UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CÂMPUS CORNÉLIO PROCÓPIO

DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO (DACOM)

ENGENHARIA DE COMPUTAÇÃO

GABRIEL DIAS DADALTO

LUCCAS HENRIQUE CAYRES PIOLA

ROGER VERZOLA PERES DE LIMA

VITOR VINICIUS GOMES DA SILVA

**Parking App**

OFICINA DE INTEGRAÇÃO 1

ORIENTADOR: Francisco Pereira Junior

CORNÉLIO PROCÓPIO

2017

**1 PROBLEMAS E PREMISSAS**

A mobilidade urbana no Brasil possui diversos desafios devido, em parte, ao intensivo uso de automóveis. Uma dificuldade oriunda deste excesso de automóveis é a incapacidade de se encontrar vagas para estacionar o veículo. Dessa forma, visando facilitar esse processo, este projeto visa desenvolver um aplicativo Android que, utilizando um sensor de presença nas vagas de estacionamento, permitirá ao usuário se informar se há vagas disponíveis em uma localidade e determinar um caminho para alcançar uma vaga desejada.

**2 OBJETIVOS**

**2.1 Objetivo geral**

* Implementar um aplicativo Android para auxiliar em vagas de estacionamento.

**2.2 Objetivos Específicos**

* Localizar a posição atual do indivíduo;
* Determinar se uma vaga está ocupada ou não;
* Reservar e indicar o caminho para a vaga de estacionamento;
* Armazenar a posição na qual o veículo foi estacionado;
* Permitir a montagem de estacionamentos.

.

**3 TRABALHOS RELACIONADOS**

1. Recentemente, o Google lançou um serviço dentro de sua plataforma Google Maps que disponibiliza informações sobre vagas em regiões da cidade de São Paulo - SP e Rio de Janeiro - RJ. O serviço mostra em um mapa da região em que o usuário deseja estacionar o seu carro. Uma linha contendo faixas de cores indica a facilidade de se encontrar uma vaga na região.
2. Em 2015, a Bosh, uma multinacional de engenharia, lançou um serviço de gerenciamento de vagas de estacionamento. O sistema é capaz de reconhecer espaços livres em áreas humanas e indicar a localização destes espaços para o motorista. O sistema utiliza um sensor, que pode ser instalada em garagens ou até mesmo no asfalto, para detectar os espaços e vagas disponíveis. Informações como vagas, tipo de vaga e preço por hora são informadas para o usuário.

**4 REFERENCIAL TEÓRICO**

Para este projeto, serão utilizadas algumas tecnologias diversas para o desenvolvimento do mesmo. Para o aplicativo Android, a IDE *Visual Studio* junto com a extensão *Xamarin* serão utilizados. Essa extensão é utilizada para a programação do layout do aplicativo e também a funcionalidade ligada a cada tela, botão e elementos que estarão presentes para o usuário.

Para a confecção do site responsável pela montagem de um estacionamento, as linguagens HTML junto com a JavaScript serão utilizadas. HTML é a linguagem utilizada para o layout da página, já a linguagem JavaScript é utilizada para as funcionalidades dos elementos na tela.

Para o cadastro de usuários e estacionamentos, será utilizada uma API (Interface de programação de aplicativos) hospedada no servidor Azure, fornecido pela Microsoft.

O aplicativo do Google Maps também será utilizado no nosso app para auxiliar na rota da localização atual do usuário até a entrada do estacionamento.

O hardware do projeto será composto por um Arduino UNO e um sensor de distância infravermelho TCRT 5000. Esse sensor será utilizado para verificar se a vaga que o usuário reservou foi ocupada por ele mesmo ou por algum outro. Caso não tenha sido ele, o aplicativo deve recalcular outra vaga em tempo real.

**5 FUNCIONALIDADES**

A solução deve permitir:

1. Cadastrar usuários
2. Cadastrar estacionamentos com vagas e rotas, bem como editar um estacionamento e suas vagas e rotas
3. Ao usuário, escolher um estacionamento desejado, bem como uma vaga dentro desse estacionamento
4. Após a seleção de um estacionamento e vaga, traçar uma rota entre o usuário e o estacionamento
5. Após o usuário chegar ao estacionamento, traçar uma rota entre o usuário e a vaga
   1. Caso algum outro veículo estacione na vaga original antes do usuário, automaticamente escolher outra vaga mais próxima da vaga original no mesmo estacionamento
6. Após o usuário chegar à vaga, não permitir que outros usuários escolham a vaga ocupada

**6 METODOLOGIA**

Para alcançar os objetivos propostos, será necessário primeiro um estudo prévio sobre como utilizar as tecnologias necessárias para a realização do projeto. Após esse estudo, a divisão de tarefas entre os membros do grupo se mostra necessária. Alguns realizarão a maior parte envolvendo o software, e alguns a parte do hardware. Para a confecção do aplicativo, será necessária a realização de pesquisas em fóruns, livros e demais fontes confiáveis.

**7 RESULTADOS**

Para atender à todas as funcionalidades esperadas para este projeto, foram desenvolvidos os seguintes recursos:

**7.1 Banco de Dados SQL**

Onde ficam armazenadas todas as informações dos usuários, vagas e estacionamento. O banco de dados foi desenvolvido usando o SGBD (Sistema de Gerenciamento de Banco de Dados) Microsoft SQL Server, a partir da ferramenta de ORM (Mapeamento Objeto-Relacional) ADO.NET Entity Framework, presente na plataforma .NET. Abaixo, segue o MER (Modelo Entidade Relacionamento) do projeto:

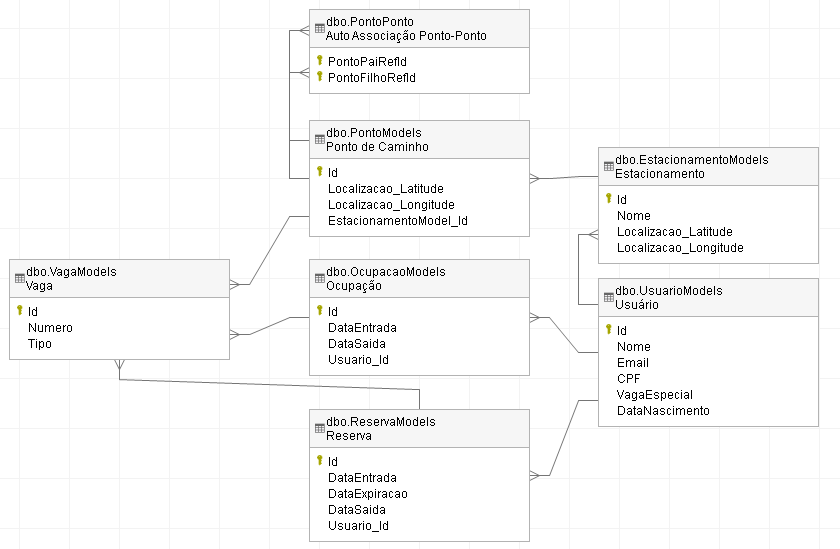


Figura 7.1.1: Modelo Entidade Relacionamento do banco de dados do projeto.

**7.2 Web API (Application Programming Interface)**

Uma Interface de Programação de Aplicativos em ambiente Web responsável por fazer a interligação entre os aplicativos Front-End e o banco de dados. Esta API foi desenvolvida usando o framework ASP.NET Web API: uma estrutura que facilita a criação de serviços HTTP compatível com uma ampla quantidade de clientes Web, tais como navegadores e dispositivos móveis. Para permitir tal compatibilidade, esta API incorpora serviços RESTful (Transferência de Estado Representacional), tecnologia que disponibiliza acesso à métodos de manipulação e obtenção de dados em um servidor Web a partir de controladores de entidades (MVC) que representam os objetos do banco de dados no formato JSON (Objeto Javascript).

Para ficar disponível na internet, a Web API foi hospedada no Serviço de Hospedagem Microsoft Azure. O Azure é plataforma destinada à execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem. A licença utilizada para consumir este serviço foi disponibilizada pela UTFPR a partir do programa Microsoft Imagine.

As principais funções desta API é atender às requisições feitas pelo programa que monitora as vagas de estacionamento através dos sensores instalados no Arduino; às requisições do Módulo Web de Edição de Estacionamento referentes à manipulação dos cadastros de vagas e plantas de estacionamentos; e, às requisições do aplicativo móvel referente a pesquisas, *logon* de usuário, trajetos, reservas e ocupações de vagas.

O funcionamento da API pode ser entendido a partir do diagrama a seguir:

**Cliente**

**(Módulo Web de Edição de Estacionamento, Aplicativo Móvel, Monitor de Vagas)**

**ASP.NET Web API**

**Entity Framework**

**Banco de Dados MS SQL**

Usuário, Vaga, Estacionamento

JSON

Figura 7.2.1 : Representação do Funcionamento da Web API

**7.3 Módulo Web de Edição de Estacionamento**

Construído na mesma estrutura que hospeda a Web API, este módulo é responsável por permitir o cadastro e manipulação de estacionamentos dos usuários. Por meio dele é possível:

* Gerenciar estacionamentos;
* Gerenciar e ligar pontos de caminhos dentro do estacionamento;
* Gerenciar vagas livres, especiais e de idosos;
* Gerenciar planta do estacionamento.

A seguir, algumas imagens de fases do Módulo Web:

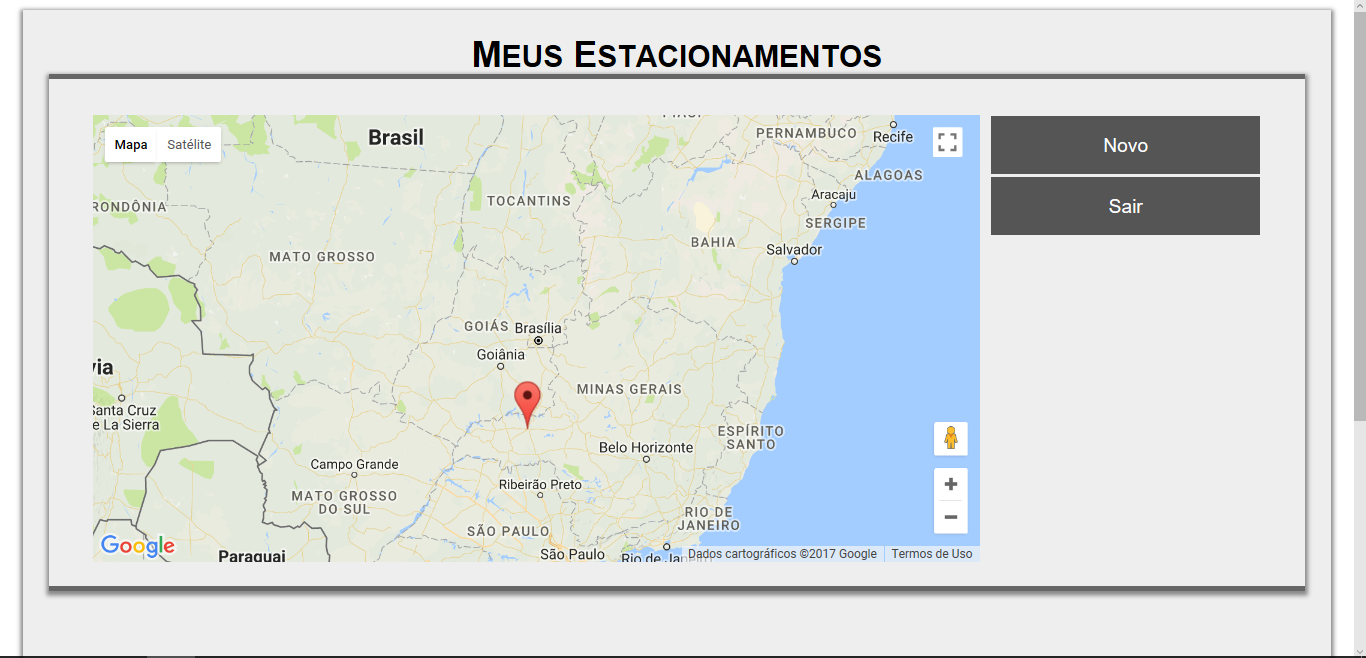


Figura 7.3.1: Tela de visualização de estacionamento do usuário. Ao clicar sobre um marcador vermelho que representa um estacionamento cadastrado, o usuário é levado a sua tela de edição.

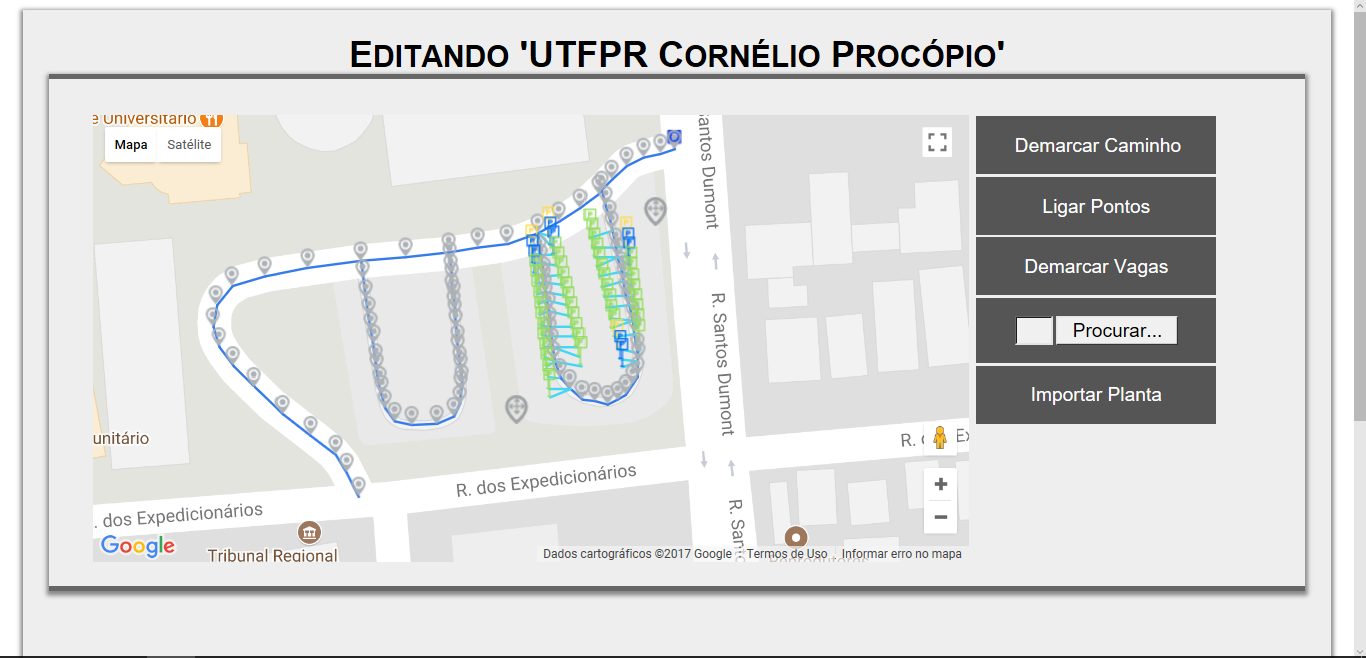


Figura 7.3.2: Tela de edição de estacionamento do usuário. Nesta tela, o usuário pode demarcar os caminhos dentro do estacionamento, inserir vagas e posicionar a planta do estacionamento. Ao clicar sobre uma vaga, (ícones verdes – Vagas Livres, azuis -Vagas De Idosos, ou amarelos – Vagas Especiais) o usuário é levado até a tela de edição da vaga.

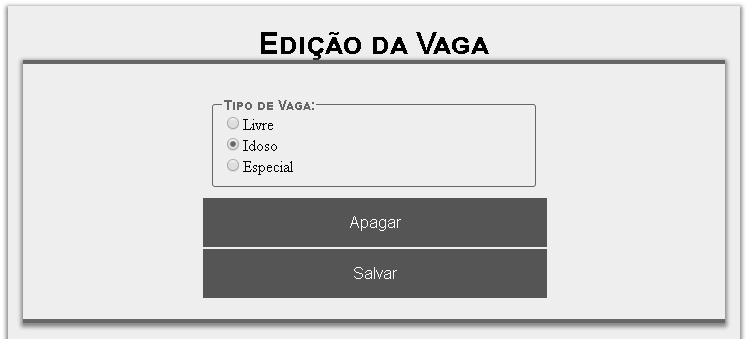


Figura 7.3.3: Tela dede Edição de Vaga. O usuário pode optar por configurar a vaga como Livre, de Idoso ou Especial. Clicando em apagar, ele exclui a vaga do estacionamento. Clicando em salvar, ele salva as alterações e volta para tela de edição do estacionamento.

**7.4 Aplicativo Móvel ParkingApp**

Aplicativoresponsávelpor permitir que usuário tenha acesso às vagas dos estacionamentos cadastrados. Por meio dele é possível pesquisar por estacionamentos, escolher vagas e obter o trajeto e navegação desde sua posição atual até a vaga desejada.

O software foi construído usando a linguagem C# para o SO Android utilizando a plataforma de desenvolvimento de aplicativos móveis para criar nativos iOS, Android e Windows chamada Xamarin. Ele faz uso da Web API para obter e manipular dados do usuário e dos estacionamentos.

As imagens a seguir representam a navegação de usuário que busca uma vaga em um estacionamento cadastrado:

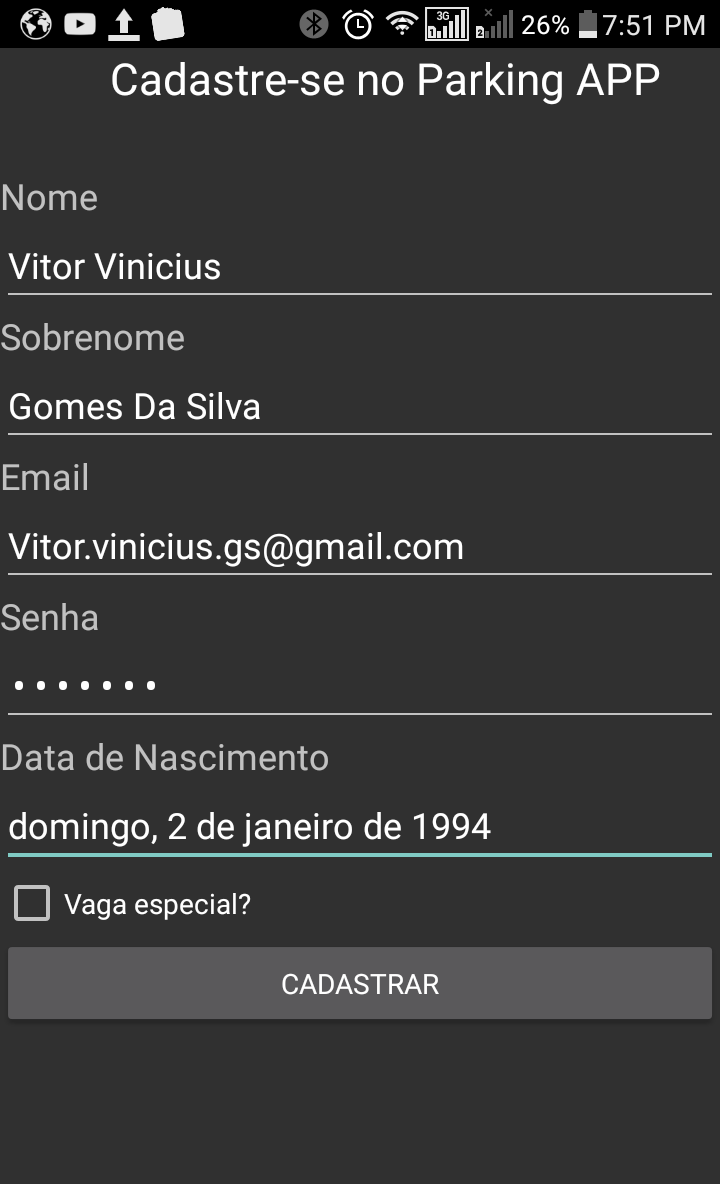


Figura 7.4.1: Tela de Cadastro do Usuário: recebe informações do perfil do usuário.

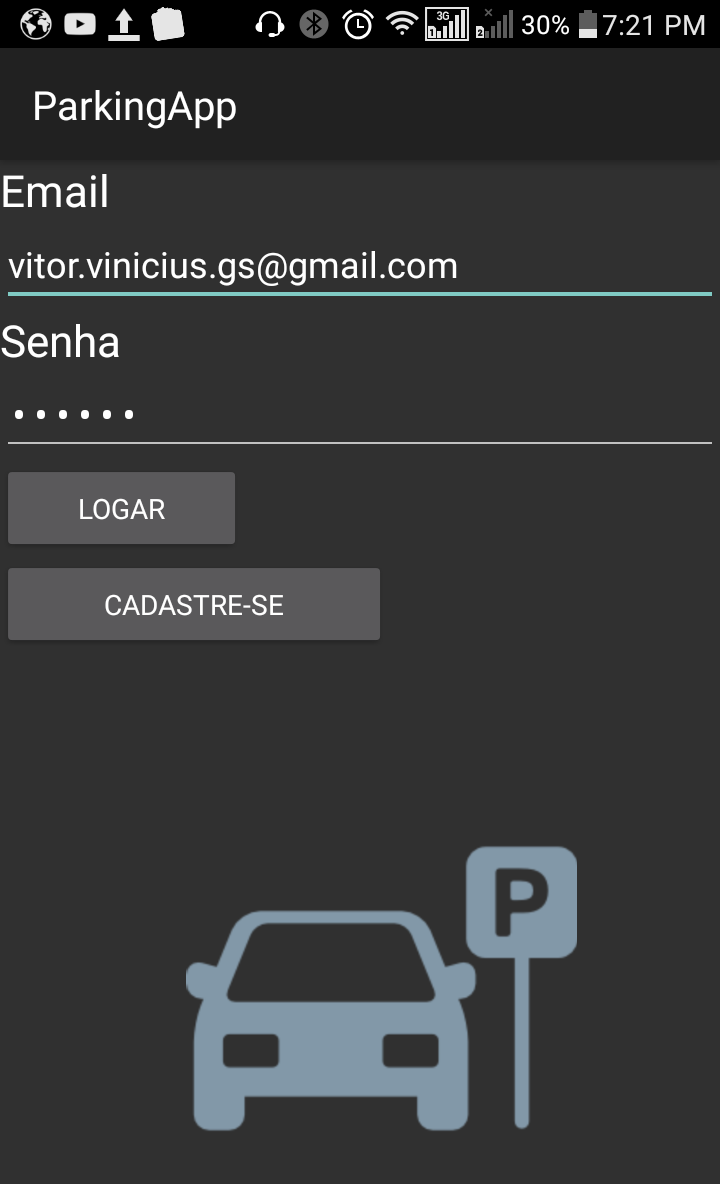


Figura 7.4.2: Tela de Logon: o usuário entra com o email e senha anteriormente cadastrados.

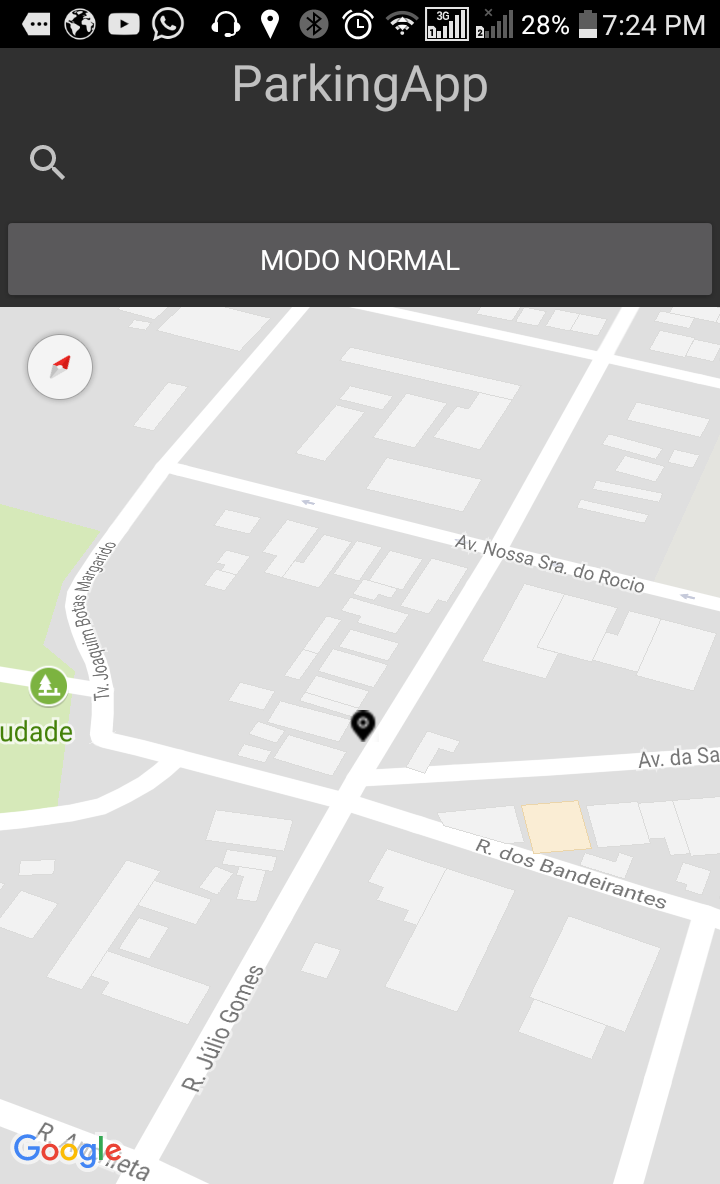


Figura 7.4.3: Tela Principal: mostra a localização atual do usuário. Permite a pesquisa de estacionamentos.

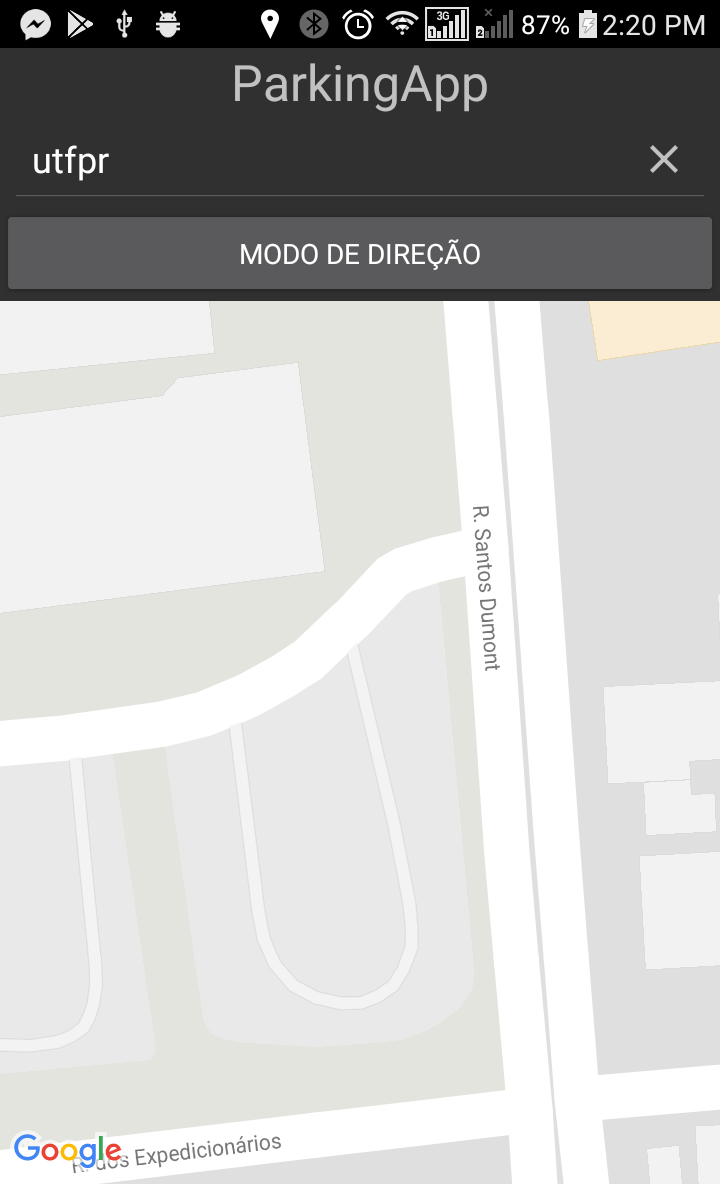


Figura 7.4.5: O usuário pesquisa por um estacionamento.

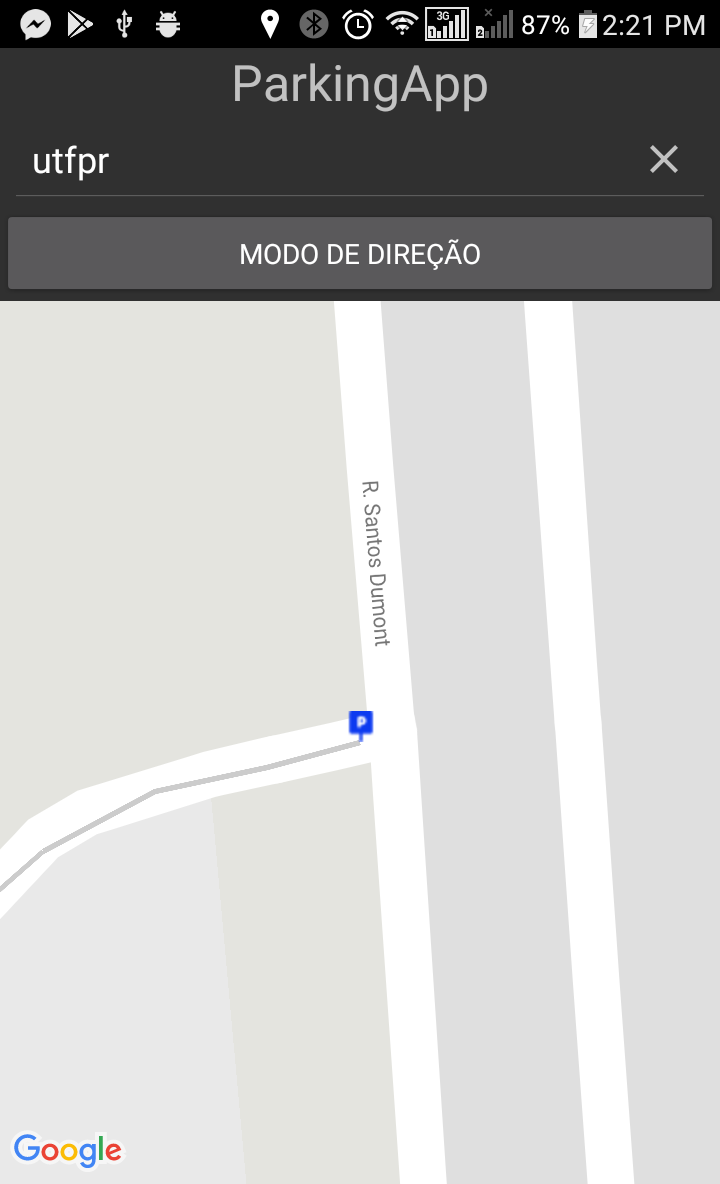


Figura 7.4.6: O aplicativo busca na Web API pelo termo de pesquisa informado. Retorna a localização do estacionamento que corresponde à pesquisa.

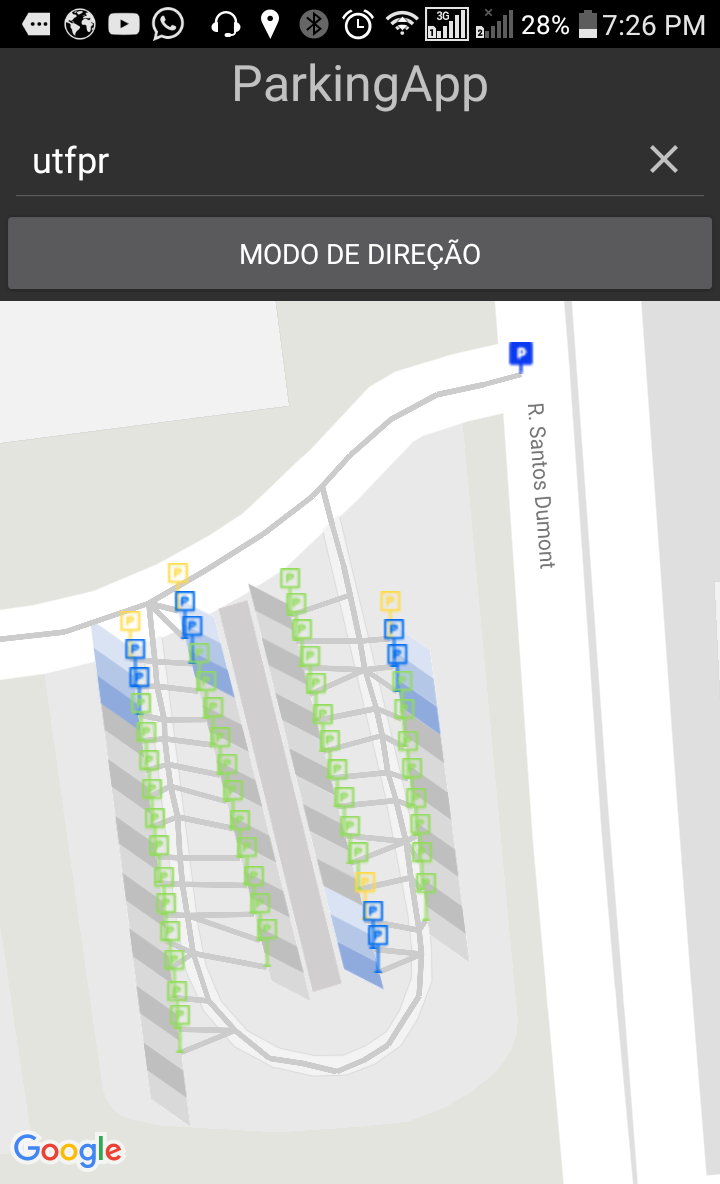


Figura 7.4.7: Visão Geral das vagas do estacionamento pesquisado.

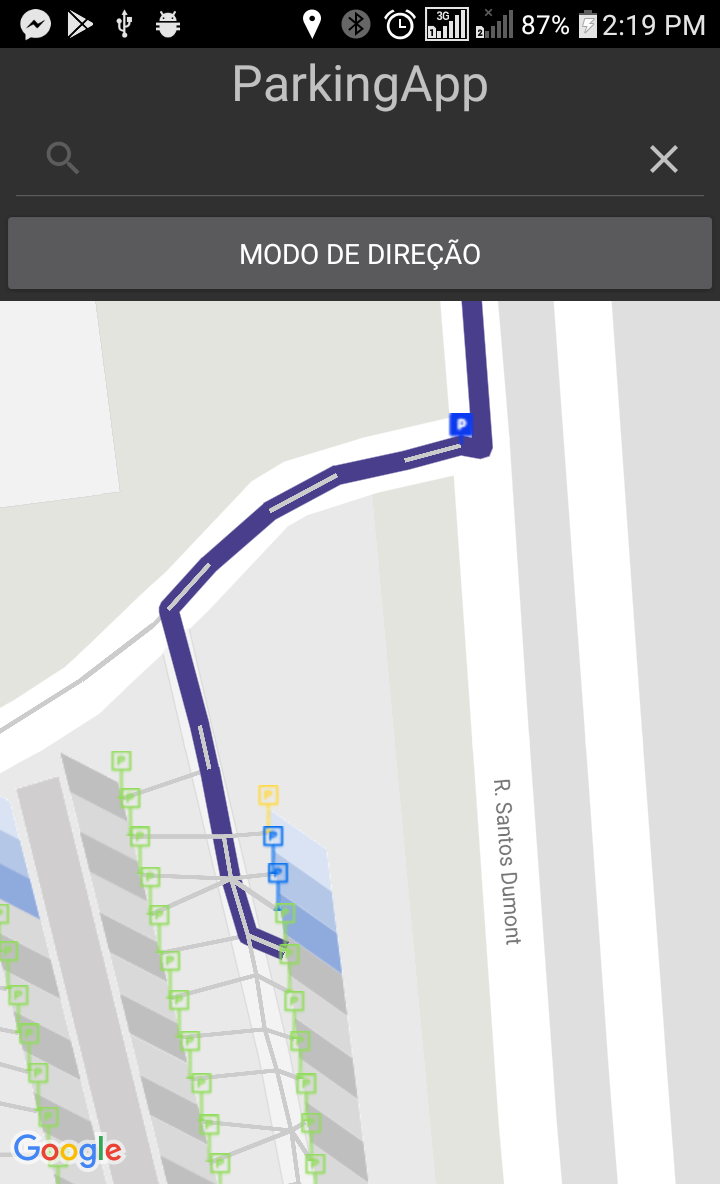


Figura 7.4.8: O usuário toca na vaga desejada. O aplicativo calcula a rota desde a posição atual do usuário até a vaga desejada passando pela entrada do estacionamento.



Figura 7.4.9: O usuário habilita o Modo de Direção. A câmera é posicionada então o tempo todo atrás da posição do usuário à 45º do chão.

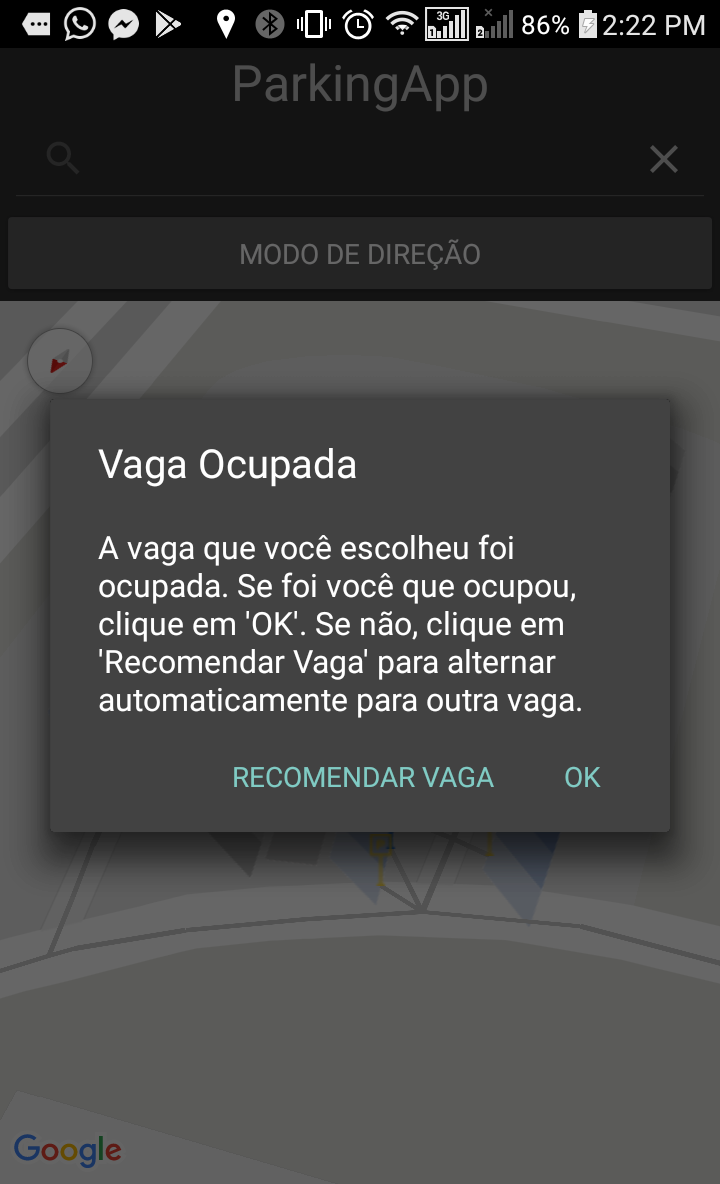
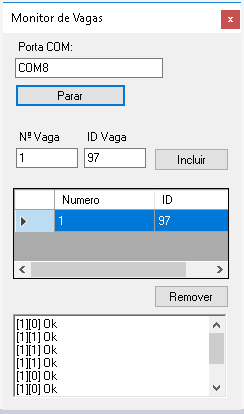


Figura 7.4.10: Sendo a vaga ocupada, os sensores instalados avisam a Web API. O aplicativo rastreia a ocupação e questiona o usuário se foi ele quem ocupou. Caso sim, a vaga some do mapa enquanto estiver ocupada. Caso não, o aplicativo sugere e orienta o usuário para outra vaga.

**7.5 Cliente Monitor de Vagas**

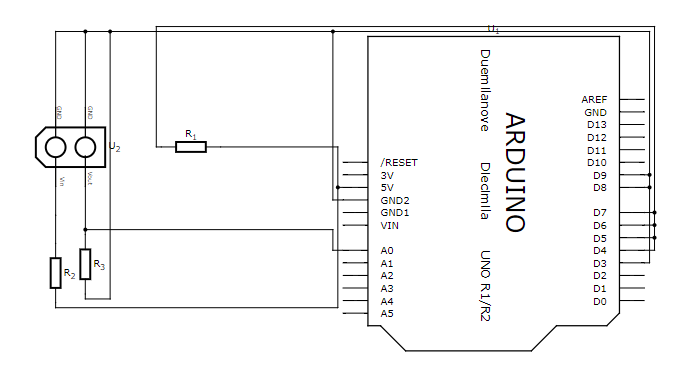
Programa responsável por receber os dados de ocupação de vagas do circuito montado com o Arduíno conectado via porta serial à um computador. As informações como estados dos sensores de proximidade conectados no Arduino são enviadas no formato XML para o Monitor de Vagas. Constantemente, o monitor envia os dados das vagas para a API. Desta forma, o monitor é o principal responsável por informar a ocupação ou liberação das vagas à Web API. A seguir pode ser visto uma imagem do programa em funcionamento.



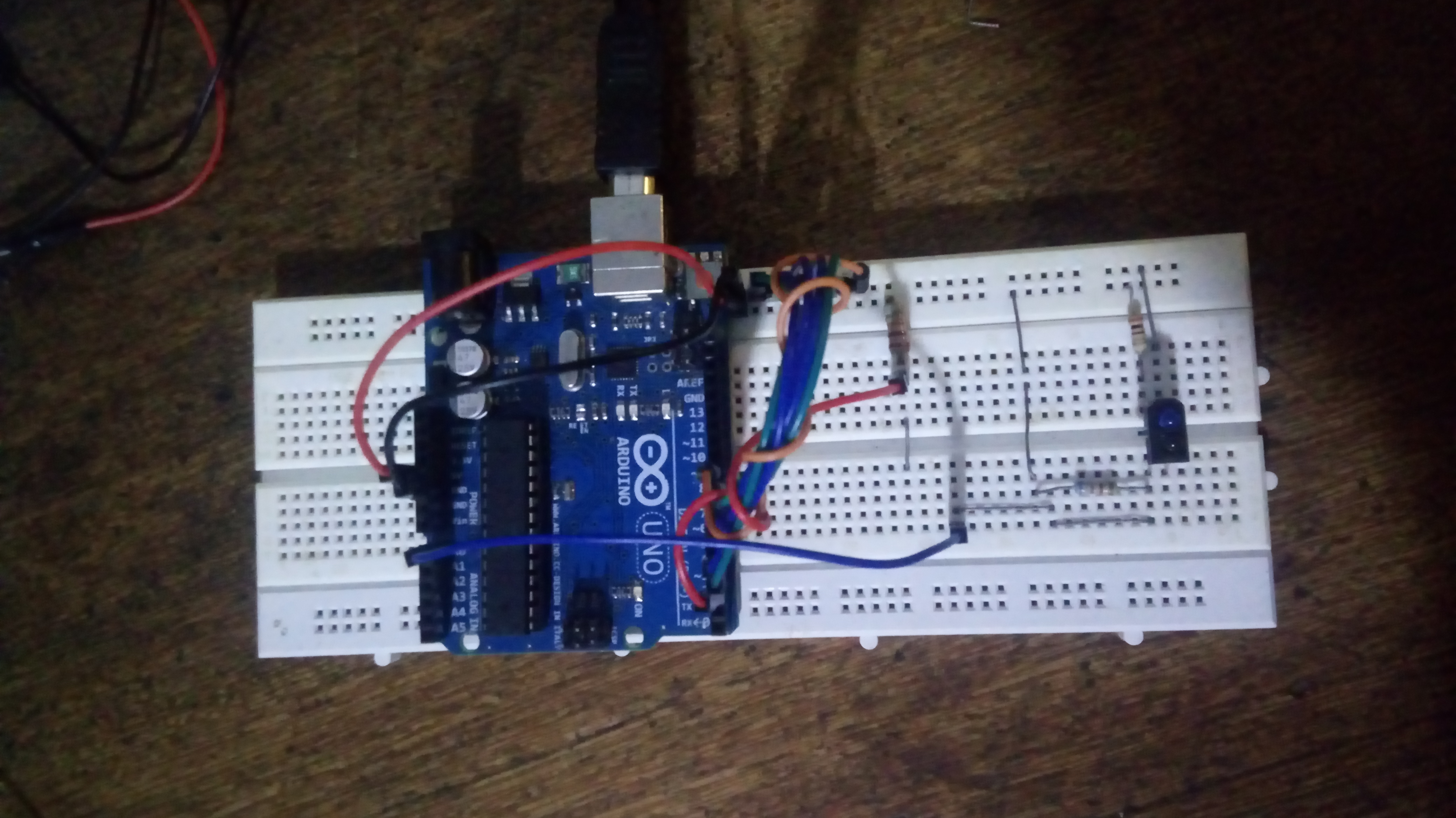
No Monitor de Vagas são configurados a porta de comunicação serial que será utilizada para se comunicar com o Arduino e a relação Nº Vaga do sensor x ID Vaga na Web API. O progresso de cada requisição e mudanças de estado das vagas podem ser vistos na caixa de texto inferior.

**7.6 Circuito de Sensores de Vagas**

Este circuito é responsável por receber dados dos sensores de proximidade TCRT 5000 conectados às vagas de estacionamento. Os sensores são conectados às portas analógicas do micro controlador Arduino sendo esse, capaz de gerenciar até seis sensores de proximidade. O microcontrolador encapsula as informações dos sensores em um envelope XML e o envia pela porta serial para o computador que executa o Monitor de Vagas. A seguir é apresentado o esquemático do circuito:



Neste esquemático, o sensor TCRT 5000 é conectado à porta analógica A0 do Arduino. As portas de 2 até 6 são ligadas na forma de um vetor de 5 bits para corresponder a sensibilidade do controlador. As portas de 7 até 9 são ligadas na forma de um vetor de 3 bits para corresponder à quantidade de sensores conectados à placa. Até 6 sensores podem ser ligados às portas analógicas e cada porta analógica do Arduino corresponde à uma vaga. No máximo 6 vagas podem ser gerenciadas sem multiplexação.

A seguir é apresentado uma foto do circuito montado em uma protocolar:

**8 BIBLIOGRAFIA**

[1] Google Maps ganha recurso para mostrar onde é fácil estacionar no Brasil. G1. Disponivel em: <http://g1.globo.com/tecnologia/noticia/google-maps-ganha-recurso-para-mostrar-onde-e-facil-estacionar-no-brasil.ghtml>. Acesso em 31 de agosto de 2017.

[2] CONTROLE INTELIGENTE DE VAGAS PROSIGA, S/D. Disponivel em: < http://www.prosiga.com.br/controle-inteligente/ >. Acesso em 31 de agosto de 2017.

[3] Bosch cria sistema que acha vagas de estacionamento AUTOMOTIVE BUSINESS. Disponivel em: <http://www.automotivebusiness.com.br/noticia/22924/bosch-cria-sistema-que-acha-vagas-de-estacionamento>. Acesso em 31 de agosto de 2017.

[4] ESTACIONAMENTO INTELIGENTE NA ESTANCIA TURISTICA DE AGUAS DE SÃO PEDRO. G1.

Disponivel em : <http://g1.globo.com/sp/piracicaba-regiao/noticia/2014/11/estacionamento-inteligente-mostra-vagas-desocupadas-por-aplicativo-guasdesaopedro.html>. Acesso em 31 de agosto de 2017.