|  |  |
| --- | --- |
|  | INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS - CAMPUS FORMIGA |

Aplicativo híbrido para corridas de mototáxi

Proposta de Trabalho de Conclusão de Curso

Nome do aluno: Luiza nascimento Bambirra

Orientador: Prof. Fernando Paim lima

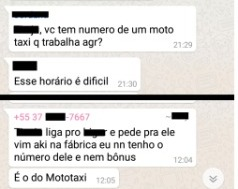
FORMIGA - MG  
2022

1 INTRODUÇÃO

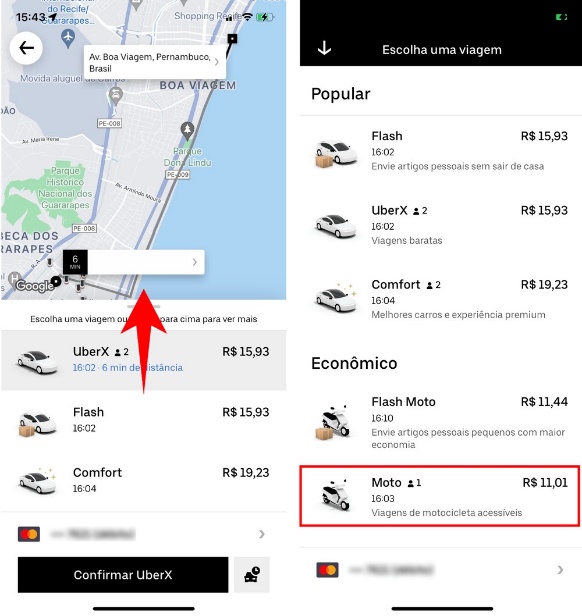
Hoje em dia nós ficamos muito dependentes de nossos dispositivos eletrônicos, sendo o principal destes o nosso telefone celular. A todo momento surgem novos *softwares* para auxiliar as pessoas nas tarefas cotidianas como: calculadora, mapas, calendários, transações bancarias, guias culinários, professores virtuais, entre outros. Os smartphones têm uma presença perceptível na vida das pessoas, pois eles facilitam a interação entre indivíduos. A função de um *smartphone* de efetuar ligações é mais uma entre as diversas possibilidades disponíveis nele, e com isso se foram criados diversos aplicativos moveis, em que sua função deveria ser ajudar em nosso cotidiano, como para trabalho, escola ou para diversão, sendo assim muito constantes em nossas vidas, nos deixando muito dependentes, Aplicativos de mapas por exemplo, indicam as rotas com melhor caminho onde são avaliados menor distância, pedágios, segurança do local, entre outros fatores. Não se pode ignorar o fato de que nem sempre o aplicativo *Global Position System* (GPS) está totalmente correto, podendo te guiar para bairros perigosos, mas é uma grande ajuda para quem vai de veículo próprio a alguma localidade que não conhece. Quando não se tem um veículo, uma alternativa é esperar ate que um ônibus ou metrô passe, o que muitas vezes pode demorar, além disso, dentro desse tipo de condução existe a possibilidade de não ter poltronas disponíveis para todos e a segurança e o conforto nem sempre são eficientes. Estar nessa situação pode ser mais complicado quando se utiliza dois ou três ônibus por dia. Outra alternativa para quem tem mais pressa e pode pagar um pouco mais e chamar um taxista mas estes muitas vezes cobram taxas abusivas das pessoas que não tem tanta noção dos preços, não prestam um serviço de qualidade e tendem a demorar mais para completar o trajeto por serem veículos maiores e mais pesados. Existem também como escolha para se locomover de modo rápido e barato, os mototaxistas. Essa classe de trabalhadores esta crescendo muito nas cidades pequenas, pela grande quantidade de pessoas que dependem desse tipo de serviços por não conseguirem se locomover para seu trabalho, escola ou qualquer tipo de compromisso. É notável a facilidade de deslocamento e rapidez de uma moto, por ser um veículo relativamente pequeno, econômico e leve, consegue percorrer por ruas e avenidas sem muitos problemas. Quanto aos preços e taxas, podem sair bem mais em conta se comparados aos dos táxis. O problema que geralmente encontra-se nas motos são as bagagens, que não dão tanto suporte assim para malas e bolsas grandes, o que faz com que muitas pessoas desistem das corridas de mototaxistas. Mas perante este cenário, pela sua agilidade e valor acessível muitas pessoas optam pelas motocicletas. As dúvidas surgem no momento em que se quer alguém de confiança e que faça todo o trajeto de modo seguro. Quando encontra-se um mototaxista que presta os serviços de modo idôneo e com preço razoável todos querem manter o contato com ele para transportes futuros. O problema é quando não se tem referências de pilotos capacitados, indicações de usuários que já passaram pela experiência são uma saída para fazer a escolha mais rapidamente, pode-se pegar cartões, anotar o número no celular ou então memorizar. Pensando nisso este trabalho de conclusão de curso foi desenvolvido para criar um aplicativo móvel híbrido que funcione com as chamadas dos motoristas mais próximos de sua localização atual, mas que é específico para mototaxistas, já que uma vez estes encontram muitos problemas na parte de sua divulgação pois na maior parte do tempo está é feita por meio de cartões que são entregues com seu número de telefone como na imagem abaixo.



Tudo isso para que possa ocorrer o contato com seus clientes, ou também podem trabalhar em pontos específicos onde apenas mototaxistas ficam e qualquer pessoa pode ir e pedir uma corrida, mas nem sempre se tem um disponível. Mas pense e se por qualquer motivo esse trabalhador decida mudar seu número de telefone, como neste já perde clientes que vão ficar sem seu contato e não conseguiram se comunicar. Isso é apenas a visão do mototaxista ainda temos os problemas que os usuários de mototáxis tem, como por exemplo os que necessitam de uma corrida e não tem o contato e não estão perto de nenhum ponto onde poderia encontrar mototaxistas livre para fazer uma corrida, sendo assim ficam a procura de um em suas redes sociais como WhatsApp como mostrado na imagem abaixo.



Ademais pode ocorrer de pessoas que necessitam de uma corrida ficarem muito tempo nos pontos de mototáxi esperando um livre aparecer para atendê-los. Atrelado a isso o aplicativo seria uma forma de diminuir este problema pois ele teria os mototaxistas cadastrados e pegaria suas localizações em tempo real, com isso os clientes poderiam fazer pedidos de corrida e este poderia aceitar ou não, e teria também um chat de conversa para facilitar o acesso de comunicação entre ambos, descartando assim as dificuldades de intercomunicação dos trabalhadores e junto a isso também resolverá os contratempos dos usuários de encontrar um mototáxi ou ficar horas nos pontos esperando um livre para atendê-lo, além de que no aplicativo só serão cadastrados mototaxistas de confiança para que a segurança do usuário seja garantida. Com todos os dados abordados este aplicativo está sendo desenvolvido para que possa ajudar tanto os mototaxistas mas também seus constantes usuários para que possa facilitar suas vidas cotidianas. Hoje já existe um aplicativo que funcione assim o Uber mas nela há corrida com o mototaxista é opcional, ele mostra a opção dos carros e das motos, e você decide qual a melhor viagem para você. Funciona exatamente como o de carro você faz o pedido de corrida para o motorista disponível mais perto de você, e tem chat para que possam conversar, o aplicativo desenvolvido tem a mesma função, mas não terá a opção do carro já que é exclusivo para mototaxistas, mas sua funcionalidade será muito parecida já que também chamara o motorista disponível mais perto além do chat de conversa. Suas diferenças vão ser principalmente na parte visual já que a estética dos aplicativos é diferente, e o que está sendo programado agora tem trabalhadores específicos.



DEPOIS COLOCRA IMAGEM DO APLICATIVO APAR COMPARAR AQUI!!!!!!!!!

* 1. Objetivo geral

O objetivo deste trabalho foi tentar reduzir as dificuldades de comunicação entre os usuários de mototáxi descritas anteriormente, provendo ou melhorando a intercomunicação entre o cliente que necessita de algum tipo de serviço prestado pelos mototaxistas e os trabalhadores da categoria. Para isso foi desenvolvido um aplicativo para dispositivos móveis que possibilite a solicitação de corridas ou serviços em geral aos mototaxistas, que será feita pelo mototaxista mais próximo de sua localização atual e este poderá ou não aceitar a corrida, e pode conversar com seu cliente pelo chat do próprio aplicativo. Pensando nos mototaxistas, os objetivos eram criar um ambiente para que ele possa melhorar os seus meios de trabalho além de lhe dar a opção de aceitar ou não um de chamado para corrida. O piloto poderá visualizar todo histórico de corridas direcionadas a ele, inclusive as corridas que não aceitou. E pensando nos clientes, o objetivo era diminuir as dificuldades de achar um mototaxista livre, pois o aplicativo faria um pedido de corrida para o motorista mais próximo de sua localização atual, além de dar mais segurança para o usuário uma vez que os mototaxistas que serão cadastrados no sistema serão avaliados para garantir a melhor corrida possível e uma ótima experiência para o usuário além disso tera o chat do próprio aplicativo que irá ajuda na comunicação entre o usuário e o mototaxista ajudando ambos.

1.2 Objetivos específicos

Atrelado com o que foi escrito acima podemos citar mais alguns fatores que ajudarão os usuários ao procurarem seus mototaxistas, com o fato de o aplicativo chamar o motorista mais próximo nos evitamos que muitas pessoas fiquem horas nos pontos de mototáxis esperando um livre para poder atendê-lo, também evitamos as constantes mensagens em aplicativos moveis como o WhatsApp, onde pessoas que não tem contato de mototaxistas ficam em grupos, procurando contatos de pessoas que executam o serviço de motorista para atendê-las, e ficam dependentes dessas redes para encontrar um mototaxista ou no próprio Google onde muitos colocam seus contatos para serem chamados, mas procurar os trabalhadores assim não traz nenhum segurança para o usuário já que não tem nenhum tipo de histórico, quando poderiam facilmente encontrar um pelo aplicativo e com proteção garantida, que tera no aplicativo desenvolvido, ganhando assim muito tempo que muitas pessoas perdem ao longo dos dias nessa procura por uma corrida. O aplicativo têm a arquitetura da aplicação na qual e possível observar que o servidor faz a ligação entre o cliente e o mototaxista através do serviço em nuvem. O servidor recebe as informações de cada requisição feita pelos usuários, trabalha em cima das regras de negócio em conjunto com o banco de dados e faz o retorno contendo os dados de resposta.

2 FUNDAMENTAÇÃO TEÓRICA

Aplicações Híbridas:

Devido ao tipo de programação utilizada para desenvolver aplicativos nativos como Android, IOS e Windows Phone é necessário que haja uma compilação em cada uma destas plataformas. Um aplicativo desenvolvido de forma híbrida utiliza linguagens que podem ser interpretadas pelo navegador, isso possibilita sua execução em diferentes plataformas. Os aplicativos híbridos possuem uma limitação de acesso aos recursos de hardware, o sistema operacional ainda não pode ser acessado diretamente por eles e possuem alguns problemas de compatibilidade entre os navegadores. Apesar disso, a cada novo lançamento estes problemas são amenizados tornado-os então uma opção de tecnologia nas empresas. A estrutura de desenvolvimento de aplicações híbridas utilizam linguagens de desenvolvimento *web* possibilitando os testes no navegador. Apos a construção, é possıvel gerar instaladores nativos para várias plataformas. As ferramentas *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), Javascript, AngularJS, Ionic e CORDOVA podem conceder aplicações para facilitar o dia a dia dos seus usuários. Ser mostrado a seguir, algumas dessas tecnologias que foram empregadas neste projeto.

Java: Java foi laçada pela Sun Microsystems em 1995 sendo uma linguagem de programação e plataforma computacional que hoje pertence a Oracle. Existem várias aplicações e sites que dependem de Java instalado na máquina para funcionarem e mais desses são projetados todos os dias. Java tem como caraterísticas a rapidez, segurança e confiabilidade estando em vários lugares como notebooks, celulares, videogames, sites a supercomputadores científicos. O Java é a base para praticamente todos os tipos de aplicações em rede e o padrão globa para o desenvolvimento e distribuição de aplicações moveis e incorporadas, jogos, conteúdo baseado na Web e *softwares* corporativos. O Java foi feito para a mais ampla variedade possível de plataformas de computação para permitir o desenvolvimento de aplicações portáteis com um desempenho superior. Quando as aplicações são feitas em ambientes heterogêneos, as empresas podem aumentar a produtividade, fornecer mais serviços e melhorar a comunicação com o usuário final. O Java tornou-se fundamental para os desenvolvedores, permitindo que eles escrevam um *software* em uma plataforma e o executem em qualquer outra plataforma por meio da máquina virtual Java, crie *softwares* robustos e eficientes para smartphones, gateways, microcontroladores, processadores remotos, sensores, modulos sem fio e praticamente qualquer outro aparelho eletrônico. A Oracle tem dois produtos que implementam *Java Standard Edition* (JSE) que são Java SE *Development Kit* (JDK) e Java SE *Runtime Environment* (JRE). O JRE fornece as bibliotecas, a JVM, e outros módulos necessários para executar *softwares* escritos em Java, já o JDK contem tudo que está na JRE e mais algumas ferramentas, como os compiladores e depuradores necessários para o desenvolvimento de aplicativos Java.

Interface EntityManager: A EntityManager é usada no processo de persistência de dados de um contexto dentro de uma aplicação. Cada um destes contextos é dado como um conjunto de instâncias de entidades, cada instância referencia a uma única entidade. Os ciclos de vida das entidades são geridos dentro do contexto de instâncias e podem ser criados ou removidos pela EntityManager de acordo com seu identificador. A unidade de persistência descreve o grupo de todas as classes que são usadas na aplicação e que devem ser mapeadas de acordo com o banco de dados. Dentro deste cenário de persistência existe a *Java Persistence Query Language* (JPQL) que e uma linguagem de consulta dinâmica usada para buscar ou enviar dados ao banco de acordo com os atributos das entidades.

Intel XDK: O Intel XDK é um *Integrated Development Environment* (IDE) que combina tecnologias voltadas para o desenvolvimento *web* como por exemplo HTML, CSS, JavaScript e Cordova apresentando um ambiente em que é possível desenvolver usando o modo de clicar e arrastar para a geração das telas ou via código fonte. O modo de código permite alterar vários atributos dos componentes como cor, largura, responsividade ou ações de acordo com a necessidade da aplicação. Este IDE permite também emular o aplicativo em desenvolvimento disponibilizando vários dispositivos virtuais com diferentes tamanhos de telas. Usando-se as tecnologias de programação para *web*, é possível construir os módulos do *software* com uma programação mais intuitiva e compacta. Como o Intel XDK possui as ferramentas necessárias para se trabalhar com aplicações híbridas que podem ser executadas em vários sistemas operacionais, ele foi usado para a construção da aplicação moveis.

Cordova: O Apache Cordova disponibiliza um conjunto de APIs que auxiliam no desenvolvimento de uma aplicação com HTML, CSS e JavaScript gerando um pacote de instalação de aplicação móvel nativa. A aplicação é executada no aparelho móvel e tem acesso as funcionalidades nativas do dispositivo, como Giroscopio, GPS ou câmera. Cordova. Todo o código fonte da aplicação fica localizado em *web App*. A aplicação é desenvolvida como se fosse uma página da *web*, como padrão, tem um arquivo principal que funciona como um ponto de partida para iniciar o aplicativo, chamado *index.html*, nele contem os links e caminhos das bibliotecas de JavaScript, CSS, AngularJS e todos os outros recursos necessários para que tudo funcione. O *software é* executado dentro de um *WebView* que por sua vez, fica dentro de um aplicativo nativo o qual pode-se gerar, por exemplo, um pacote de instalação do Android. Este aplicativo nativo, funciona como um recipiente e tem um arquivo chamado *config.xml*, e este XML que fornece dados sobre o aplicativo e dita os parâmetros que afetarão o funcionamento do *software*, por exemplo, uma mudança de orientação. O *WebView* do Cordova fornece toda a interface de usuário da aplicação e pode também, estar dentro de outro aplicativo maior como se fosse um modulo de um *software nativo. O* Cordova contem um núcleo de *plugins* que oferece ao aplicativo acesso aos recursos nativos do aparelho, como giroscópio, GPS, câmera ou contatos. Os *plugins são* muito importantes dentro deste contexto do Cordova. São eles que fornecem as conexões entre o Cordova e os componentes nativos provendo a comunicação entre eles. Isso permite que o desenvolvedor chame uma função nativa através de JavaScript. Além dos *plugins* deste núcleo, existem também os *plugins* de terceiros que fornecem conexões para componentes que não fazem parte dos padrões de todas as plataformas, ou seja, os componentes personalizados.

HTML: HTML é uma linguagem de marcação de hipertexto que pode ser resumida como algum conteúdo que está alocado dentro de uma página da *web* e que tem como característica principal a alternativa de se conectar com outras páginas *web*. HTML faz uso de *tags* que são marcadores que descrevem uma página web. Essas *tags são* elementos descritos entre *<>*, que posteriormente, são interpretados pelos browsers que exibem as páginas formatadas com elementos visuais, essas páginas, podem conter vários componentes de mídia como textos, vídeos, imagens ou áudios. O HTML foi desenvolvido por Tim Berners-Lee em 1989 no *European Organization for Nuclear Research*, também conhecido por CERN. O aparecimento do HTML5 alterou muitos quesitos no mundo do desenvolvimento para web, com novos componentes, novas funcionalidades e várias outras novidades que permitem melhor usabilidade e interações que antes eram apenas expectativas de alguns programadores desta área. Mesmo com algumas dessas novas funções ainda em processo de determinação, já se pode aproveitar diversas novidades que o HTML5 trouxe consigo. O HTML5 sozinho, não pode controlar uma aplicações *web*, como e apenas para marcação dos elementos de uma página precisa do JavaScript e do CSS para implementar a apresentação e interação do *front-end*.

CSS: O CSS e uma linguagem aplicada na formatacao, apresentação e padronização de componentes de alguma página escrita em HTML. Ele viabiliza a organização entre os conteúdos das páginas. Assim como o HTML, ele não necessita de nenhum outro *software além* do navegador para ser utilizado. Com o surgimento do CSS3, vários padrões e imagens que antes eram usados para adicionar cores e formas aos componentes HTML puderam ser substituídos, o que não era possível apenas com CSS. Outro fator importante foi que os tamanhos dos documentos HTML tiveram uma redução significativa visto que menos códigos eram escritos. O desprendimento de dependências externas elevou o desempenho dos sites e a flexibilidade de uso das propriedades permite várias combinações de estilos diferentes.

JavaScript: A linguagem JavaScript é multi-plataforma, orientada a objetos e geralmente e utilizada com alguma outra tecnologia, como HTML5 nos casos de aplicações *web*. Ele implementa um grupo de objetos como Dates, Arrays, operadores, estruturas de repetição e de controles de fluxo. O JavaScript e inspirado em ECMAScript e normalmente e usado dentro das aplica coes do lado do cliente em validações de formulários e *inputs*, mas também pode ser usado nos servidores como manipulador de ficheiros ou como conexão com bases de dados. As qualidades mais marcantes dessa linguagem são sua sintaxe estruturada e dinamicidade dos objetos pois não se associam as variáveis e sim aos seus valores. As funções são consideradas como objetos e podem ser usadas como variáveis, passadas como parâmetros ou retornadas de alguma outra função. JavaScript é uma linguagem que normalmente é usada em conjunto com HTML e CSS, sendo ele, a parte de regras de negócio, interação e finalização de *design*.

Jquery: O jQuery é um *framework* que facilita a maneira de programar utilizando a linguagem JavaScript. Ele é compatível com a maioria dos navegadores e é capaz de gerar animações, manipular HTML e eventos. Sendo um dos pioneiros a trabalhar com JavaScrip ele revolucionou a maneira de programadores usarem esta linguagem.

AngularJS: Como dito por Branas (2014) o AngularJS é um *framework open source* usado na criação de projetos *web* utilizando a linguagem de programação JavaScript. Ele permite fazer a estruturação da aplicação em camadas melhores definidas, a criação de componentes reutilizáveis e já possui toda a estrutura necessária para integrar-se com *back-end* via Ajax não necessitando de alguma outra biblioteca para isso. Foi criado em 2009 por Misko Hevery e Adams Abrons com a intenção de facilitar a criação de plicacoes *web* focando a redução de código fonte e melhorias na produtividade. Hoje em dia, o AngularJS é mantido pelo Google e tem uma grande comunidade de desenvolvedores e fóruns. Um ponto forte do AngularJS é o *Two-way Data Binding* que capta qualquer alteração de dados feita na variável *$scope* (versão 1.x) a qual faz a ligação entre a *view* e o *controller* da aplicação facilitando a mudança de estado dos componentes.

Banco de dados MySQL: Um banco de dados é um conjunto de informações estruturadas, ele pode ser desde uma lista de compras até uma grande quantidade de informações de negócios de uma empresa. Para acessar, editar, remover ou processar informações de um banco em alguma máquina, ha necessidade de um *software* que gerencie todas essas requisições como o Servidor MySQL. Os computadores são bons em processar uma grande massa de dados, nesse ambiente, o gerenciamento de bancos se torna uma peça fundamental para que tudo funcione corretamente. O banco de dados a ser usado para armazenar todas as informações de corridas e cadastros será o MySQL. O MySQL é um servidor de banco de dados *Structured Query Language* (SQL) multiusuário, multitarefas, robusto e muito rápido. MySQL, o mais popular sistema de gerenciamento de banco de dados SQL *Open Source*, é desenvolvido, distribuído e tem suporte da MySQL AB. O MySQL é um banco de dados relacional, ou seja, ele armazena os dados em tabelas separadas e depois as relaciona quando necessário proporcionando uma maior velocidade e flexibilidade. Ele é *Open Source*, isso quer dizer que qualquer pessoa pode usá-lo sem pagar nada. O Servidor MySQL foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta demanda por diversos anos de maneira bem-sucedida. O MySQL é um banco de dados relacional bastante popular no mundo inteiro[...].

Webservices: Os *webservices são* utilizados para fazer a integração e troca de mensagens entre sistemas, podendo estes, realizar chamadas de outros sistemas a fim de obter dados. Os dois formatos mais conhecidos de troca de informações são *eXtensible Markup Language* (XML) e o *JavaScript Object Notation* (JSON). Os *webservices* permitem acessar servicos de uma forma padronizada independente da linguagem o que acaba sendo uma grande vantagem, um exemplo disso é que se o seu programa for escrito em Java, ele pode trocar informações com sistemas feitos em C#, PHP, Python ou qualquer outra linguagem. SOAP e REST são duas das várias formas de criar *webservices* e também são as mais usadas. (LECHETA, 2015). Loudon (2010) disse que a internet que conhecemos hoje é muito diferente daquela de dez anos atrás pois a complexidade das aplicações *Web* aumentou muito e as mudanças acontecem a todo momento, isso dificulta muito a manutenção das mesmas. Ele cita sete boas praticas de desenvolvimento que tem que ser observadas durante o ciclo de vida das aplicações. Disponibilidade contínua: sugere que a maior parte das grandes aplicações *Web* devem rodar o tempo todo, sem pausas. Além disso, todas as repostas têm quer ser em tempo real e a todo momento, mesmo nos instantes em que está ocorrendo vários acessos simultâneos. Desenvolvedores *Web* tem que projetar sistemas que sejam muito robustos. Grande base de usuários: o autor aponta que Grandes aplicações *web* normalmente apresentam volumes impressionantes de usuários. Isso requer o gerenciamento de um grande número de conexões simultâneas ou de várias camadas de *caching*. Entrega em partes: ao contrário de aplicações que são entregues como um pacote fechado de módulos, aplicações *Web* muitas vezes são entregues modulo a modulo e mantendo sempre a disponibilidade dos serviços que já foram entregues. Diversidade: como existem várias possibilidades de desenvolvimento como aplicações financeiras, mecanismos de busca ou site de notícias, os desenvolvedores devem se preocupar em escrever códigos genéricos que possam ser reutilizados em várias situações diferentes. Longevidade: mesmo as aplicações *web* que já existem ha anos ainda estão no começo de seu ciclo de vida. Desenvolvedores *Web* tem de escrever códigos que sejam capazes de se manter aplicáveis mesmo apos anos de alterações e manutenção. Múltiplos ambientes: a *Web* e um ambiente muito heterogêneo e com mudanças rápidas, existem vários navegadores e dispositivos diferentes que não oferecem tanto suporte. Usuários buscam várias aplicações a partir de inúmeros ambientes e com todo tamanho de tela inclusive smartphones. Os desenvolvedores devem se preocupar em tratar o maior número possível de situações diferentes. Atualiza¸coes em tempo real: as grandes aplicações *Web estão* o tempo todo sofrendo modificações e evoluindo, a todo momento, várias alterações são enviadas aos servidores, cada módulo deve acompanhar essas mudanças.

Webservices RESTful: *Representational State Transfer* - Transferência de Estado Representacional (REST) é um conjunto de técnicas e padrões para desenvolver *webservices* com a maioria dos conceitos baseados nos metodos *GET*, *POST*, *PUT*, e *DELETE* que são do protocolo *HyperText Transfer Protocol* (HTTP). O RESTful nada mais é do que os *webservices* que sao criados seguindo este padrão. (LECHETA, 2015).

Principais comandos do RESTful: No REST o metodo *GET é* normalmente usado para fazer consultas e requisições que usam poucos parâmetros, uma requisição do tipo *POST é* geralmente usada caso o objetivo seja inserir um novo registro. De acordo com as boas praticas de programação do REST, deve-se utilizar o *PUT* quando pretende-se atualizar algum dado e para finalizar o *Create, Read, Update e Delete* - Criar, Pesquisar, Atualizar e Remover (CRUD) deste padrão de desenvolvimento, usa-se o *DELETE* quando se quer remover registros.

Padrão de comunicação JSON: O JSON representa a estrutura de dados de um objeto JavaScript. O seu formato é mais simples e leve se comparado com o XML, por isso vem substituindo-o como forma de troca de informações entre sistemas. (LECHETA, 2015). Fonseca e Simoes (2007) apontaram nos seus estudos que nos últimos anos o *eXtensible Markup Language* (XML) tem tomado conta do mercado em várias áreas desde finanças a agricultura. Ele tem sido a forma de comunicação escolhida na maior parte das aplicações. Jogadores salvam seus progressos em XML, torcedores recebem resultados do seu time via XML, são muitas aplicações que usam esta tecnologia. O XML é uma das sintaxes mais robustas e confiáveis já criadas, mas também tem seus pontos fracos. A sua sintaxe é cheia de palavras redundantes o que pode afetar o entendimento humano e obriga as aplicações a terem um espaço de armazenamento a mais, este fator também atrapalha quando a largura de banda de transmissão é baixa. Construir um paragrafo em XML não é tao simples, tem que se colocar vários elementos aninhados e fazer testes para detectar dados mal formatados. O XML não possui tipos de dados complexos como número de ponto flutuante, por exemplo, ele reconhece 3.14159 como uma palavra de sete caracteres e não um número. O conjunto de caracteres usados para escrever uma sentença XML são particularmente difíceis de se usar em um teclado convencional, isso dificulta a rápida criação de documentos. O uso de editores próprios pode amenizar este problema. Existem várias tecnologias no mercado aptas a substituir o XML, mas duas delas têm se destacado das demais, a *Yet Another Markup Language* (YAML) e o JSON. (FONSECA; SIMOES, 2007). O JSON consegue codificar e decodificar quatro tipos primários, sendo eles, strings, números, booleanos e nulos e também objetos e vetores. Cada objeto é uma coleção de zero ou mais pares representando nome/valor de forma não ordenada sendo, o nome uma string e o valor qualquer um dos quatro tipos primários citados anteriormente ou um objeto ou um vetor, Assim o JSON foi desenhado com o objetivo de ser simples, portável, textual, e um subconjunto do JavaScript. (FONSECA; SIMOES, 2007).

Ionic: Ionic é um [framework](https://www.treinaweb.com.br/blog/para-que-serve-um-framework/) para criações de aplicações móveis, além de ser um facilitador no desenvolvimento de diversas aplicações e, sem dúvida, sua utilização poupa tempo e custos para quem utiliza, pois de forma mais básica, é um conjunto de bibliotecas utilizadas para criar uma base onde as aplicações são construídas, um otimizador de recursos ele permite ao desenvolvedor focar nos “problemas” da aplicação , e ajuda as equipes a criarem e distribuírem belos aplicativos híbridos em várias plataformas, possuindo foco na experiência do usuário ou interação do usuário com o aplicativo.

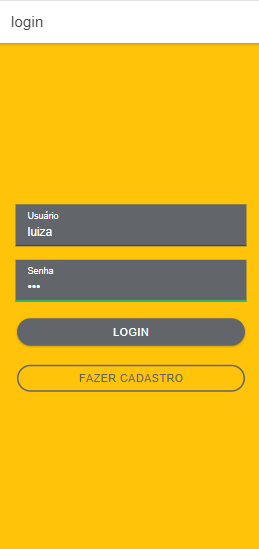
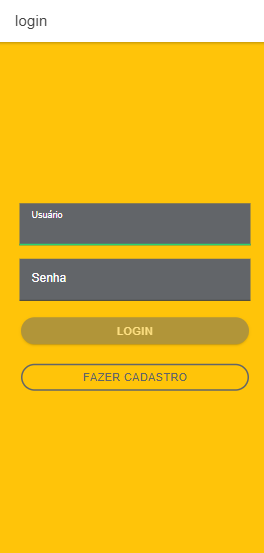
Consideracoes Finais:

As tecnologias citadas anteriormente foram entendidas para formar o conhecimento básico sobre o assunto de aplicações moveis e as ferramentas necessárias para conectar dispositivos via servidor. Para este projeto, as ferramentas utilizadas na construção de aplicações híbridas se mostraram mais eficazes do que as de aplicações nativas enquanto desenvolvimento, elas foram combinadas e podem ajudar os mototaxistas a amenizar a questão de comunicação com seus clientes permitindo assim atenderem o maior número de passageiros e da forma mais eficiente possível.

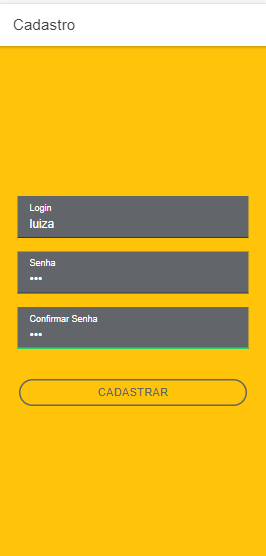
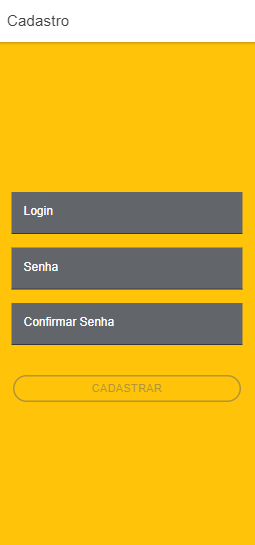
* 1. METODOLOGIA

As ferramentas utilizadas serão o Trello que é uma ferramenta de quadro virtual gratuita usada para o gerenciamento de projetos. Essa é uma ferramenta que fornece auxilio visual para organização da agenda do projeto deixando mais aparentes atividades adiantadas e atrasadas, sendo uma ferramenta flexível de gerenciamento de trabalho em que os times podem criar planos, colaborar em projetos, organizar fluxos de trabalho e acompanhar o progresso com visualização, produtividade e gratificação. Da troca de ideias ao planejamento e à execução, o Trello gerencia os grandes marcos e as tarefas diárias de trabalhar em conjunto e cumprir os objetivos; o DIA que é uma ferramenta de modelagem de código aberto, que possibilita a criação de diagramas UML, mapas de rede e fluxogramas, ele tem como principais recursos um sistema integrado de bibliotecas de símbolos e um pequeno grupo de ferramentas gráficas primitivas para textos, caixas, elipses, polígonos e curvas de bézier. É possível criar gráficos vetoriais “do zero” no Dia, mas a interface favorece o uso dos símbolos que acompanham o programa e a conexão entre eles usando as linhas em zig-zag; a programação do aplicativo será usado o Cordova que é o projeto aberto base de criação de aplicativos híbridos para mobile. Ele utiliza tecnologias web (HTML, CSS, JS) para criar aplicações para celulares e outros dispositivos, mas que não rodam em um browser normal como um site. Os recursos do dispositivo ficam acessíveis, ao contrário de um site normal. Também será utilizado o Ionic- que é uma extensão do Cordova usando [AngularJS](http://ionicframework.com/) preferindo [TypeScrip](https://pt.stackoverflow.com/q/15998/101) do que JavaScript. Isto permite que a formatação visual seja muito mais rica e facilitada, seu foco é a UI tentando abstrair as diferenças de cada plataforma e não cuidar do acesso aos recursos do dispositivo, esse framework além de ser um facilitador no desenvolvimento de soluções digitais, ajudando a otimizar processos, recursos e tempo ele pode ser usado para diversas finalidades e, no caso do Ionic, o seu propósito é servir de plataforma para a criação de [aplicativos híbridos](https://mundodevops.com/blog/aplicativos-hibridos/). O Angular é um [framework](https://www.treinaweb.com.br/blog/para-que-serve-um-framework/) [JavaScript](https://www.treinaweb.com.br/blog/o-que-e-e-como-comecar-com-javascript/) de código aberto mantido pela Google para a construção de [SPA](https://www.treinaweb.com.br/blog/o-que-sao-aplicacoes-spa/) (sigla para *Single Page Applications* ou *Aplicações de Página Única*). Resumidamente, uma [SPA](https://www.treinaweb.com.br/blog/o-que-sao-aplicacoes-spa/) é basicamente uma aplicação web construída em uma só página, na qual a interação e a navegação entre as sessões de uma página ocorrem de maneira a qual não é necessário recarregar a página em cada uma dessas mudanças. Para a construção da aplicação móvel foi usada a IDE Intel XDK que é capaz de construir aplicações híbridas, este aplicativo tem um foco mais voltado para a plataforma Android. Todas as telas foram crias usando as tecnologias HTML, CSS e Ionic, para as funcionalidades delas foram usados AngularJS e JavaScript e jQuery. O AngularJs gerencia a transição de informações entre as telas e o *controller através* do *$scope* que é injetado na criação do modulo da aplicação *angular.module("myApp").controller(...)*, neste controle ha injeção de todas as dependências usadas para gerar notificações, mensagens, criar filtros de dados entre outros. O *$scope é* responsável por guardar funções e variáveis e é atualizado pelo *ngBind* sempre que há algum evento na tela ou no *controller*.

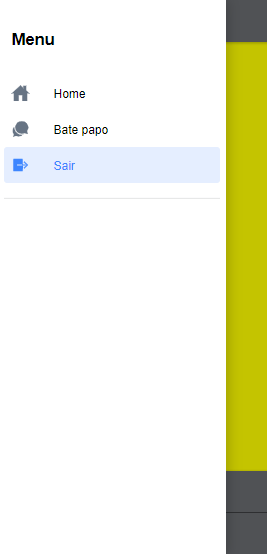
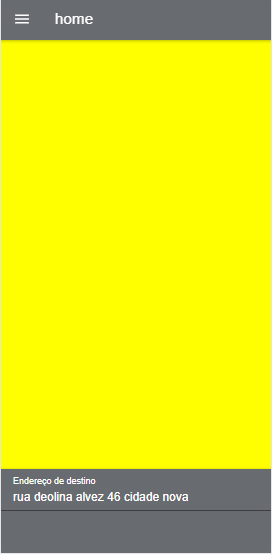
As figuras abaixo mostram a tela de login do aplicativo, onde temos a opção de nos cadastrar ou fazer o login para entramos no aplicativo. Para fazer o login já deve existir os dados cadastrado, caso não tenha, terá a opção de se cadastrar Para entrar no sistema. O login só poderá ser efetuado assim que forem inseridos dados do usuário e a senha, caso isso não ocorra o botão não é liberado, como mostrado na imagem abaixo, além disso se forem colocados dados inválidos aparecera uma notificação de erro.



Caso o usuário deseje fazer seu cadastro, devera clicar no botão “fazer cadastro” que irá o redirecionar para a tela de cadastro como mostra a imagem abaixo. Nessa tela o usuário também não consegue se cadastrar caso esteja algum campo vazio, como demostrado na figura. Assim que o cadastro é realizado o usuário é redirecionado para a tela de login, onde ele devera entrar com os seus dados cadastrados que terminou de efetuar.



Com tudo isso feito o usuário entra na tela home onde ele digitara o endereço para onde deseja ir, e clicara no botão para procurar um mototaxista disponível, para poder fazer uma corrida como demonstrada na imagem abaixo. Este pedido de corrida vai ser feito para o motorista mais próximo de sua localização atual onde este recebera a notificação já com o destino descrito anteriormente e este poderá ou não aceitar a corrida. No canto superior esquerdo podemos perceber um menu que quando clicado vemos que nele se tem as opções de podemos ir para a tela home, abir um chat de conversas, podemos sair do aplicativo e voltara para a tela de login, e para os mototaxistas temos a opção de ver o seu histórico de corridas, as aceitas e as negadas para melhor controle de suas viagens.



Caso necessite o usuário poderá conversar com seu mototaxista pelo chat, o qual é interligado ao firebase para poder haver essa comunicação como mostrado na imagem abaixo.



5 CRONOGRAMA

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Etapa | Descrição | Mês | | | | | | | | | | |
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 |
| 1 | Avaliar as soluções existente no mercado | x | x |  |  |  |  |  |  |  |  |  |
| 2 | Modelar a solução a ser criada, junto ao estudo e revisão sistemática da literatura (RSL) |  | x | x | x | x |  |  |  |  |  |  |
| 3 | Experimentos e testes iniciais |  |  |  |  | x | x | x | x |  |  |  |
| 4 | Correções e teste finais em produção |  |  |  |  |  |  | x | x | x |  |  |
| 5 | Escrita do TCC |  |  |  |  |  |  |  |  |  | x | x |

REFERÊNCIAS

Insira aqui referências conforme normatizado pela ABNT ou SBC. Sugiro o uso de geradores de referência como:

* O que é Cordova? Disponível em: https://pt.stackoverflow.com/questions/236145/qual-%C3%A9-a-diferen%C3%A7a-entre-o-apache-cordova-e-o-ionic
* O que é o Trello?. Disponível em: https://trello.com/about.
* O que é o Trello?. Disponível em: <https://trello.com/pt-BR/about>
* O que é o DIA?. Disponível em: https://www.hardware.com.br/dicas/dia.html
* O que é Angular. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-o-angular-e-para-que-serve>
* O que é Ionic. Disponível em:https://pt.stackoverflow.com/questions/236145/qual-%C3%A9-a-diferen%C3%A7a-entre-o-apache-cordova-e-o-ionic
* O que é Ionic. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-ionic>
* CORDOVA. Documentation: Apache Cordova. Disponıvel em:*<*<https://cordova.apache.org/docs/en/latest/>*>*
* JQUERY (New York). **What is jQuery?:** jQuery API. 2016. Dispon´ıvel em:*<*<https://jquery.com/>*>*
* Java Disponível em: *<*<http://goo.gl/pgV7xF/>*>*
* Alternativas ao XML: YAML e JSON Disponível em: *<*<http://goo.gl/5VOl7S>*>*
* Ajuda do Google Maps Disponível em: *<*<https://support.google.com/maps/?hl=pt-BR>#*>*
* What is jQuery? Disponível em: *<*<https://jquery.com/>*>*
* HTML 5: A LINGUAGEM DE MARCA ¸CAO QUE REVOLUCI- ~ ONOU A WEB. 2. ed. S~ao Paulo: Novatec, 2014. 330 p