**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA  
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE**

**Bacharelado em Engenharia de Software**

**Daniel Henrique Rodrigues Costa**

**João Vitor Alves Teixeira\***

**Mateus Santos Fonseca**

**Tiago Botelho Arabi Gonçalves\***

**SOLUÇÃO DE CARONAS UNIVERSITÁRIAS**

Belo Horizonte

2018

**Daniel Henrique Rodrigues Costa**

**João Vitor Alves Teixeira\***

**Mateus Santos Fonseca**

**Tiago Botelho Arabi Gonçalves\***

**SOLUÇÃO DE CARONAS UNIVERSITÁRIAS**

Trabalho de Software apresentado como requisito parcial à aprovação na disciplina Trabalho Interdisciplinar de Software II.

\*Observação: os alunos marcados pelo “\*” colaboraram até a segunda entrega, uma vez que desistiram da disciplina.

Professores: Tadeu dos Reis Faria e Hugo Bastos de Paula.

Belo Horizonte

2018

**SUMÁRIO**

[**1. Definição do modelo de negócios 5**](#_Toc517101110)

[**1.1. Objetivo do trabalho 5**](#_Toc517101111)

[**1.2. Modelo de negócios 6**](#_Toc517101112)

[**1.3. Definição dos perfis do usuário 7**](#_Toc517101114)

[**2. Modelagem do processo de negócio 9**](#_Toc517101115)

[**2.1. Descrição do processo 9**](#_Toc517101116)

[**2.2. Propostas de melhorias 10**](#_Toc517101117)

[**2.3. Indicadores de desempenho 11**](#_Toc517101118)

[**3. Projeto da Solução 12**](#_Toc517101119)

[**3.1. Requisitos funcionais 12**](#_Toc517101120)

[**3.2. Desenho das interfaces 14**](#_Toc517101121)

[**3.3. Diagrama de Classes 24**](#_Toc517101122)

[**3.4. Tecnologias utilizadas 25**](#_Toc517101123)

[**4. Implementação 26**](#_Toc517101124)

[**4.1 Node.js e npm 26**](#_Toc517101125)

[**4.1.1 Framework Ionic, cordova, Angular e SQLite 26**](#_Toc517101126)

[**4.2 Sublime Text 3 28**](#_Toc517101127)

[**4.3 Github e GitKraken 28**](#_Toc517101128)

[**4.4 O que não foi plenamente implementado 30**](#_Toc517101129)

[**4.5 Planos futuros para o projeto 30**](#_Toc517101130)

[**5 Considerações finais sobre o projeto 30**](#_Toc517101131)

[**REFERÊNCIAS 31**](#_Toc517101132)

# 1. Definição do modelo de negócios

## 1.1. Objetivo do trabalho

Para grande parte dos universitários, o deslocamento é um fator complicador. Atividade que gera estresse, gastos e demanda muito tempo diariamente. A dependência de transporte coletivo pode agravar o tempo e estresse e para os que tem transporte particular, o gasto é o principal problema.

Visando diminuir os problemas dos universitários que se encaixam nas características citadas, propomos desenvolver um software, onde estudantes com transporte privado, cadastram as rotas que fazem, para os que necessitam de transporte público, avaliarem se é compatível com seu trajeto.

Objetivos específicos:

- Diminuir os problemas geradores do trânsito diário para estudantes (principalmente estresse, tempo gasto e despesa financeira).

- Compartilhar rotas visando atender com facilidade e rapidez a necessidade dos universitários.

- Ampliar a comunicação interpessoal dos discentes, sendo que o software visará atender a todos os universitários de uma faculdade específica ou região.

- Promover a segurança para todos os usuários independente do gênero, com foco no gênero feminino, devido ao assédio moral e sexual.

## 1.2. Modelo de negócios

## 

Figura 1 – Canvas do grupo – Feito no site Sebrae Canvas

O modelo de negócios do grupo consiste -por meio da facilitação da interação entre universitários motoristas e universitários que buscam opções em conta de transporte (“caroneiros”)- na proposta de valor de carona universitária para divisão de custos (como gasolina e manutenção, por exemplo).

O segmento de mercado é, em essência, o público universitário, a princípio alunos da PUC Minas Unidade Praça da Liberdade.

Os canais de distribuição pretendidos são aplicativo móvel, site e redes sociais.

A estrutura de custos, assim como consta no canvas feito no site “SebraeCanvas”[[1]](#footnote-1), são além da estrutura online de host, domínio, hospedagem e anúncios, a distribuição percentual do retorno da fonte de renda que é um percentual sobre cada corrida.

Os parceiros chave são basicamente fornecedores. O grupo planeja conseguir uma validação de estudante com a PUC Minas, além de uma interação facilitada de usuários por mês via Google Analytics. Os alunos que dirigem (referidos como motoristas) atuam como parceiros na geração de valor.

Por fim, a relação com o cliente se dará por meio de aplicativo móvel e site da solução. Além disso, por meio de redes sociais também haverá o contato. No próprio aplicativo móvel, o “caroneiro” receberá créditos por corrida fechada e o motorista, ao realizar corridas, paga ao grupo um percentual a definir (provavelmente 15% a 20%) – a cada 10 corridas, 1 receberá o valor integral.

Durante uma pesquisa de mercado[[2]](#footnote-2), que até no presente momento se encontra em aberto, o grupo percebeu a necessidade do sistema englobar uma categoria só para mulheres. Tal necessidade foi descoberta por feedback e comunicação com universitárias (por um formulário online anônimo e segmentado) que abordaram o tema em um campo específico para feedbacks.

## 1.3. Definição dos perfis do usuário

O sistema em si, tem como participantes - chave dois tipos de usuários: os que oferecem caronas (motoristas) e os que procuram por caronas (“caroneiros”), sendo que ambos os tipos contam com a segmentação possível exclusiva para o sexo feminino. Contudo, o perfil de ambos tem bastantes semelhanças: adultos (a faixa etária do público alvo do projeto consiste em 18 a 30 anos, podendo ser utilizado sem problemas por maiores de 30 anos); alunos de um curso de graduação (ou pós) – a princípio da própria PUC Minas Unidade Praça da Liberdade- confirmados no cadastro por meio de declaração de matrícula; não é direcionado a um gênero em específico – qualquer identificação social pode utilizar livremente o sistema; pessoas que buscam por economias – seja de gasolina no caso do motorista, seja de tempo no caso do "caroneiro”; existindo “caroneiros” com necessidades especiais/restrições, haverá um filtro para motoristas que também atendam esse público.



Figura 2 – Formulário de validação de mercado. Respostas Disponíveis no site: http://bit.ly/FormsTis2

# 2. Modelagem do processo de negócio

## 2.1. Descrição do processo

O processo é iniciado com o cadastro de um aluno universitário da “PUC Minas – Praça da liberdade”, onde o mesmo escolherá a opção de motorista ou “caroneiro”. Em ambos, o discente deverá informar seus dados pessoais (nome, CPF, foto) e declaração de matrícula. Para o motorista haverá o campo para o número da Carteira Nacional de Habilitação (CNH) e a opção para incluir as rotas de ida e/ou volta com os pontos de parada, a fim de promover melhor mobilidade as rotas poderão ser alteradas. Após a verificação os dados deverão ser confirmados e para finalizar o cadastro os termos de uso do aplicativo deverão ser aceitos. Em ambos os casos será criado “login” e senha.

Após o cadastro, o motorista, para criar sua proposta de carona, selecionará entre suas rotas cadastradas previamente com seus pontos de parada, incluirá na oferta o período e o valor total pelo tempo determinado anteriormente entre o “caroneiro” e ele, a criando o seu “Card de Oferta”, nele também terá o campo opcional de gênero, filtrando os que podem pegar carona. O motorista pode gerenciar suas rotas quando quiser, gerando uma nova, alterando ou excluindo.

O motorista, ativará o “Card de Oferta” para iniciar o processo de negociação. O “caroneiro” visualizará as propostas, filtradas pelo gênero escolhido, que poderá ser selecionado no aplicativo após efetuar o login. Ele terá três opções: aceitar uma carona selecionada, que enviará uma notificação para o motorista com o ponto destino selecionado; ignorar caso não seja de seu interesse ou registrar uma contra-oferta. No último caso o “Card de Oferta” será alterado com as novas informações e enviado ao motorista.

Agora o motorista escolhe se a contra-oferta será aceita. Em caso positivo a carona será confirmada. Em caso negativo ele escolhe se encerrará o processo ou enviará um “Card de Oferta” alterado para o “caroneiro”, sendo essa a última fase de atualização do processo. Caso a alteração seja aceita a corrida é confirmada, se não o processo é encerrado.

Após as confirmações o “caroneiro” fará o pagamento, o aplicativo receberá o valor e fará a transferência para o motorista retirando uma porcentagem. Assim, o motorista irá ao local de partida, se forem várias caronas combinadas no “Card de Oferta” o local de partida já estará indicado. Quando todos os envolvidos estiverem no carro, o motorista levará todos para o primeiro ponto de partida até que não haja mais destinos. Ambos ao finalizar a carona farão a avaliação para futuras recomendações.

O grupo mapeou os processos em fluxogramas feitos no software Bizagi Modeler. Os processos, mapeados em cada um dos documentos, podem ser encontrados no link: https://github.com/caronastis2/doc.

## 2.2. Propostas de melhorias

O aplicativo apresenta uma proposta diferenciada em relação às soluções de mobilidade particulares já existentes, sendo organizado de forma a proporcionar segurança e conforto para ambos, caroneiros e motoristas, além de apresentar uma opção sobre caronas exclusivas por gênero.

Com a preocupação de entender e atender o mercado, o grupo por meio de uma pesquisa, constatou que a motivação das pessoas para aceitar a carona é uma das principais barreiras para a aceitabilidade da mesma. De acordo com dados levantados, há uma baixa probabilidade de caroneiros aceitar caronas de um motorista qualquer, contudo, o índice de caroneiros que aceitam caronas de motoristas da mesma instituição é alto.

A proposta desta solução visa garantir a segurança de seus usuários, pois garante que apenas alunos da mesma instituição utilizarão a plataforma para negociação de caronas. Além disso a praticidade do motorista definir o trajeto e poder negociar diretamente com os caroneiros que se interessam pela rota é um diferencial do aplicativo, viabilizando a carona. Outro ponto a ser destacado é a melhoria do custo para o caroneiro, podendo até sair mais barato- e proporcionando mais conforto- do que o transporte público.

Por meio de comentários em pesquisa do grupo, foi definido que poderá ser filtrado canoreiros e motoristas pelo mesmo gênero, desta forma proporcionando uma melhor experiência para ambos. No entanto, através da mesma pesquisa de campo o grupo pôde observar que o aplicativo seria necessário somente em primeira instância para estabelecer a relação motorista/caroneiro, uma vez que ambos tivessem um vínculo iriam deixar a solução de lado. Levando isso em consideração, o grupo planeja recompensar ambos tipos de usuário de alguma forma para engajar e fidelizar mais pessoas ao aplicativo em relações futuras – mantendo e potencializando o crescimento de Monthly Active Users (MAU`s).

Em relação ao modelo TO BE, tendo em vista que nosso programa ainda não foi implementado até o presente momento, o modelo citado faz referência aos fluxogramas de processo que foram descritos no item 2.1.

## 2.3. Indicadores de desempenho

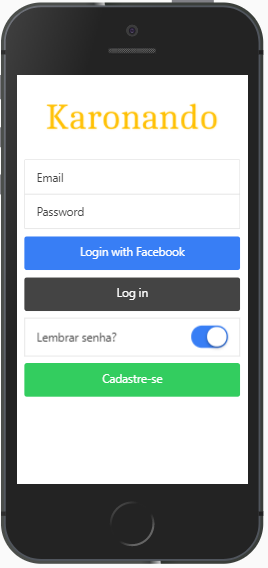
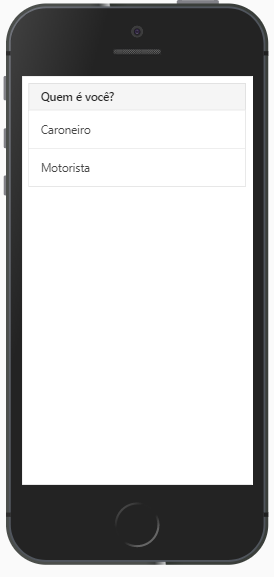
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Indicador** | **Objetivos** | **Descrição** | **Cálculo** | **Fonte dados** | **Perspectiva** |
| Número de caroneiros | Analisar quantitativamente o número de caroneiros | Soma de usuários caroneiros da solução |  | Tabela caroneiros | Clientes |
| Número de motoristas na plataforma | Análise quantitativa de motoristas na plataforma. | Soma de usuários motoristas da solução |  | Tabela motoristas | Motoristas (em alcance). |
| Quantidade de caronas confirmadas | Avaliar o alcance quantitativo de caronas confirmadas. | Soma caronas fechadas. |  | Tabela Cards de Oferta | Avaliar Cards fechados |
| Caronas bem sucedidas | Avaliar o índice % (quantitativo) de caronas bem sucedidas | Mede % caronas bem sucedidas por caronas fechadas |  | Tabela negociações | Clientes |
| Corridas confirmadas em relação a oferta | Avaliar o percentual de corridas confirmadas em relação a oferta de caronas | Mede % do nº de corridas confirmadas pelo nº de ofertas de caronas |  | Tabela motorista e tabela caroneiro | Avaliar interação de usuários |

# 3. Projeto da Solução

## 3.1. Requisitos funcionais

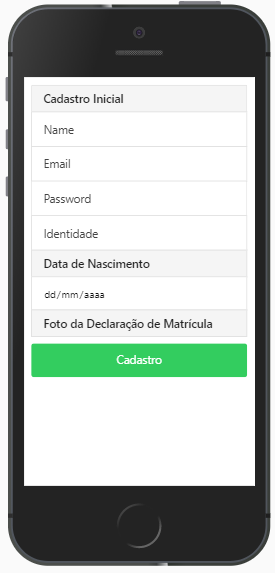
|  |  |  |  |
| --- | --- | --- | --- |
| **Número de Ordem** | **Requisito** | **Descrição** | **Prioridade** |
| **[Número sequencial identificador do requisito.]** | [Nome do requisito.] | [Descrição resumida do requisito.] | [Alta, média ou baixa.] |
| 1 | Cadastrar Rota | Eu como motorista quero cadastrar minha rota para que os “caroneiros” visualizem os pontos de parada | Alta |
| 2 | Alterar Rota | Eu como motorista quero alterar uma rota já cadastrada para continuar operando pelo app | Média |
| 3 | Cadastrar Usuário | Eu como usuário quero me cadastrar para poder utilizar o app | Alta |
| 4 | Alterar Cadastro | Eu como usuário quero alterar meu cadastro para poder garantir o melhor uso do aplicativo com minhas informações atualizadas | Baixa |
| 5 | Filtrar Companheiro de Carona (Diferencial) | Eu como usuário quero poder filtrar por identificação de sexo meus companheiros de viagem para poder me sentir mais seguro | Alta |
| 6 | Cadastrar Card | Eu como motorista quero cadastrar um card de oferta para divulgar minhas rotas, período, preço | Alta |
| 7 | Gerar Novo Card | Eu como motorista quero incluir um novo card para entrar com minhas rotas alternativas | Média |
| 8 | Ativar Card | Eu como motorista quero ativar o card para colocar minha proposta para negociação | Alta |
| 9 | Visualizar Card | Eu como “caroneiros” quero visualizar o card de proposta para selecionar a ideal ao meu trajeto | Alta |
| 10 | Ignorar Card | Eu como “caroneiro” quero ignorar um card por não condizer com o esperado | Baixa |
| 11 | Registrar Contraoferta | Eu como usuário quero registrar uma contraoferta na tentativa de alinhar um melhor preço com a outra parte | Média |
| 12 | Receber Contraoferta | Eu como usuário quero receber a contraoferta para poder avaliar se a oferta me atende | Alta |
| 13 | Aceitar Proposta Inicial | Eu como “caroneiro” quero aceitar a proposta inicial para confirmar minha carona | Alta |
| 14 | Aceitar Contraoferta | Eu como usuário quero aceitar a contraoferta para poder aceitar a carona | Alta |
| 15 | Receber Mensagem | Eu como usuário quero receber mensagem ao fim de cada processo para ficar ciente do resultado | Média Baixa |
| 16 | Pagar Carona | Eu como "caroneiro" quero efetuar o pagamento para que minha carona seja confirmada | Alta |
| 17 | Receber Pagamento | Eu como motorista quero receber meu pagamento para iniciar a corrida | Alta |
| 18 | Avaliar Carona | Eu como usuário quero avaliar a carona para poder indicar e mostrar minha opnião | Baixa |

## 3.2. Desenho das interfaces

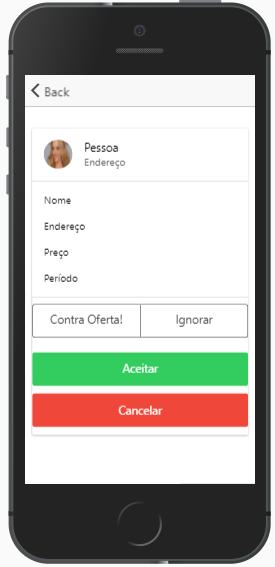
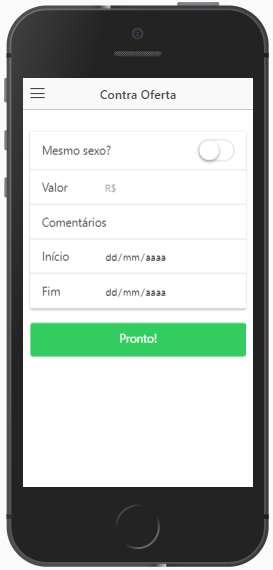


Tela 1. Login Tela 2. Seleção do tipo do usuário

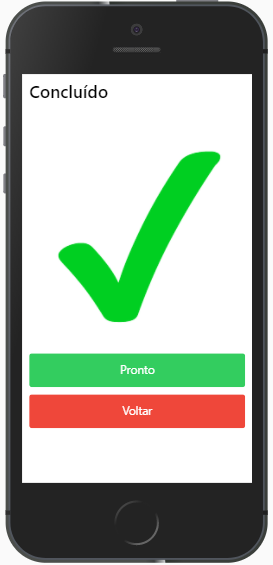
### 3.2.1 Caroneiro



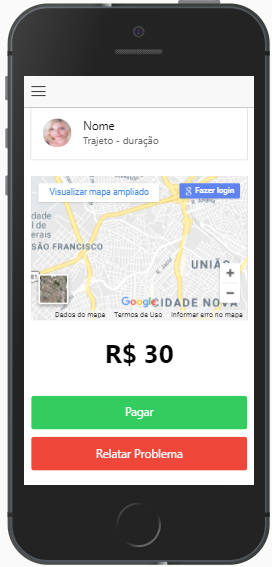
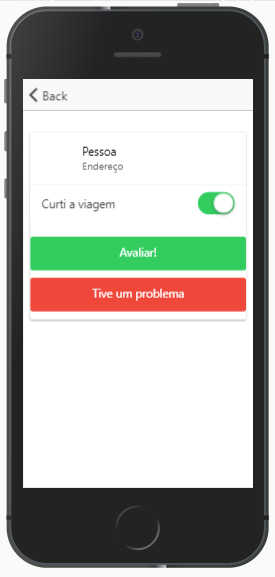
Tela 3. Cadastro do caroneiro. Tela 4. Ofertas para o caroneiro



Tela 5. Opções do caroneiro ao selecionar uma oferta Tela 6. Cadastrar contra oferta



Tela 7. Cadastro do meio de pagamento Tela 8. Concluído.

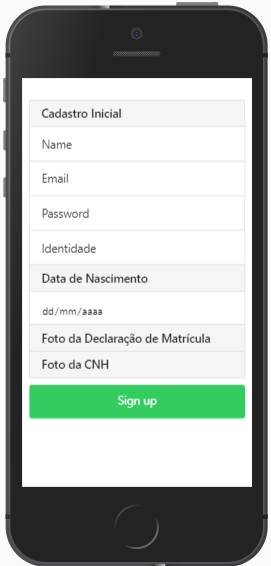


Tela 9. Confirmação do Pagamento Tela 10. Avaliação

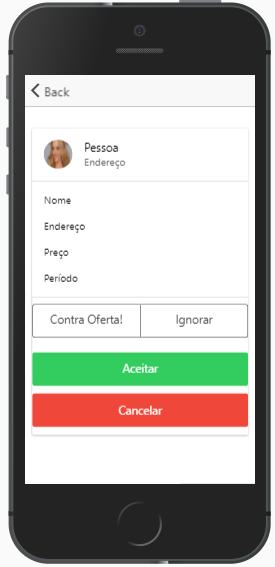
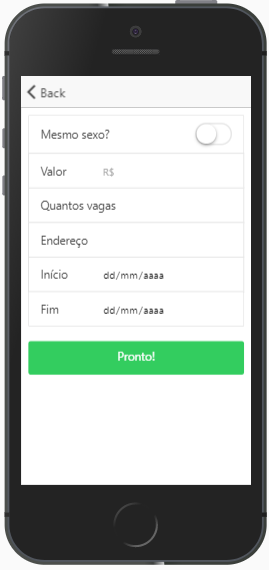


Tela 11. Problemas e Ajuda

### 3.2.2 Motorista



Tela 3. Cadastro motorista Tela 4. Ofertas ativas do motorista



Tela 5. Visualizar oferta Tela 6. Nova oferta





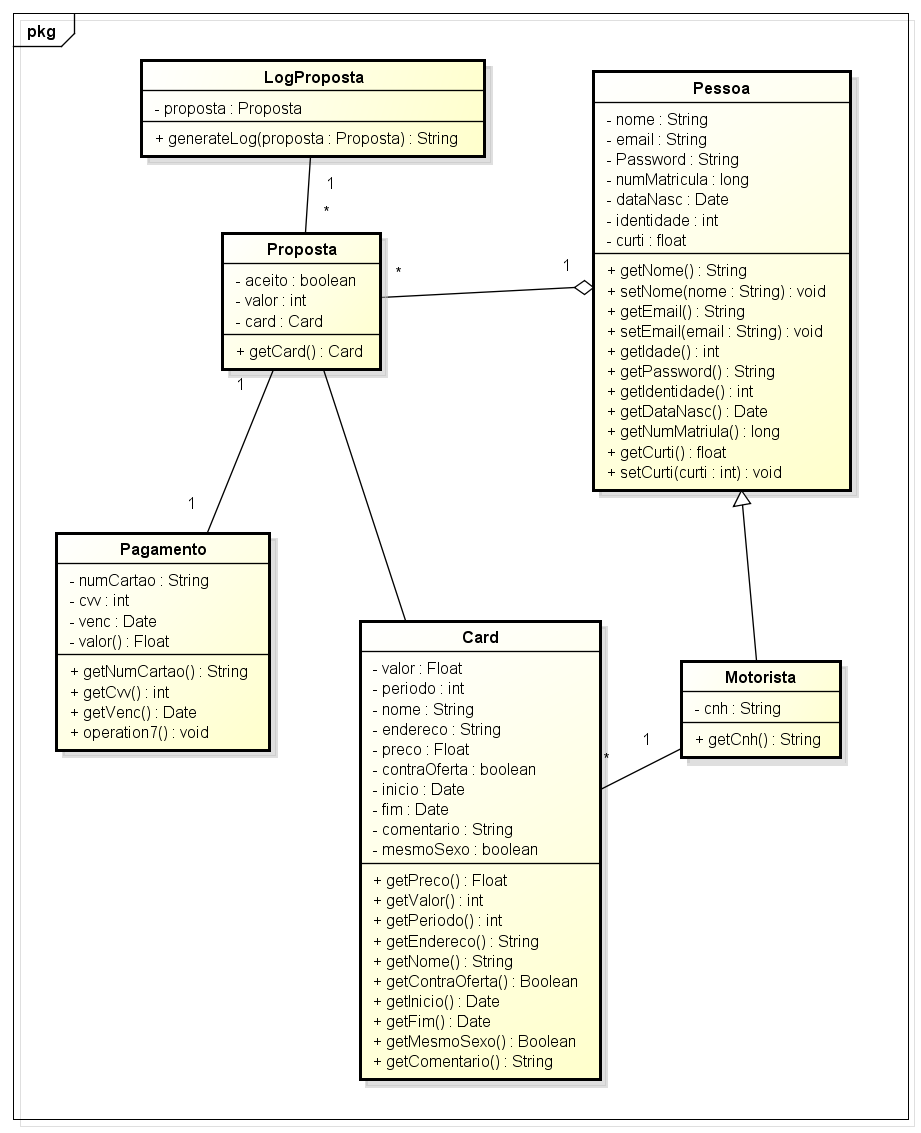
Tela 7. Avaliar Caroneiro Tela 8. Resgate motorista



Tela 9. Feedback

## 

## 3.3. Diagrama de Classes



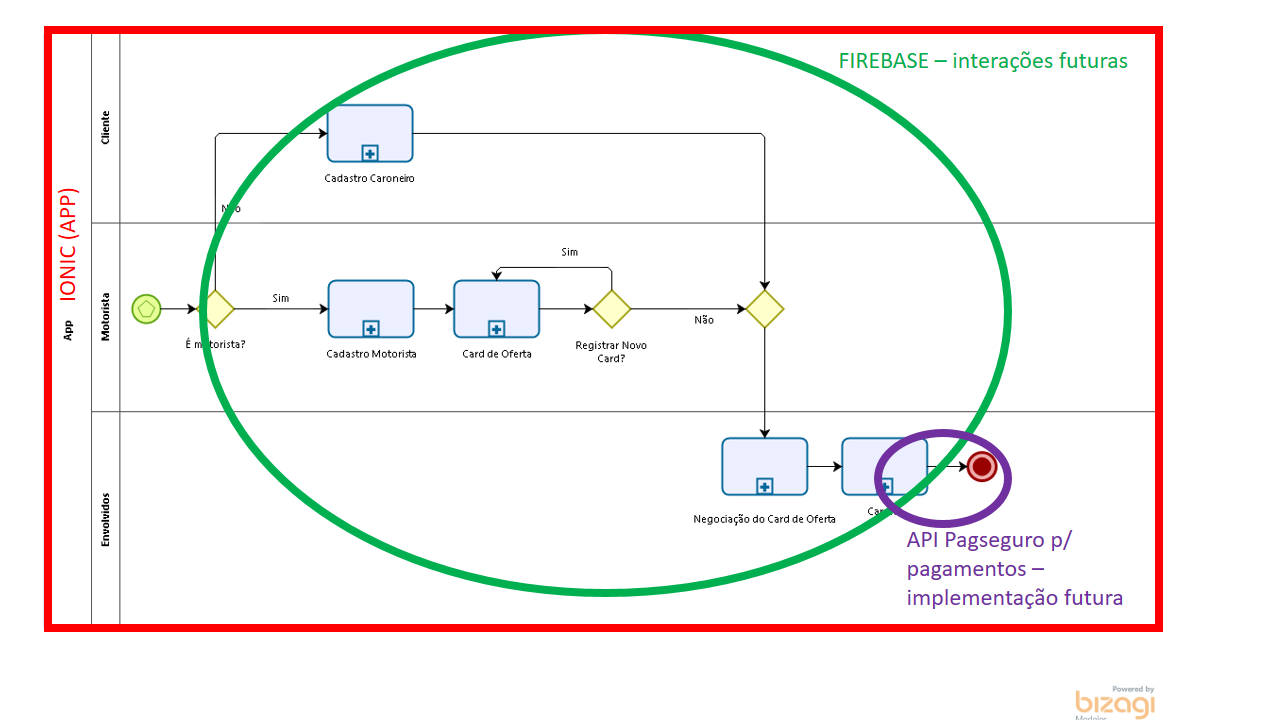
## 

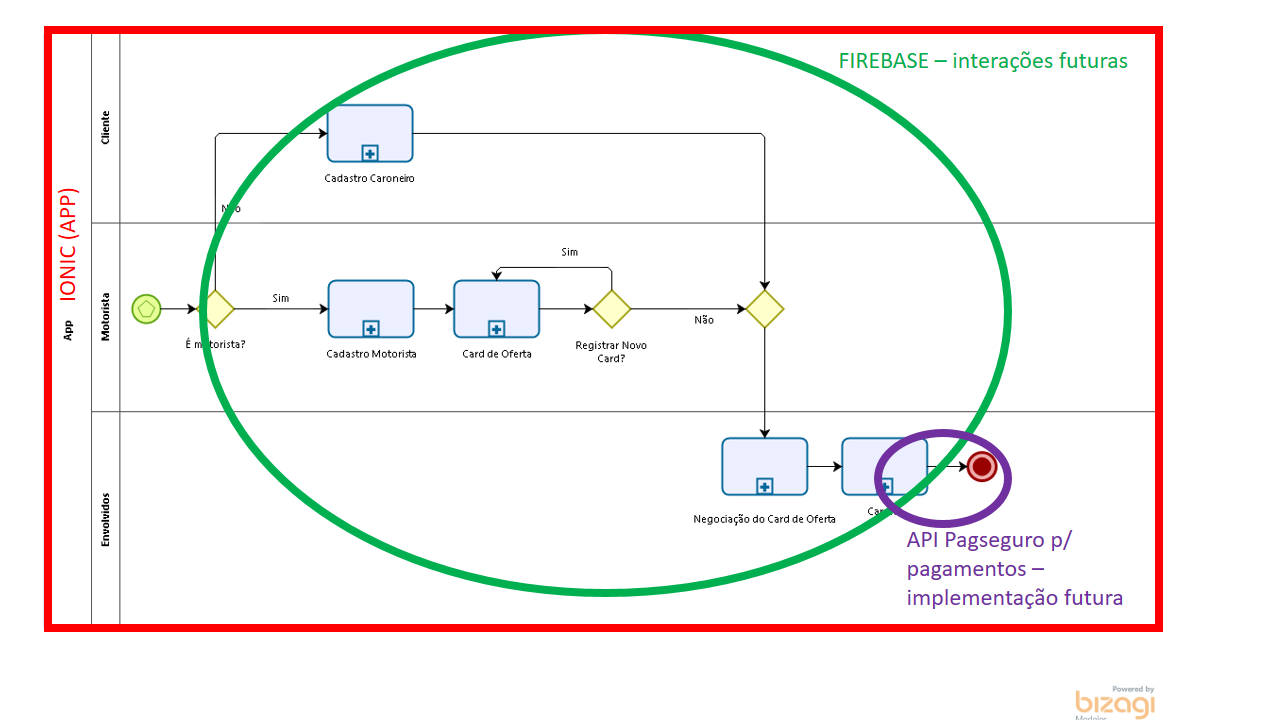
## 3.4. Tecnologias utilizadas

Para a implementação das telas, foi utilizado creator ionic- construtor online da Framework Ionic específico para a construção de apps. Assim, utilizou-se HTML, CSS e JS, para compor cada uma das telas do projeto. Para abrir o projeto em “localhost”, foram necessários os comandos “ionic start Karonando blank” e “ ionic serve” do Node.js para uma edição mais aprofundada dos códigos em editor de texto ( utilizado o Sublime Text 3).

Para termos de persistência, a solução utilizou SQLite em integração com Angular por meio do Nodejs npm.

Para o pagamento, será utilizada a API do PagSeguro, onde o dinheiro da corrida será pago à equipe do projeto e então repassada a parte do motorista.





Persistência: Angular+ SQLite

Interação resumida das tecnologias (implementadas e ainda não implementadas).

# 4. Implementação

Link do github com todos os códigos do projeto:

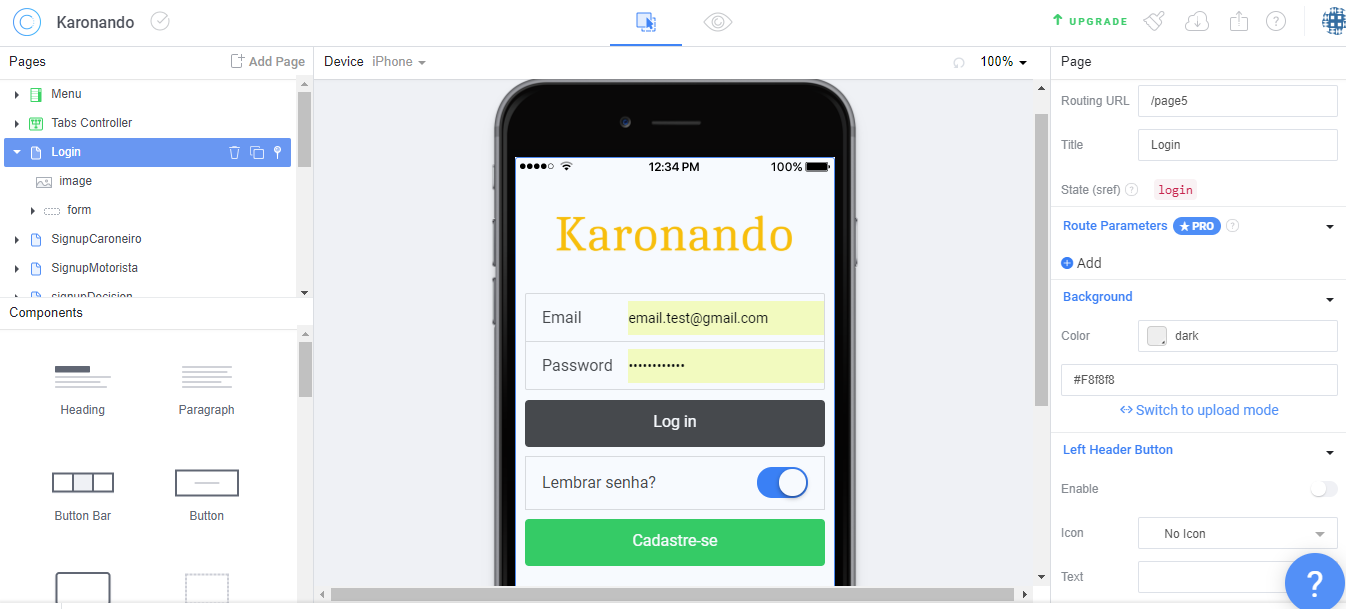
https://github.com/caronastis2/doc/tree/master/codigo

## 4.1 Node.js e npm

Node.js e npm têm diversas utilizações, até como iniciar um projeto ou um server. A utilização de Node.js se deu para a realização do projeto em Ionic.

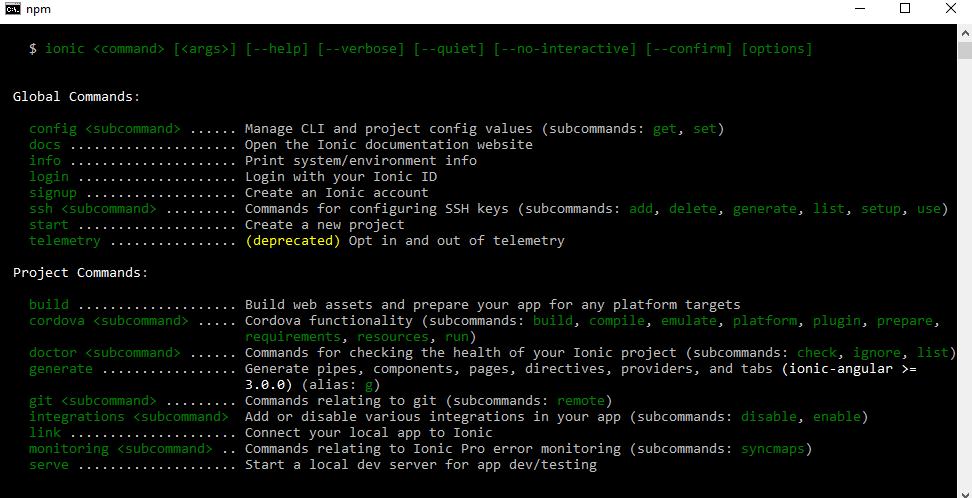
## 4.1.1 Framework Ionic, cordova, Angular e SQLite

A Framework Ionic, que utiliza linguagens web – html, CSS, JavaScript- possui um desenvolvimento facilitado para mobile com um Creator Online, que permite também a exportação de seus templates.



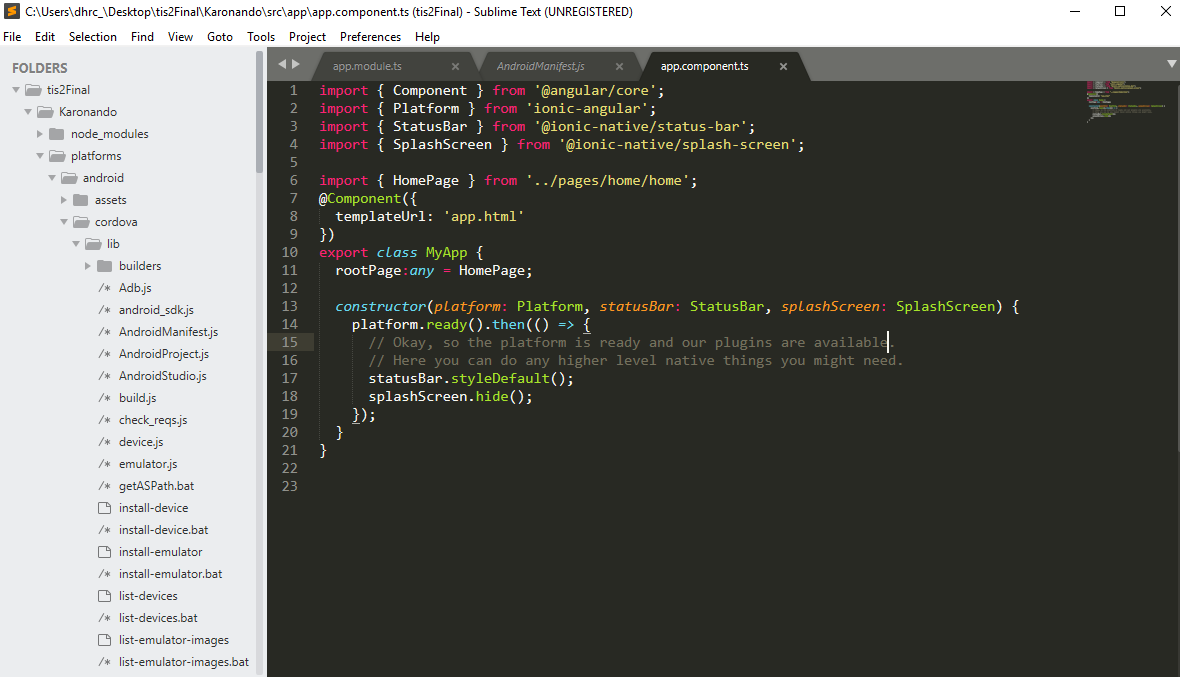
Ionic Creator – Facilitador de interface

As diretivas do código, tal como suas persistências foram feitas em Nodejs, npm, comandos cordova, Angular e SQLite. Nodejs e npm são ferramentas poderosas quando combinadas. O cordova foi útil no que se diz às plataformas: o projeto é completamente exportável para android e ios. O framework ionic necessita de um server provido pelo Nodejs e npm com comandos iniciados em “ionic”.



Terminal Node.js com npm para utilização dos comandos ionic

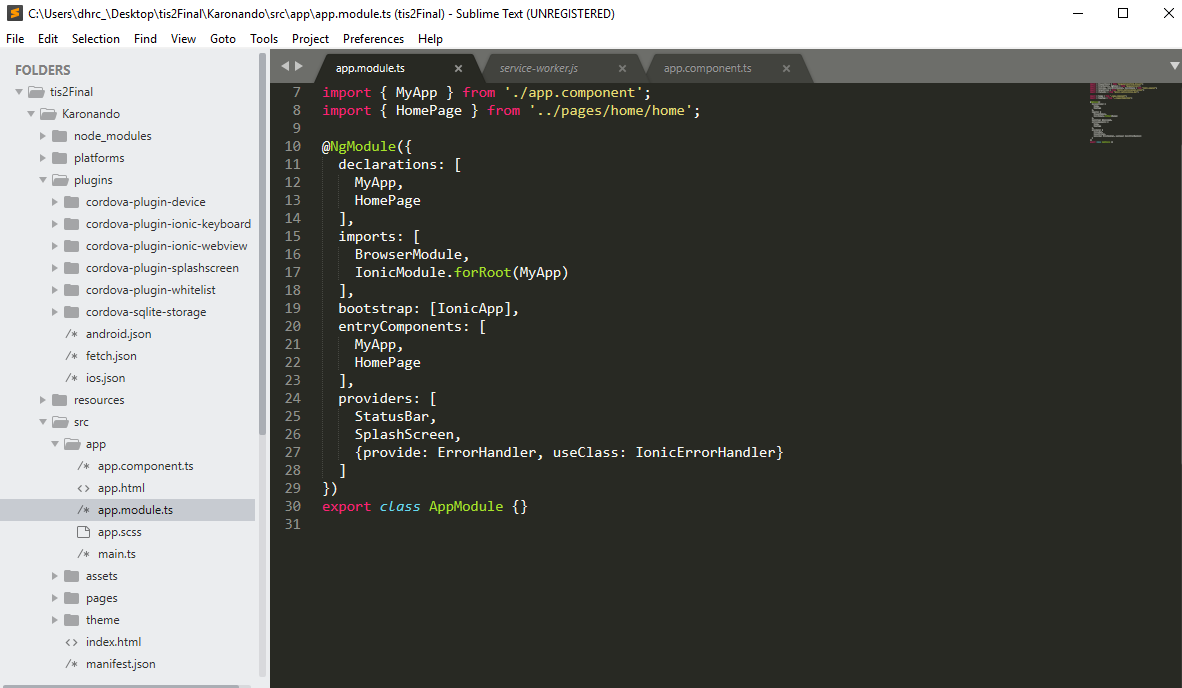
A database.ts (da notação dos arquivos angular) também foi feita com comandos pelo Nodejs, sendo sua provedora (provider) SQLite.



Arquivo do tipo .ts (componente) aberto no Sublime Text 3

# 4.2 Sublime Text 3

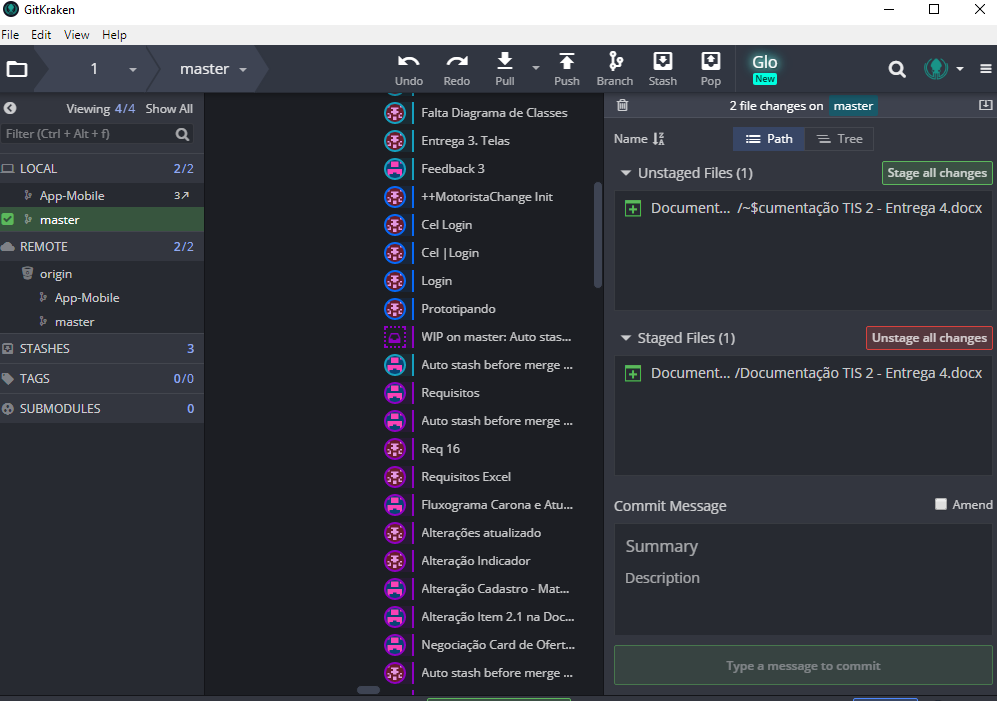
Utilizado como editor de texto. Além de um visual moderno e confortável, proveu inteligência em predição ao projeto por meio do Package Control (plug-ins open source integrados ao editor).



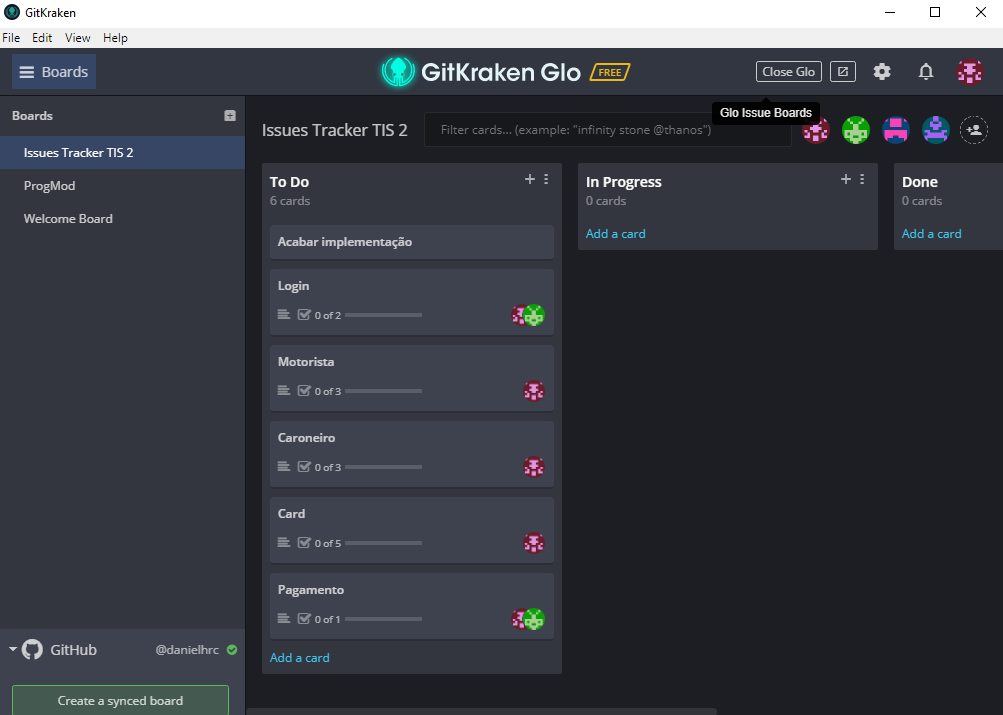
## 4.3 Github e GitKraken

Github, ferramenta para versionamento de software, foi utilizado para, além dessa função, controle de versão dos fluxogramas do processo de negócio deste projeto.

O GitKraken, Cliente visual de Git, bem semelhante ao Git GUI, possui, além do controle de versão, um gerenciador de tarefas baseado em Kanban chamado GitKraken Glo, utilizado brevemente pelo grupo (antes da desistência de 2 integrantes). Como somente um integrante implementou a solução, o uso deste método de gerenciamento não se fez justificado, sendo portanto abandonado.



Tela principal do Gitkraken: versionamento e gerenciamento de conflitos facilitado.



GitKraken Glo Boards. Ferramenta de gerenciamento de tarefas.

## 4.4 O que não foi plenamente implementado

A persistência de dados (SQLite), nem o pagamento (com API PagSeguro), foram plenamente implementadas, por questão da insuficiência de desenvolvedores no projeto. Como apenas um aluno desenvolveu todo o código ao longo do projeto, foi, por consequência, sobrecarregado, inviabilizando a execução total do projeto. Enquanto a persistência teve uma pequena implementação, o pagamento não foi implementado.

## 4.5 Planos futuros para o projeto

Implementar, além dos métodos de pagamento, a plena integração com o banco de dados.

# 5 Considerações finais sobre o projeto

A solução, do ponto de vista de sua ideação, foi realizada por completo, tendo a modelagem de negócios feita em grupo e seus processos totalmente detalhados. Do ponto de vista do código, com a ausência do efetivo trabalho de equipe – por conta da desistência da disciplina por parte de alguns integrantes- houve o prejuízo em termos de projeto. Contudo, o projeto tem demanda, passou por validação e colaborou para o desenvolvimento do aluno que efetivamente produziu o presente trabalho, sendo este aluno plenamente ciente da importância do trabalho em grupo e dos ensinamentos dos professores da disciplina.

# REFERÊNCIAS

COSTA, Daniel H. R. et al. **Carona Universitária**: Canvas feito na plataforma do Sebrae pelo grupo. Disponível em: <http://bit.ly/CanvasCaronas>. Acesso em: 25 fev. 2018.

COSTA, Daniel H. R. **Questionário Mobilidade Universitária**: Questionário desenvolvido no Microsoft Forms com o objetivo de validar o modelo de negócios do grupo. Disponível em: <http://bit.ly/QuestionarioTis2>. Acesso em: 25 fev. 2018.

COSTA, Daniel H. R. Repositório com Processos do Negócio – fluxogramas do Bizagi. Disponível em: < https://github.com/caronastis2/doc>. Acesso em: 02 abr. 2018.

1. http://bit.ly/CanvasCaronas [↑](#footnote-ref-1)
2. Disponível para respostas em http://bit.ly/QuestionarioTis2 e link para administração e respostas: http://bit.ly/FormsTis2 [↑](#footnote-ref-2)