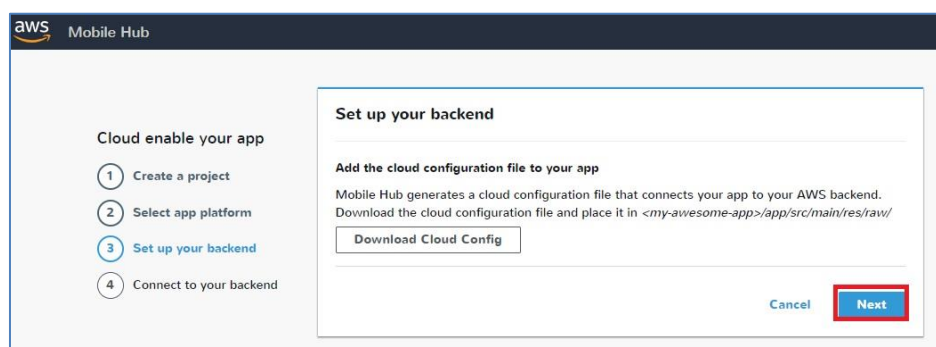


Figura 5.6 – Clicar em **Done**

6. Na página do seu projeto, selecione a opção “**NoSQL Database**” em “**Add More Backend Features**”.

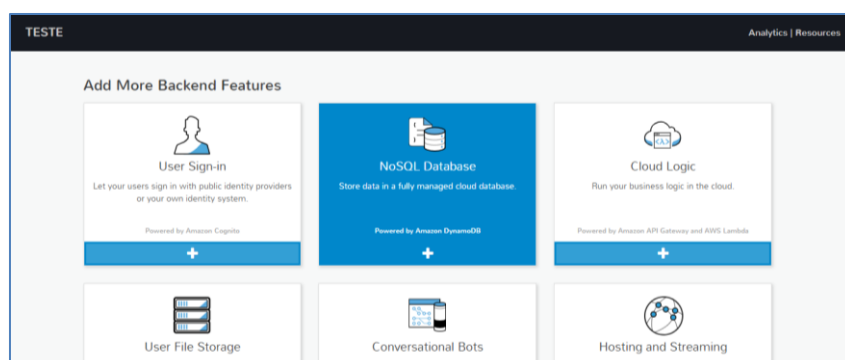


Figura 5.7 – Selecionar a opção **NoSQL Database**

7. Selecione a opção **“Enable NoSQL”** e adicione uma tabela.

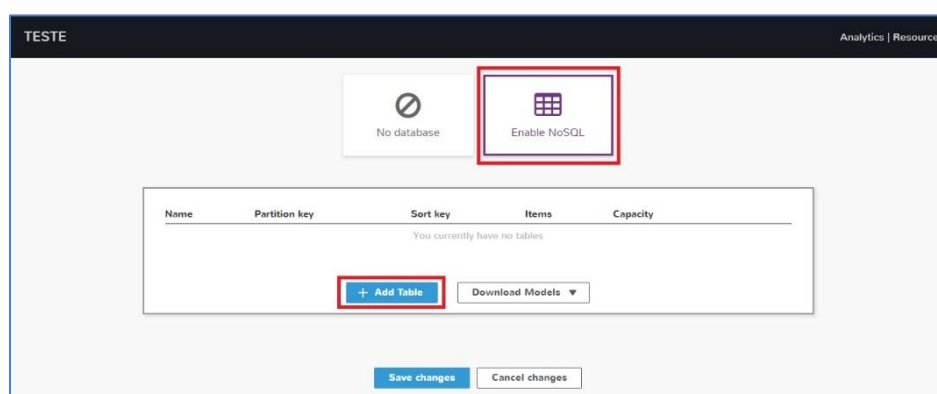


Figura 5.8 – Opção **Enable NoSQL** e **+ Add Table**

8. Escolha a opção **“Custom”** e nomeie a tabela como **“Pacientes”**.

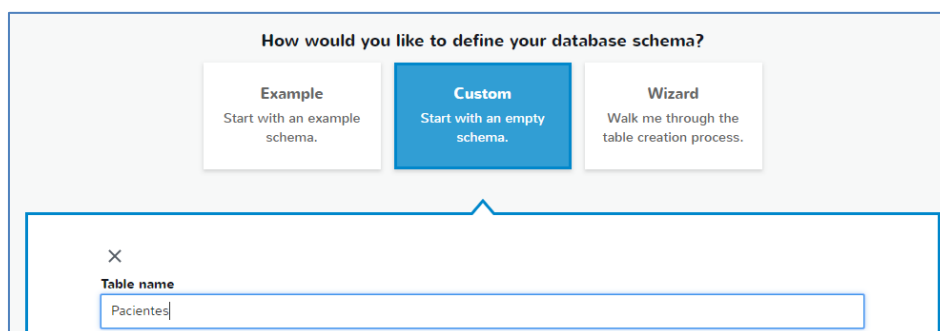


Figura 5.9 – Criar nova tabela **Pacientes**

9. Em seguida, configure os atributos para que a tabela tenha as mesmas configurações que a imagem a seguir. O primeiro atributo deve se chamar “userId” e será do tipo “string” e “Partition key”. Já o segundo atributo deverá ser “Location” e será do tipo “number” e “Sort key”. Em seguida adicione outros dois atributos do tipo “number”: “Latitude” e “Longitude”. **IMPORTANTE:** as configurações devem ser exatamente estas pois algumas funções no aplicativo foram feitas para estas configurações.

What attributes do you want on this table?

Each table has a built-in Primary Index which must have a Partition key and optionally may have a Sort key. Key attributes must be of type string, number or binary. Non-key attributes may be added here or from within your mobile app code.


Attribute name	Type	Partition key	Sort key	
userId	string	<input checked="" type="checkbox"/>	<input type="checkbox"/>	x
Location	number	<input type="checkbox"/>	<input checked="" type="checkbox"/>	x
Latitude	number	<input type="checkbox"/>	<input type="checkbox"/>	x
Longitude	number	<input type="checkbox"/>	<input type="checkbox"/>	x

[Add attribute](#)

Figura 5.10 – Configuração da tabela criada


10. Por fim clique no botão “**Create table**” para criar a tabela e confirme clicando novamente em “**Create table**” na caixa que irá aparecer. Ao final desta etapa deverá aparecer uma faixa com um aviso para atualizar o projeto. Clique no link sublinhado.

TESTE Analytics | Resources


 **NoSQL Database** Powered by Amazon DynamoDB

Store data in a fully managed cloud database.

Do you want to add a database to your app?



No database



Enable NoSQL

Name	Partition key	Sort key	Items	Capacity	
Pacientes	userId	Location	0	3 / 3	Edit Remove

Figura 5.11 – Crie a tabela

11. A seguir clique no botão “**Integrate**”.

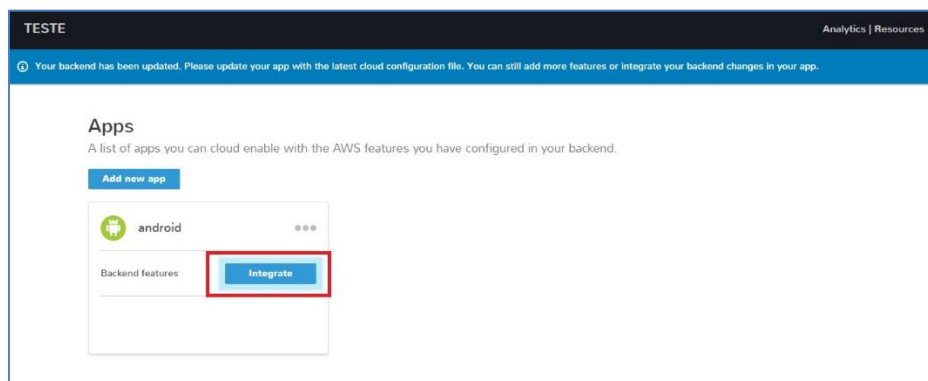


Figura 5.12 – Clique no botão **Integrate**

12. Primeiro, baixe o arquivo de configuração do AWS selecionando a opção “**Download Cloud Config**”. No Android Studio, abra a pasta “res” e copie o arquivo “awsconfiguration.json” baixado para lá. Esta pasta já deverá ter um arquivo com este mesmo nome. Selecione então “**Overwrite**”.

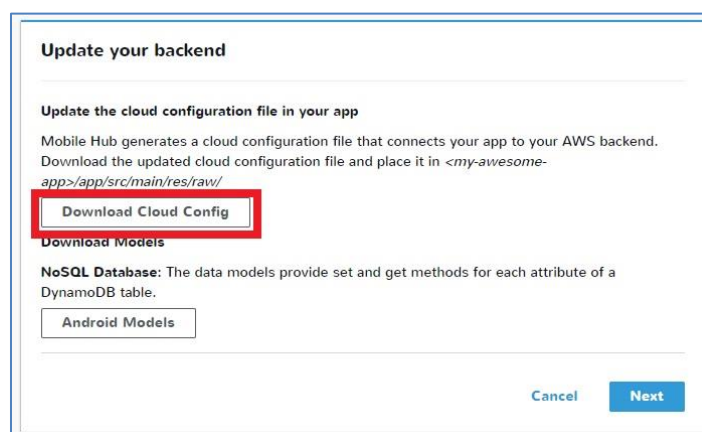


Figura 5.13 – Baixe o arquivo de configuração do AWS

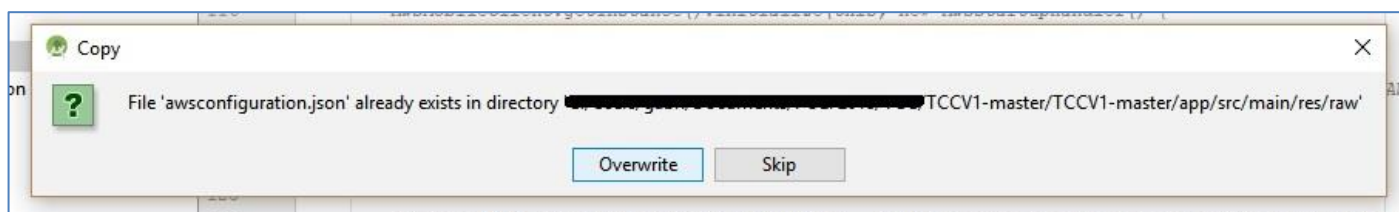


Figura 5.14 – Selecione a opção **Overwrite**

13. Em seguida, baixe os modelos de dados selecionando a opção “Android Models”. Copie o arquivo “PacientesDO.java” dentro do zip baixado e copie e cole este arquivo para o mesmo diretório do arquivo “MainActivity.java”. Assim como no item anterior, selecione a opção “**Overwrite**”.

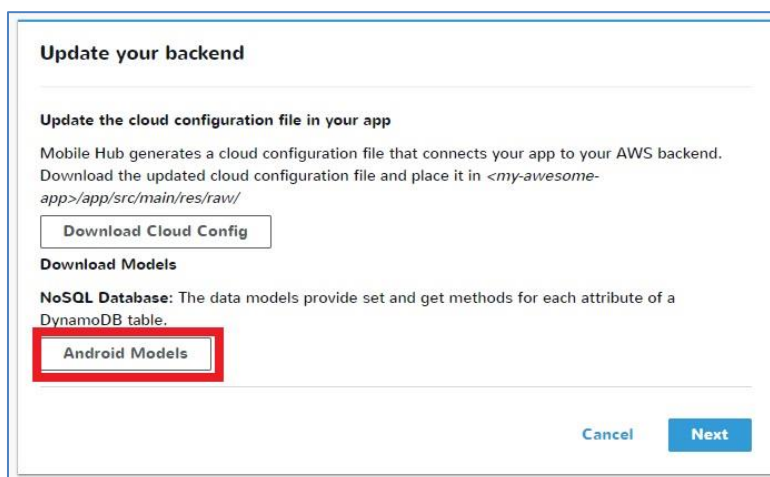


Figura 5.15 – Faça o download dos modelos de dados

14. Neste momento o aplicativo já estará configurado com a opção de monitoramento. Para instalá-lo no celular, selecione a opção Build -> Build APK(s). Após o processo terminar, acesse o arquivo apk gerado na pasta /app/build/outputs/apk/debug e baixe no seu smartphone para instalar.
15. Assim como configuramos o aplicativo do usuário para a função de monitoramento, agora iremos configurar e baixar o aplicativo de mapa de calor que é justamente uma ferramenta de visualização para os dados salvos. Para baixar o aplicativo de mapa de calor basta fazer o download em zip no seguinte GitHub: <https://github.com/GTutia/TesteHeatMap>. Em seguida, extraia o projeto para uma pasta conhecida e abra-o no Android Studio.

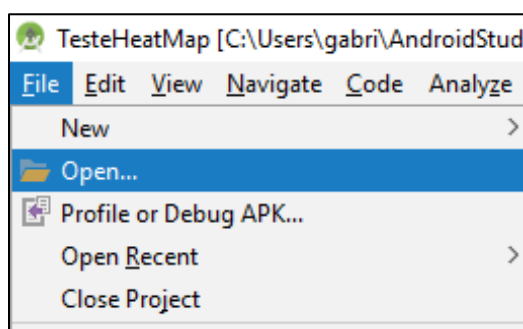


Figura 5.16 – Abra o zip baixado no Android Studio

16. Agora é necessário incluir no aplicativo naixado as mesmas configurações do projeto da Amazon Web Services criado. Assim, abra a página do seu projeto de Mobile Hub criado na AWS e clique na opção “**Integrate**”.

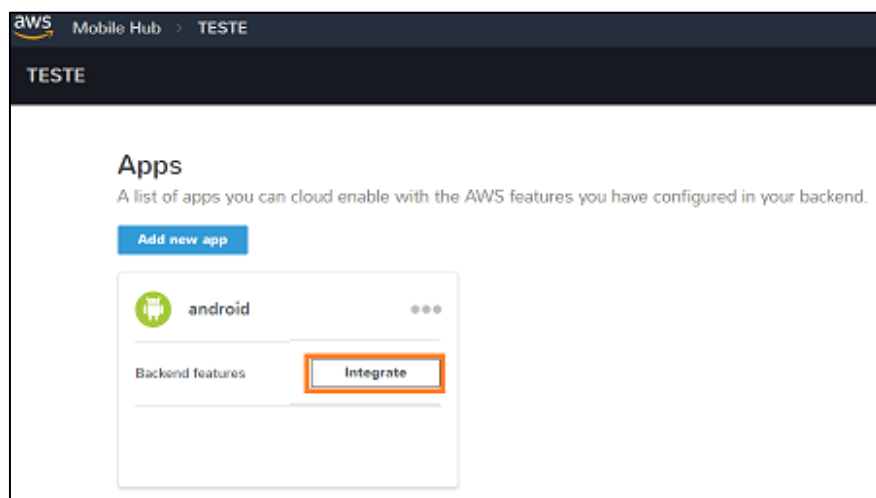


Figura 5.17 – Abra novamente o seu projeto no Mobile Hub

17. Clique então na opção **"Download Cloud Config"** e baixe o arquivo de configuração **"awsconfiguration.json"**.

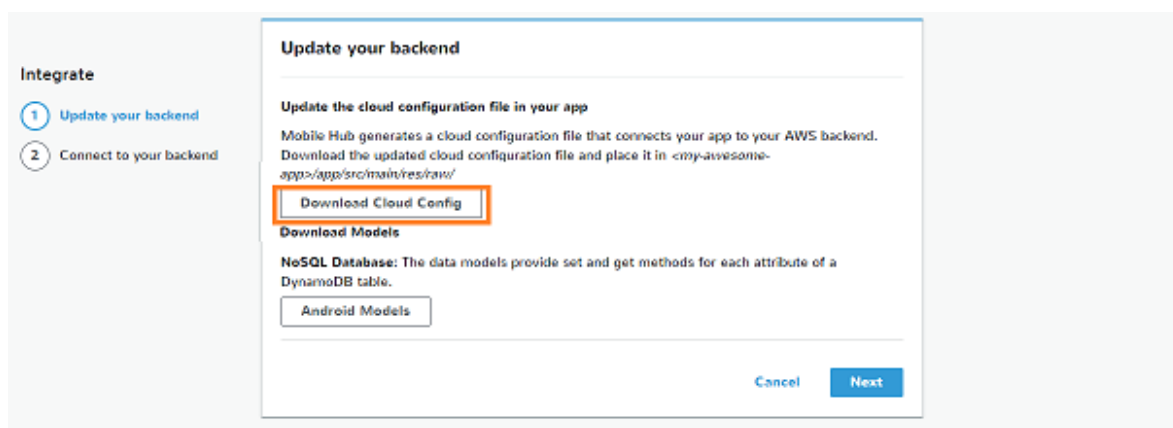


Figura 5.18 – Baixe o arquivo awsconfiguration.json

18. Extraia esse arquivo e copie ele para a pasta raw substituindo o arquivo presente pelo arquivo atual. Clique em **"Overwrite"**.

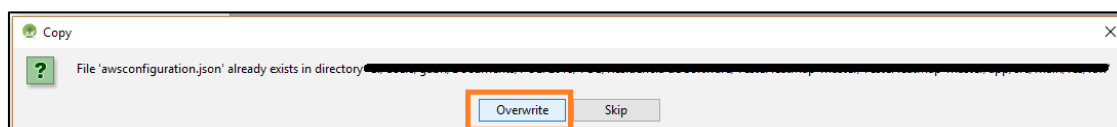


Figura 5.19 – Substitua o arquivo awsconfiguration.json na pasta raw

19. Em seguida, retorne a página de "Update your backend" e clique na opção **"Android Models"** para baixar os arquivos de estrutura de dados.

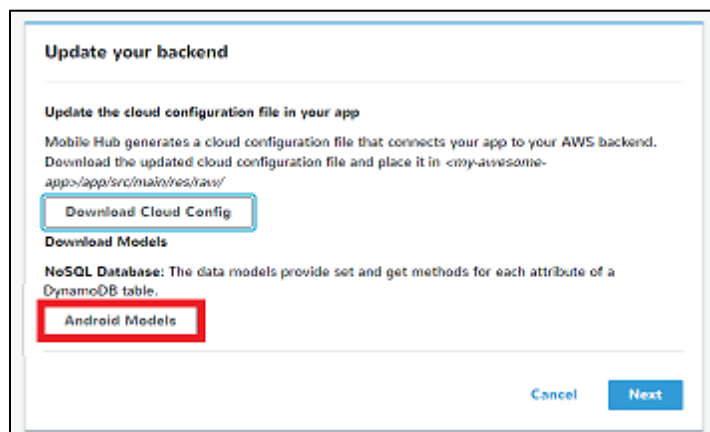


Figura 5.20 – Baixe os arquivos de estrutura de dados

20. Abra o arquivo Zip baixado e extraia o arquivo "PacientesDO.java" na pasta nosql. Copie este arquivo para a mesma pasta onde se encontra a MainActivity.java no Android Studio e substitua o arquivo presente pelo atual.

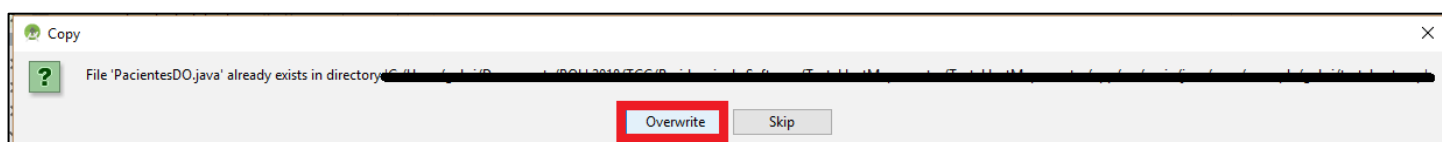


Figura 5.21 – Substitua o arquivo PacientesDO.java dentro da pasta de sua MainActivity.java

21. Agora o aplicativo de monitoramento já está configurado. Para instalá-lo no celular, selecione a opção Build -> Build APK(s). Após o processo terminar, acesse o arquivo apk gerado na pasta /app/build/outputs/apk/debug e baixe no seu smartphone para instalar.
22. O aplicativo de mapa de calor é bastante simples, basta clicar no botão esquerdo superior que ele irá atualizar a tela com os locais mais visitados salvos na base de dados.

