Introdução

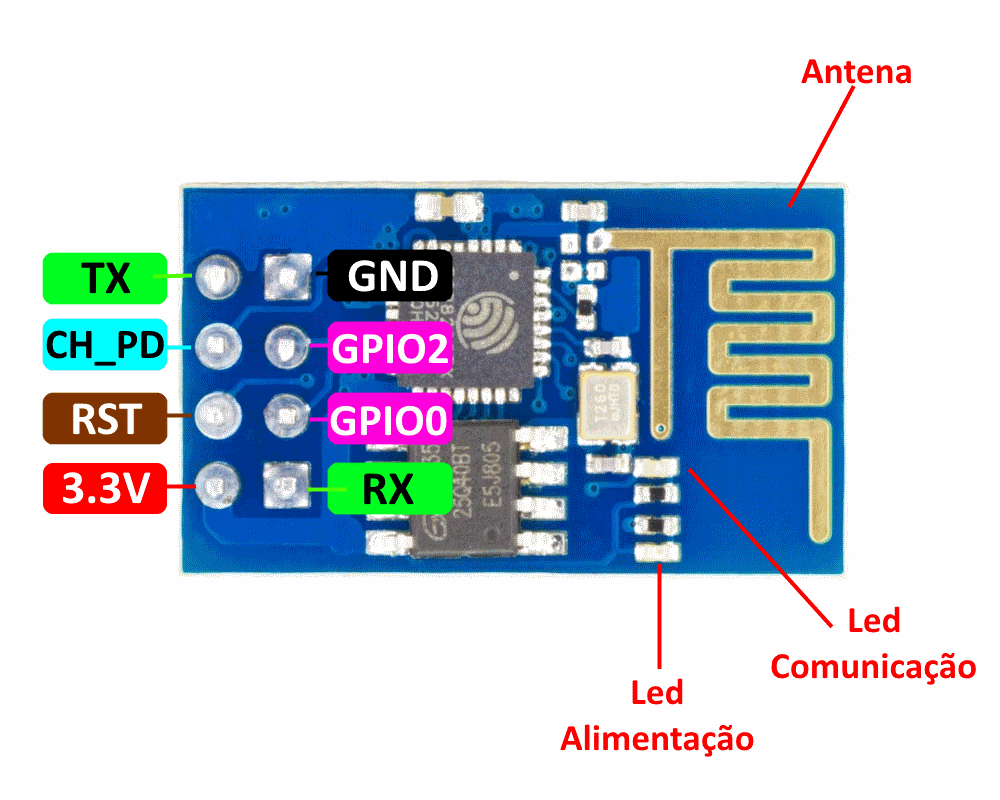
Modelo de propagação da Onda Eletromagnética

Internet das Coisas

O Módulo ESP8266

Tendo em vista as principais características das aplicações da Internet das Coisas que são conectividade e mobilidade, o módulo Wi-Fi ESP8266, da empresa Espressif, vem ganhado destaque, devido ao seu tamanho, recursos, facilidade de uso e preço acessível (menos de USD 3,00 em sites internacionais). O ESP8266 é um SOC (System On Chip) com protocolo TCP/IP integrado.

O ESP8266 foi criado em diversas variantes por sua fabricante, oficialmente são 12, numeradas de ESP-01 até ESP-12, a principal diferença está no que diz respeito ao número de IOs e tamanho do chip. A imagem a seguir mostra o ESP-01 e após e feita uma descrição dos pinos



**Vcc:** Tensão de alimentação 3.3 V, consome até 300mA;

**GND:** sinal de Terra GND;

**Tx:** Sinal do Tx do módulo a ser conectado no Rx do microcontrolador (sinal em 3.3 V);

**Rx:** sinal do Rx do módulo, a ser concectado no Tx do microcontrolador (sinal em 3.3 V!);

**RST:** sinal de Reset/Restart acionado em nível baixo (GND);

**CH\_PD:** Sinal de habilitação do chip (chip enable), usado na gravação de firmware ou atualização. Deve ser mantido em nível ALTO para operação normal;

**GPIOO:** Pode ser controlado pelo firmware, e **deve ser colocado em nível baixo (GND) para modo update, ou em nível alto para operação normal;**

**GPIO2:** I/O que pode ser controlada pelo firmware;

**LED:** Quando está ligado, fica aceso em cor vermelha, e aciona a cor Azul para indicar atividade. Pisca uma vez para indicar momento de boot.

Esse módulo possui as seguintes características de hardware:

Wi-Fi integrado em frequência de 2.4 GHz com suporte a WPA e WPA2; conectores GPIO (General Purpose Input Output), barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura; CPU que opera em 80 MHz, com possibilidade de operar em 160 MHz; Arquitetura RISC de 32 bits; 32 KBytes de RAM de intruções; 96 KBytes de RAM de dados; 64 Kbytes de ROM para boot; Memória Flash SPI Winbond W25Q40BVNIG de 512 KBytes; núcleo baseado no IP (Intellectual Processor) Diamand Standard LX3 da Tensilica, esse núcleo é baseado em um IP Xtensa da Cadence, que foi modificado a critérios da Espressif; Existem módulos de diferentes tamanhos e fabricantes.

Características de operação do ESP8266:

Conexão às redes padrão 802.11 B/G/N, porém com velocidade limitada a velocidade da porta serial;

Alcance aproximado 91 metros;

Tensão de operação: 3.3 VDC;

Comunicação serial: pinos TX e Rx

Modos de operação: Cliente, Acces Point, Cliente + Acces Point;

Modos de segurança wireless: OPEN/ WEP/WPA\_PSK/WPA\_WPA2\_PSK;

Suporta comunicação TCP e UDP, com até 5 conexões simultâneas;

Pode operar em faixas de temperatura de - 40ºC a 125ºC;

Energia de consumo em modo sleep menor que 10 uA;

Tempo para sair de sleep e transmitir pacotes menor que 2ms;

Potência de standy menor que 1.0mW;

Esse módulo pode ser utilizado em uma ampla gama de aplicações, exatamente pelo fato de possuir conectividade Wi-Fi, um poder de processamento e tamanho reduzido, possíveis aplicações para esse módulo são: Tomadas inteligentes; Automação residencial; Monitoramento remoto; Segurança doméstica, comercial e industrial; Redes de sensores; Controle industrial sem-fio; Monitores de bebês e crianças; Eletrônica vestível; Dispositivos para localização Wi-Fi; Tags de identificação para segurança; Câmeras IP; Robótica; E muito mais.

Por padrão os módulos saem de fábrica com o firmware AT, que consiste em uma série de comandos para o uso do módulo em si. Para o desenvolvimento desse trabalho foi utilizado o módulo ESP8266-12E, que além das características já citadas possui alimentação micro USB 5V com regulador de tensão 3.3V e vem com o firmware NodeMCU Lua, esse firmware permite programar o dispositivo em linguagem Lua, transformando-o em um microcontrolador com Wi-Fi integrado. A imagem a seguir mostra o módulo EPSP-12E.



Técnicas de Localização em Redes Wi-Fi

Existem três técnicas comumente usadas para estimar a posição nodal em redes ad hoc (redes que não possuem um nó central): AoA (*Angle of Arrival*), RSSI (*Received Signal Strength Indicator*) e TDOA (*Time Difference Of Arrival*).

Descrição do Sistema

Atualmente, pesquisas mostram que as pessoas utilizam muito mais o smartphone do que PC para acessar a internet, isso se deve principalmente à mobilidade, permitindo que uma pessoa com smartphone esteja conectada à rede de praticamente qualquer lugar do planeta. Tendo isso em vista, o sistema proposto foi desenvolvido como uma aplicação para smartphones com sistema Android, visto que o sistema para dispositivos móveis da Google detém a maior fatia de mercado, o que não impede que no futuro essa solução também possa ser implementada para outros sistemas tais como o iOS, ou permitir que as informações de localização possam ser acessadas em um navegador por meio de um computador. O aplicativo desenvolvido também tira proveito de outros recursos que qualquer smartphone possui hoje em dia, tais como GPS, módulo Wi-Fi, funções como mostrar redes Wi-Fi disponíveis, mandar e receber informações através de redes móveis, etc.

A função de localização foi implementada tendo como base a técnica de localização RSSI, assim, a distância entre emissor (ESP8266) e receptor (smartphone) é estimada através da potência do sinal recebido.

A distância entre emissor e receptor é calculada por meio da fórmula de Friis, que relaciona potência recebida com potência transmitida e é dada pela equação a seguir:

Onde:

Pr = Potência recebida;

Pt = Potência transmitida;

Gt = Ganho antena transmissor;

Gr = Ganho antena receptor;

λ = Comprimento de onda;

R = distância entre as antenas;

A Pr é dada pela potência do sinal recebido pelo smartphone; Pt é dada pela potência de transmissão do ESP8266 (20 dBm = 100 mW); os ganhos Gt e Gr foram definidos como 1; λ foi obtido para a frequência de operação da Wi-Fi (2.4 GHz) e R é o valor da distância a ser calculado pelo aplicativo.

O diagrama de interação da figura a seguir mostra como funciona a função de localização do módulo.



Figura Fluxo de execução da função de localização.

Não é possível estimar a posição exata do módulo se encontra, mas sim a posição do GPS do smartphone e a distância aproximada em que o animal se encontra. Assim, o aplicativo irá mostrar ao dono a marcação de uma circunferência no mapa, com centro nas coordenadas do GPS que identificou o módulo e raio igual a distância estimada.

Logo, caso o dono de um animal deseje saber em que local seu animal foi avistado pela última vez, basta selecionar a opção no aplicativo que busca a última informação salva no servidor na nuvem.

Desenvolvimento do Aplicativo Android

O aplicativo foi implementado em linguagem Java, utilizando o ambiente de desenvolvimento integrado (IDE) Android Studio, cujo software, documentação e suporte é fornecido pela Google. Dentre os recursos fornecidos pelo Android Studio, destacam-se: editor, emulador com vários recursos, compilador, ferramenta para debug e integração com diversas APIs tais como a do Google Maps, que foi utilizada nessa aplicação. A imagem a seguir apresenta a interface do usuário do Android Studio.

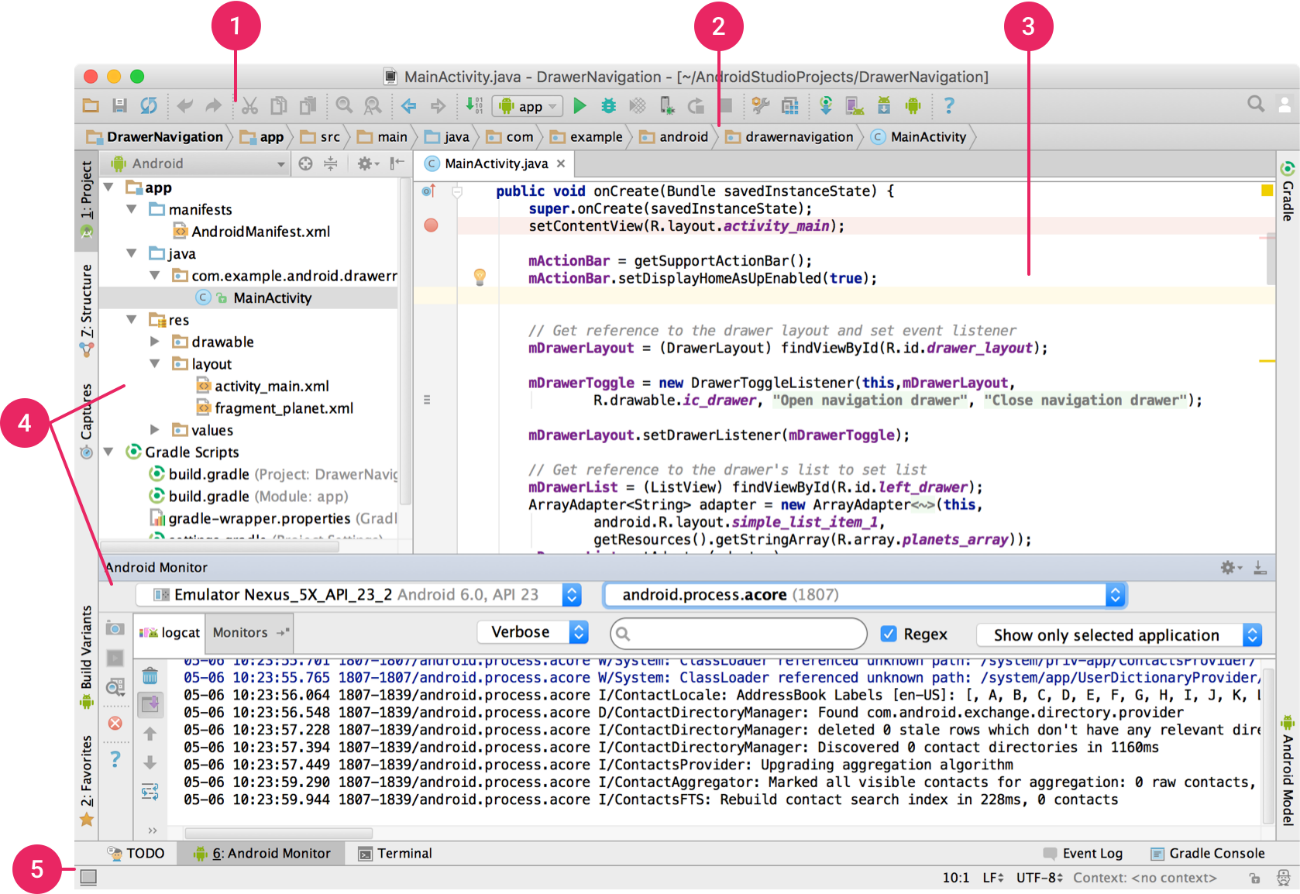


Figura xx. A janela principal do Android Studio.

1 – Barra de ferramentas: permite executar uma grande variedade de ações, incluindo a execução de aplicativos e a inicialização de ferramentas do Android;

2 – Barra de navegação: ajuda a navegar pelo projeto e a abrir arquivos para edição;

3 – Janela do editor: é o local onde é criado e modificado o código;

4 – Janelas de ferramentas: permitem acessar tarefas específicas como gerenciamento de projetos, pesquisa e controles de versões, entre outras;

5 – Barra de status: exibe o status do projeto e do próprio IDE, bem como todos os avisos e mensagens.

APIs utilizadas

Foram utilizadas duas APIs (Application Programming Interface) no desenvolvimento do aplicativo: a do Google Maps e a do Firebase. É necessário realizar uma série de passos para integrá-las ao Android Studio e utilizá-las na aplicação, mas é tudo muito bem explicado na documentação de cada uma.

Cada uma delas é descrita nos tópicos a seguir.

- Google Maps

O Google Maps é um serviço de visualização de mapas e imagens de satélite fornecido pela Google. A API para Android permite adicionar mapas baseados em dados da Google à aplicação.

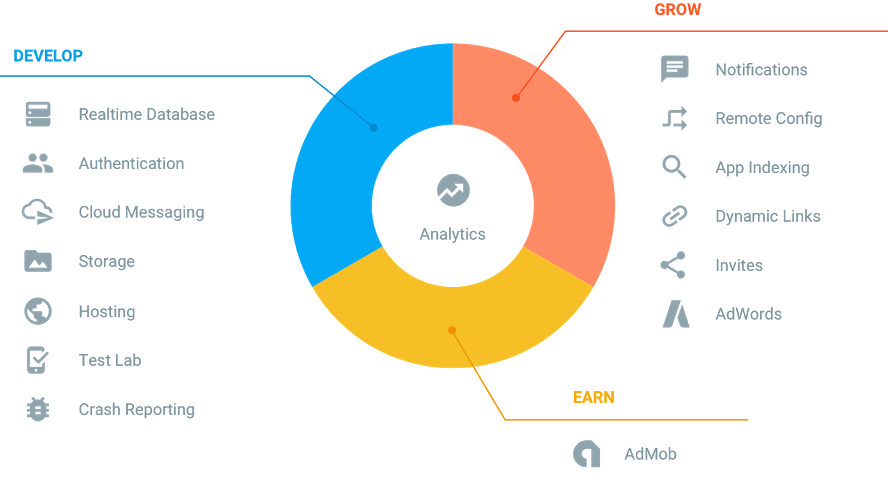
- Firebase

O Firebase fornece um conjunto de ferramentas para desenvolver aplicações de alta qualidade, de forma rápida e fácil.

A maioria dos recursos do Firebase é gratuito em qualquer escala, todos os recursos pagos têm uma camada gratuita com dois planos pagos para quando o negócio começar a expandir.

A medida que a popularidade da aplicação for crescendo não é necessário se preocupar em expandir o código do servidor ou provisionar mais capacidade, esse serviço é fornecido pelo Firebase.

A imagem a seguir mostra todos os recursos fornecidos pelo Firebase, cada recurso funciona de forma independente, cabe ao desenvolvedor decidir quais recursos integrar ou não à aplicação.



Na aplicação foi o Realtime Database, que é um banco de dados hospedado na nuvem. Os dados são armazenados em JSON e sincronizados em tempo real com todos os clientes conectados e permanecem disponíveis quando o aplicativo é desconectado. Assim, todos os clientes compartilham uma instância de Realtime Database e automaticamente recebem atualizações com os dados mais recentes.

~~As principais vantagens do Realtime Database são: resposta em tempo real – Em vez de solicitações HTTP típicas, o Firebase Realtime Database usa a sincronização de dados – sempre que os dados são alterados, qualquer dispositivo conectado recebe essa atualização em milissegundos; Off-line - Aplicativos do Firebase permanecem responsivos mesmo off-line, pois o Firebase Realtime Database SDK mantém seus dados no disco. Quando a conectividade é restabelecida, o cliente recebe as alterações perdidas, sincronizando-se ao estado atual do servidor; acessível de dispositivos clientes – A segurança e validação dos dados estão disponíveis por meio de regras de segurança do Firebase Realtime Database, regras baseadas em expressões que são executadas quando dados são lidos ou gravados.~~

O banco de dados no servidor do Firebase tem a estrutura de uma árvore JSON hospedada na nuvem. Diferentemente de um banco de dados SQL, não há tabelas nem registros. Quando dados são adicionados à árvore JSON eles se tornam um nó na estrutura JSON existente.

Descrição do Aplicativo

O aplicativo foi implementado sob duas perspectivas: a do cliente, dono de um animal, que adquiriu a coleira; e a do usuário que irá utilizar o aplicativo para localizar os animais. A imagem da figura mostra a tela inicial do aplicativo, onde é possível escolher entre a função de busca e a interface do cliente.

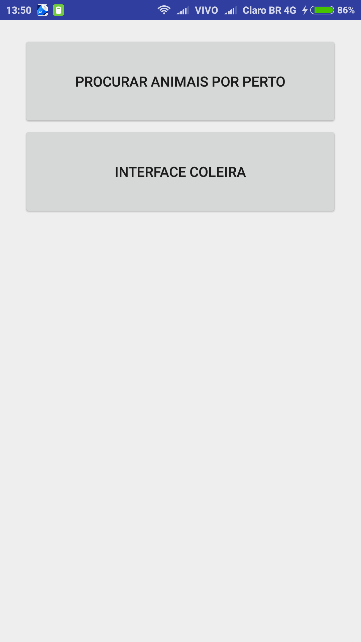
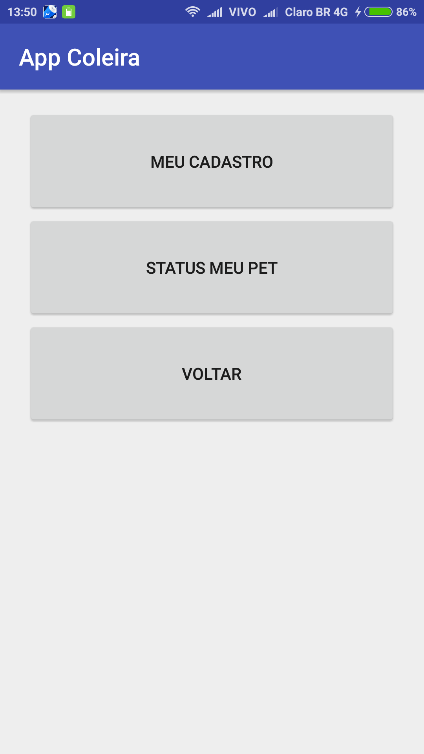


Figura . Tela inicial do aplicativo.

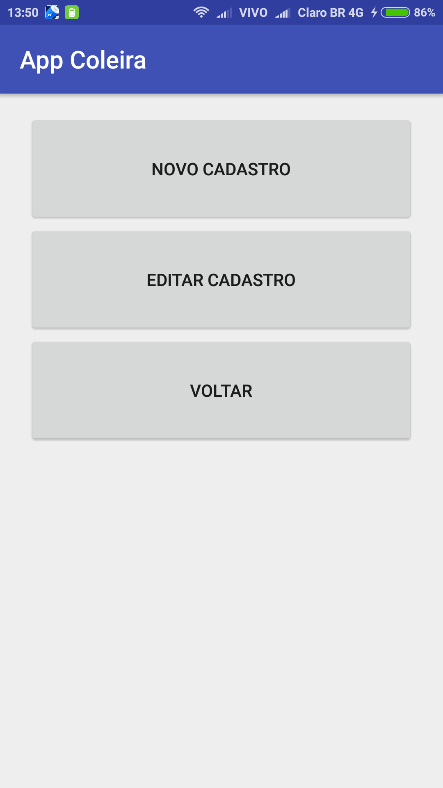
A seguir, é feita uma descrição mais detalhada de cada caso de uso, com diagramas de interação e imagens das telas do aplicativo.

Cliente

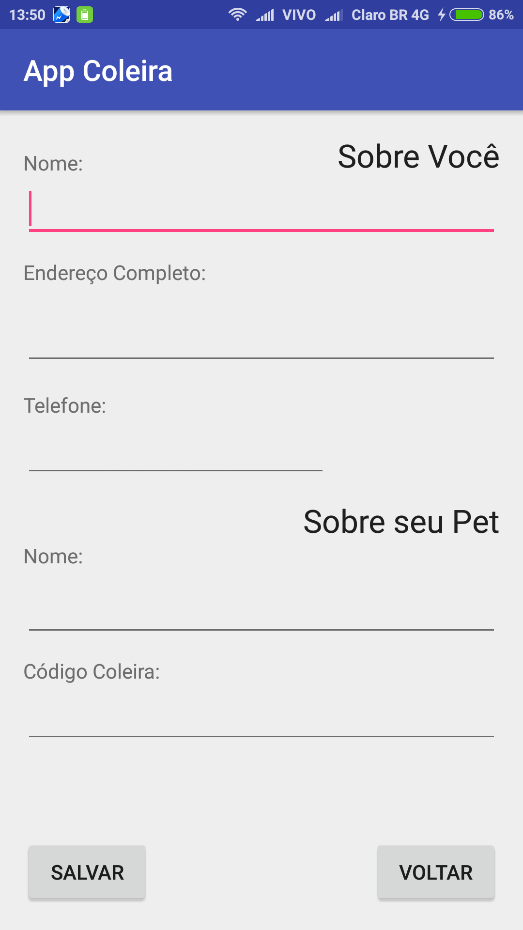
Ao entrar na interface do cliente, são exibidas opções de cadastro e de visualização de status do pet.

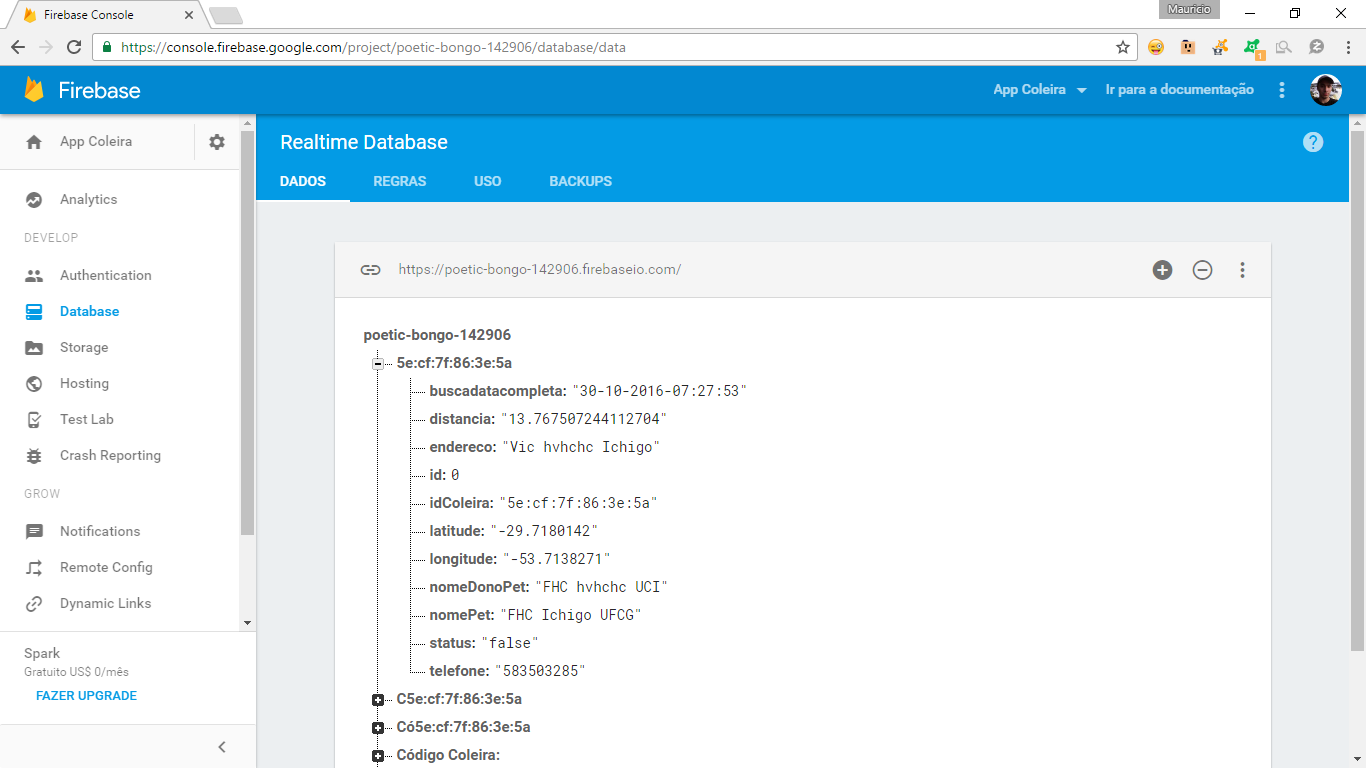


Ao selecionar a opção Meu Cadastro, o aplicativo seguirá para a uma tela onde é possível criar um novo cadastro, ou editar o cadastro feito anteriormente.



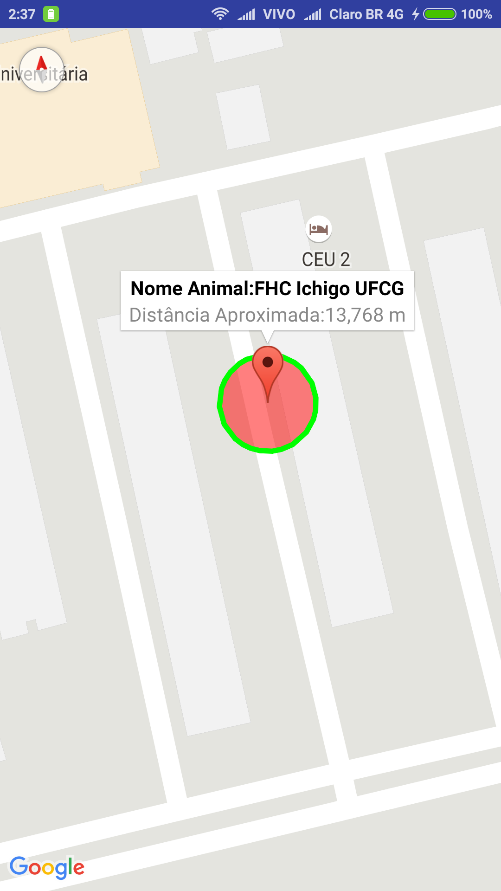
Para utilizar o sistema, primeiro é necessário que o cliente realize um cadastro. Após preencher os campos e clicar em salvar as informações são armazenadas no banco de dados do Firebase na nuvem.





Após realizar o cadastro, o cliente pode visualizar as informações cadastradas e a localização do módulo através da opção de status.





Usuário

O usuário comum do aplicativo, que também pode ser um cliente, realiza a operação de busca dos animais (módulos) que estão por perto.

Referências

http://www.embarcados.com.br/modulo-esp8266/

<https://developer.android.com/studio/intro/index.html>

<https://developers.google.com/maps/documentation/android-api/?hl=pt-br>

https://firebase.google.com/docs/database/