

# Metodologia e Desenvolvimento

## 4.1 Considerações Iniciais

Para a construção de um software espera-se a existência de um roteiro, um conjunto de passos a se seguir para que o objeto de trabalho venha a alcançar a qualidade desejada. Todas as funcionalidades devem aproximar-se ao máximo das necessidades encontradas dentro do meio de atuação ao qual os usuários se encontram.

A metodologia utilizada para a criação deste software é baseada no arcabouço<sup>1</sup> genérico de processos em que, basicamente as atividades são divididas em cinco etapas principais: comunicação, planejamento, modelagem, construção e implantação.

## 4.2 Comunicação

Uma reunião com dois representantes da categoria dos mototaxistas e dois potenciais passageiros apontou as dificuldades apresentadas em seus cotidianos como por exemplo, a ineficácia de uma linha de telefone fixo nos pontos de mototáxi e a falta de aplicativos para dispositivos móveis que operem na região de Formiga - MG que ajudem a localizar pilotos.

Descrição dos participantes da reunião: o primeiro representante trabalha há 4 anos no ramo de mototáxi e está atualmente ativo na profissão, o segundo representante trabalhou por 3 anos no setor e não exerce a profissão atualmente. Os dois passageiros já fizeram uso do serviço de mototáxi, o primeiro cliente usa com frequência e o segundo usa apenas esporadicamente.

### 4.2.1 Descrição do Problema

Os mototaxistas nem sempre trabalham apenas com corridas ou viagens, muitos se dispõem a fazer outras prestações como serviços de banco ou entregas de produtos para lojas que não

<sup>1</sup>O conjunto do que é necessário para uma construção.

possuem *motoboy*, eles não têm um único meio de divulgação para todos estes tipos de serviços. Alguma observação como formas de pagamentos e horários de trabalho geralmente ficam fixos na impressão dos cartões o que não apresenta agilidade de mudanças.



Figura 4.1: Forma de pagamento aceita - Fonte Silva (2015)

A figura 4.1 e a figura 4.3 exemplificam os problemas de apresentação dos mototaxistas descritos anteriormente, os tipos de serviços e formas de pagamento estão fixos e não possibilitam alterações instantâneas.

Um problema apontado pelos representantes foi a dificuldade no gerenciamento de atendimentos que têm horários próximos visto que eles precisam atender a várias ligações sequenciais, isso demanda um tempo pois alguns clientes demoram a confirmar os dados da corrida.

Não existe nenhum processo automatizado para acesso aos históricos das corridas efetuadas por período, de acordo com os dados obtidos no levantamento, o piloto normalmente não sabe sobre as corridas efetuadas durante o decorrer de uma semana.

Apesar dos cartões obterem os horários de trabalho do piloto muitos clientes ligam fora dele requerendo algum serviço. Geralmente, após cadastrarem o número do mototaxista no celular, as pessoas se desfazem ou perdem os cartões que contém esses tipos de dados.

Segundo os passageiros, há dificuldades de encontrar mototaxistas disponíveis sem a ajuda de alguém que já possua um número ou quando se está distante dos pontos. Outro empecilho que se encontra na parte do solicitante é a necessidade de efetuar a ligação. Mesmo com vários sistemas de conversa virtual, o mototaxista é encontrado principalmente via número do ponto ou por ligações para o seu celular.

Os pontos de mototáxi de Formiga - MG geralmente têm uma ou duas linhas de telefone fixo que são divididas entre os pilotos, eles formam uma fila que vai girando na medida que os

chamados são recebidos e o cliente não sabe quem irá atendê-lo no momento da solicitação.

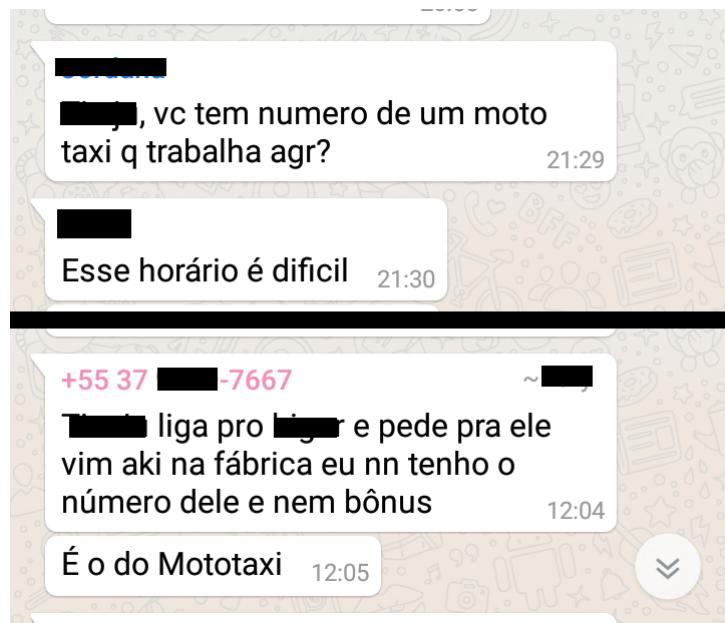


Figura 4.2: Necessidade de Encontrar um Mototaxista

A figura 4.2 exemplifica a dificuldade de encontrar um mototaxista disponível no determinado momento e também o problema da comunicação com o mototaxista visto que o cliente já possui o acesso à internet mas pode não ter créditos para ligar via operadora.

Um questionamento que surge momento que se tem acesso aos contatos dos pilotos é a escolha do mais qualificado para o seu tipo de necessidade. O cliente normalmente tem essa dificuldade quando não possui nenhuma indicação ou embasamento de corridas anteriores de cada mototaxista.

A figura 4.3 mostra os catões de alguns mototaxistas da cidade de Formiga - MG. Estes cartões normalmente não têm capacidade de facilitar a escolha de um mototaxista por parte do solicitante.

Quando o cliente se encontra em algum local que não tem costume de visitar, naturalmente tem alguma dificuldade de se localizar quando se trata do endereço. Alguns pontos de referência podem te ajudar a descrever sua localização atual para que o mototaxista possa buscá-lo, porém há algum constrangimento nesse momento.



Figura 4.3: Cartões de alguns mototaxistas de Formiga - MG

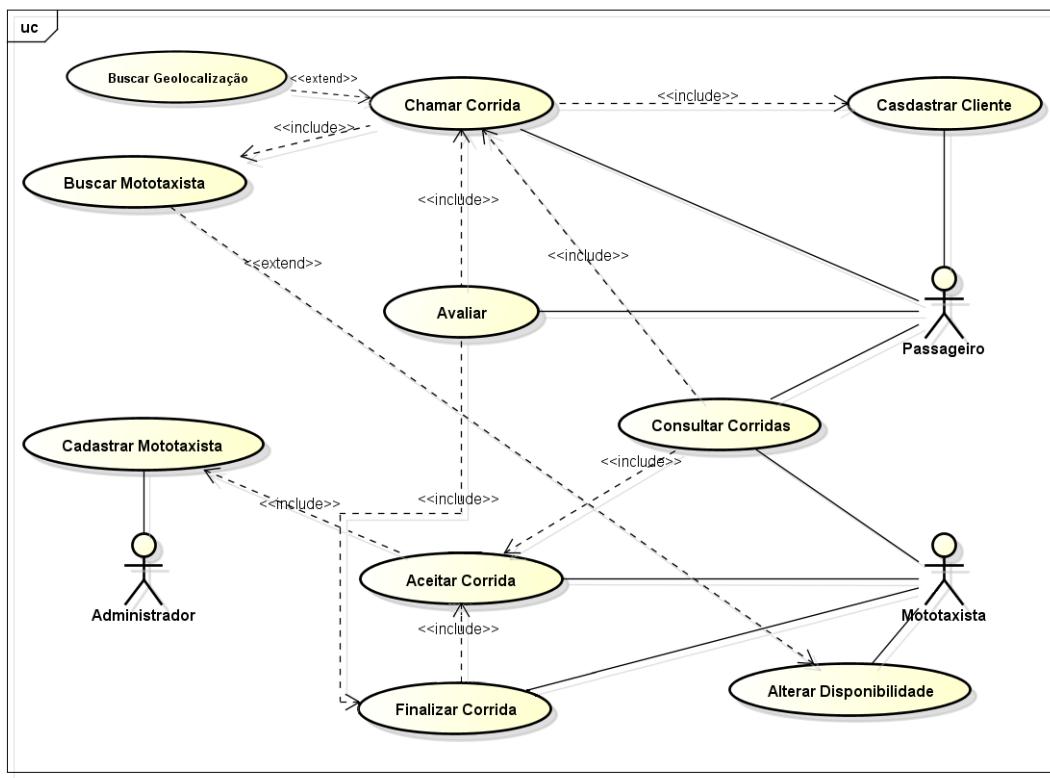
### 4.3 Planejamento

Alguns requisitos foram apontados para que haja uma boa usabilidade no software como, alerta de solicitações de corrida em primeiro plano, possibilidade de criar uma fila com atendimentos simultâneos e controle das corridas efetuadas durante um certo período. Na visão do passageiro, o sistema deve apresentar uma listagem com alguns tipos de organização como, por pontuação, por tempo de resposta e por nome dando-lhe a opção de escolher com qual piloto ele quer efetuar a corrida. Os processos que foram executados ao longo do desenvolvimento levaram em consideração o tempo disponível, e a complexidade do software. Os recursos utilizados compuseram uma série de softwares como a IDE Net Beans e a IDE Intel XDK, banco de dados MySQL, sistema operacional Windows, Android para testar a aplicação, linguagens HTML, Java, Css e Javascript, frameworks AngularJS, jQuery e Ionic. Os riscos foram mínimos visto que não havia uma equipe a ser coordenada e o escopo da aplicação era bem definido, o maior obstáculo foi o prazo de entrega.

## 4.4 Modelagem

O Diagrama de Caso de Uso mostrado na figura 4.4 apresenta as funcionalidades possíveis para cada um dos três atores: Administrador, Mototaxista e Passageiro. O ator Administrador é o responsável por cadastrar os mototaxistas que serão apresentados no uso do aplicativo bem como suas identificações, foto, e tipos de serviços prestados. O ator Mototaxista é capaz de alterar sua própria disponibilidade alternando entre disponível ou indisponível, é capaz também de consultar as corridas que ele efetuou em um determinado período que contém data inicial e data final. Neste relatório ele não é capaz de visualizar as avaliações dos clientes. Ele pode interagir com os chamados realizados para ele escolhendo entre aceitar ou negar uma solicitação de corrida. O Mototaxista poderá ainda finalizar uma corrida que já tenha sido concluída permitindo ao Passageiro avaliá-la.

O ator Passageiro, assim como o Mototaxista, é capaz de consultar em um determinado período com data inicial e data final as corridas que ele solicitou mostrando as que foram aceitas e as que foram negadas pelos mototaxistas, ele será capaz também de avaliar as corridas que foram aceitas dando-lhe a opção de visualizar as pontuações no relatório. No momento em que o Passageiro quer chamar uma corrida, ele tem liberdade de escolher entre os mototaxistas disponíveis no momento da solicitação, em caso de estar em algum local desconhecido ou que não lembre do endereço, ele pode buscar seu ponto atual via Geolocalização facilitando a confirmação do logradouro, pode também estabelecer uma pontuação para o mototaxista referente a corrida que foi finalizada por parte do piloto.



powered by Astah

Figura 4.4: Diagrama de Caso de Uso

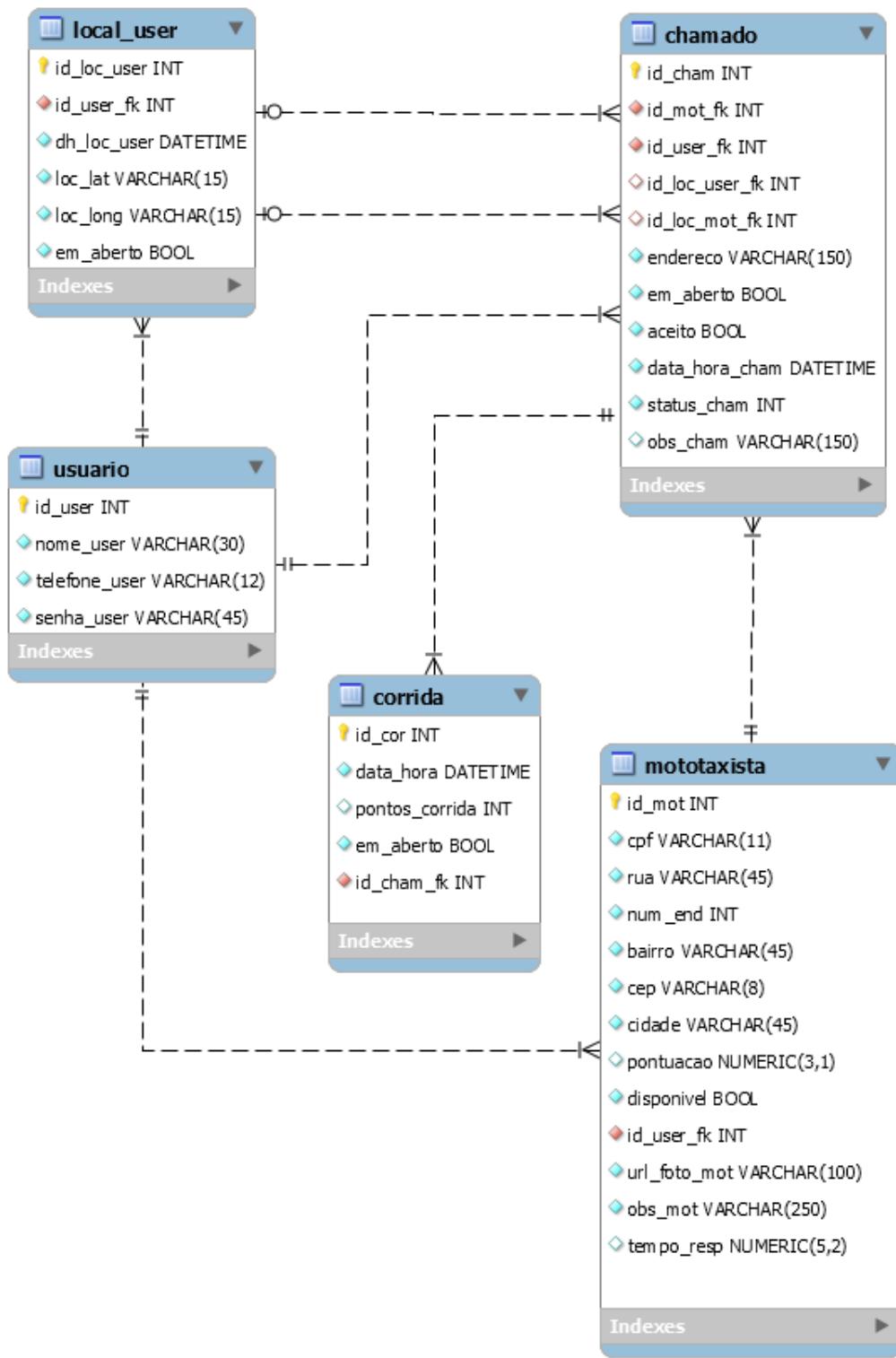


Figura 4.5: Diagrama Entidade Relacionamento

Na figura 4.5 é exibido o Diagrama Entidade Relacionamento (DER) que apresenta como foi montada a estrutura do banco de dados que serve como fonte de dados para o servidor RESTful que faz toda a troca de informações entre o passageiro e o mototaxista.

O DER apresenta cinco tabelas que são gerenciadas pelo banco de dados MySQL, a tabela *usuario* contém o identificador sequencial, o nome que ele quer que apareça nas trocas de solicitações, o número do celular que é tratado como chave única não permitindo a inserção do

mesmo com redundância e que também serve como *login* de acesso no aplicativo e a senha de acesso.

A tabela *mototaxista* apresenta uma estruturação mais elaborada de atributos visto que ele precisa ser melhor gerenciado pelo administrador do sistema. Nela é possível observar o identificador sequencial, o cpf, os dados de endereço, a pontuação que armazena a média total das avaliações feitas pelos seus cliente ao término das corridas, o atributo booleano que permite a visualização do mesmo como disponível para receber solicitações, a chave estrangeira para a tabela de *usuario* que armazena seu nome e dados de login, o atributo *url\_foto\_mot* que guarda a foto do mototaxista que será apresentada aos clientes na listagem dos pilotos e no histórico de corridas, o campo de observação guarda informações como tipos de serviço prestados e opções de pagamento. O atributo *tempo\_resp* guarda a média do tempo que o mototaxista demora para responder os chamados de corridas. Esta média permite a ele alcançar uma melhor localização na listagem dos disponíveis <sup>Justificativa</sup> qualificado ela está ordenada por tempo de resposta.

A tabela *local\_user* é usada para salvar a geolocalização atual do cliente no momento da solicitação de corrida e também conta com o identificador primário sequencial, a chave estrangeira para identificação do usuário, data e hora corrente no momento da localização, os atributos que contém as coordenadas geográficas de latitude e longitude e uma variável booleana que indica se a localização já foi usada ou não.

A tabela de *chamado* é utilizada no momento em que o cliente realiza a solicitação de corrida para algum mototaxista. Ela contém o identificador primário sequencial, as chaves estrangeiras que identificam quem é o cliente que está efetuando o chamado e qual o mototaxista que ele está solicitando. Nesta tabela existem os identificadores estrangeiros de localização do cliente e do mototaxista que não são obrigatórios visto que nem sempre o GPS dos dois estarão ativados. O atributo de endereço guarda a localização atual do cliente no momento do chamado sendo que este pode ser preenchido manualmente sem o auxílio do GPS. O campo *em\_aberto* indica se o chamado está pendente ou finalizado, o atributo denominado como *aceito* indica se o mototaxista aceitou ou recusou o chamado, a tabela disponibiliza também o campo de data e hora da solicitação.

O atributo *status\_cham* é alterado em vários momentos durante o processo de solicitação de uma corrida. Quando o cliente inicia o chamado ele tem o valor *1*, quando o chamado chega para o mototaxista e aciona a notificação no aparelho muda o valor para *2*, no momento em que o mototaxista responde o chamado aceitando ou recusando o valor deste atributo muda para *3*, a resposta do chamado retorna para o cliente, então o valor passa a ser *4* e por fim, quando o cliente confirma a visualização da resposta o atributo recebe o valor *5* que indica que o chamado já passou por todos os processos possíveis dentro do escopo da aplicação. A tabela *chamado* conta com um atributo de observação que pode ser preenchido pelo mototaxista como justificativa em caso de recusa de um chamado, em caso de não preenchimento desta observação o cliente recebe uma mensagem padrão do sistema.

A tabela *corrida* é preenchida no momento em que o mototaxista aceita o chamado. Ela contém o identificado primário sequencial, a data e hora que a corrida foi aceita gerando o

tempo médio de resposta do mototaxista, os pontos que o cliente atribuiu ao atendimento do piloto, o booleano que indica se a corrida está aberta ou finalizada e o identificador estrangeiro para a tabela de chamado o qual provê informações detalhadas da solicitação como um todo.

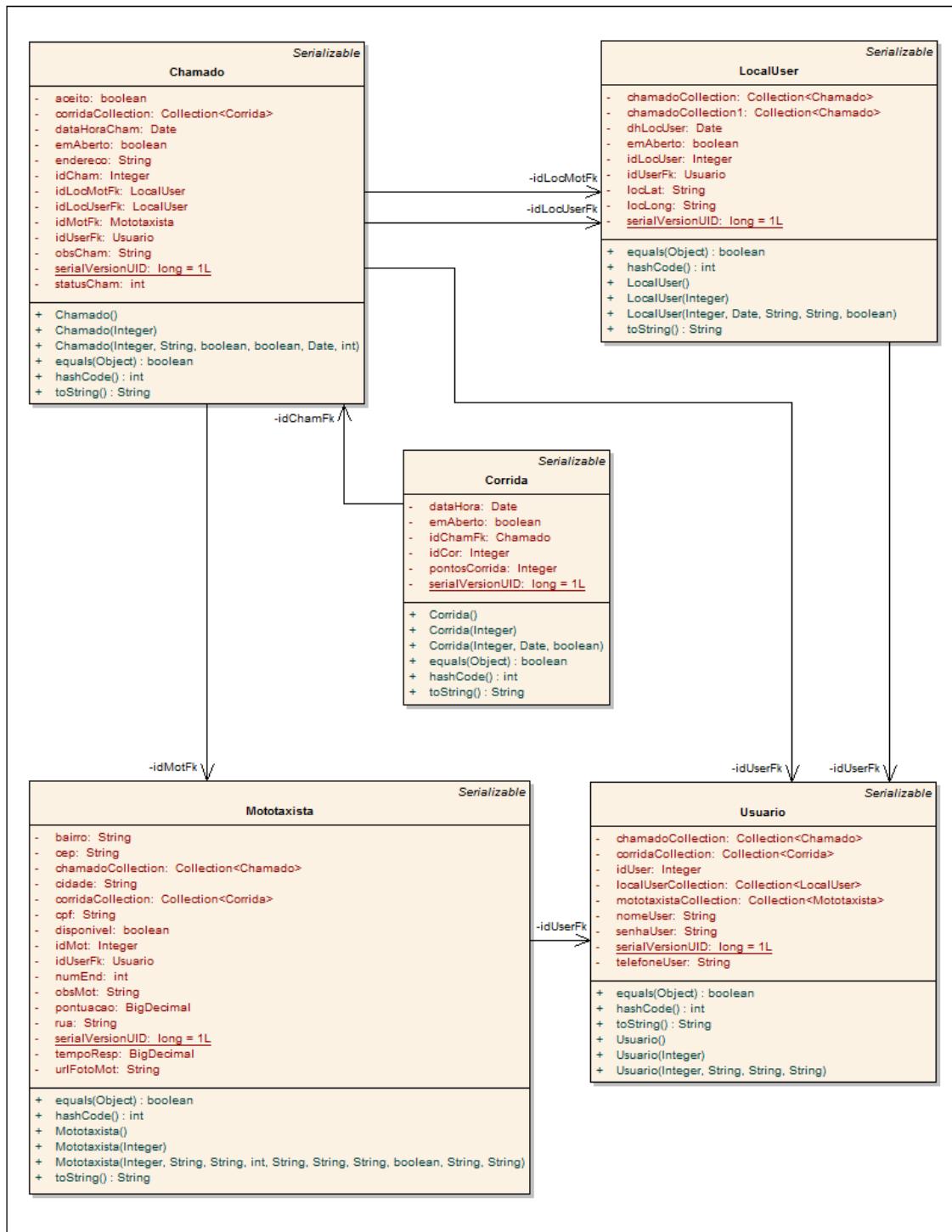


Figura 4.6: Diagrama Classe dos Modelos de Entidades

A figura 4.6 apresenta o digrama de classe dos modelos de entidades que são mapeados de acordo com as tabelas do banco de dados e são usados dentro da aplicação RESTful do servidor em conjunto com as classes de serviços.

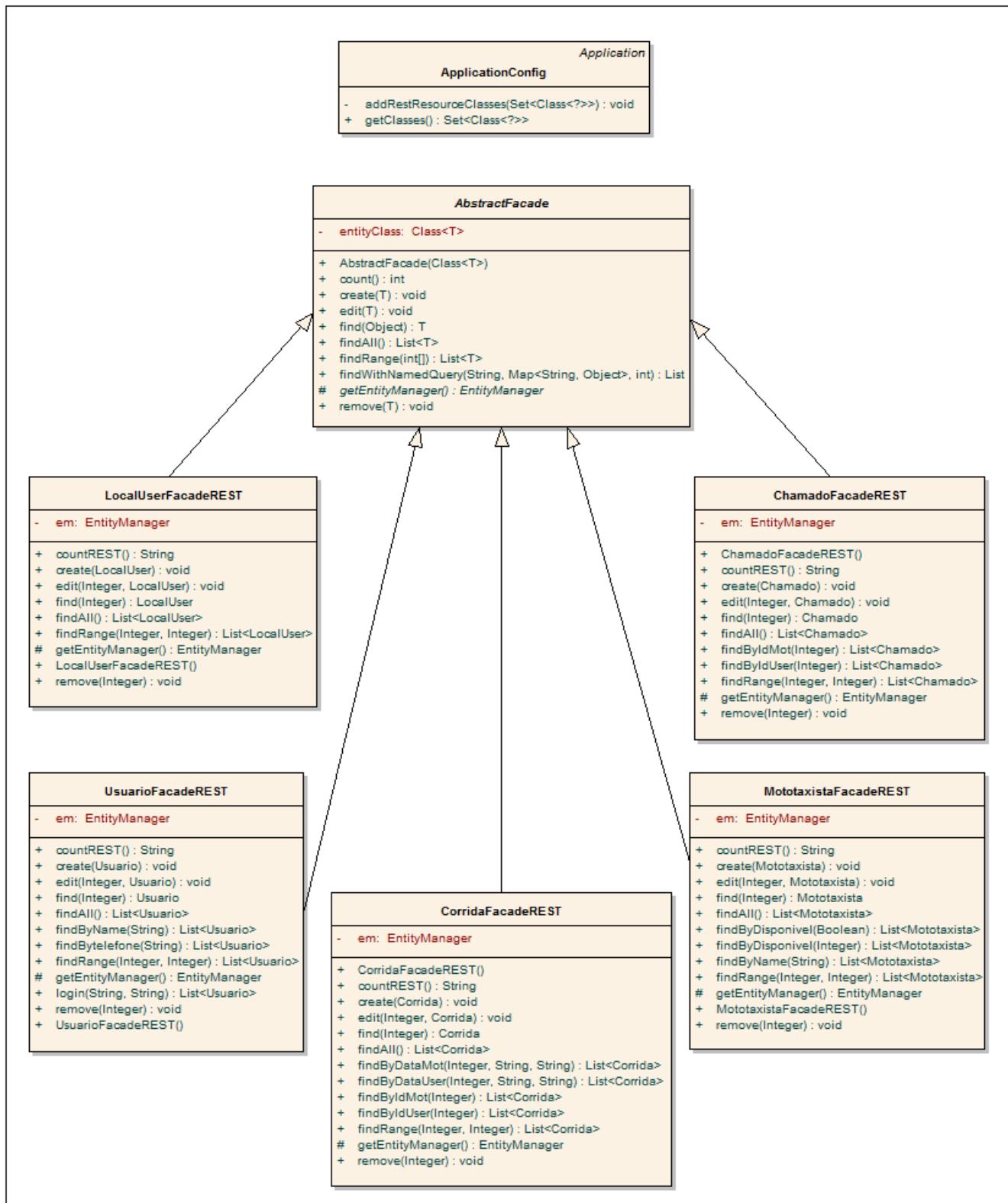


Figura 4.7: Diagrama Classe dos Serviços disponíveis

A figura 4.7 exibe o diagrama das classes localizadas no pacote de serviços. Estas são as responsáveis por usar os métodos HTTP do servidor de aplicações e prover a transição de informações entre o aplicativo móvel e o banco de dados do MySql. Todos os métodos apresentados nestas classes são acompanhados de alguma notação padrão do RESTful: GET,

POST, PUT ou DELETE. Por padrão estas classes contém os métodos para realização da persistência de informações no banco de dados sendo eles o *create(Objeto)*, *edit(Integer, Objeto)*, *find(Integer)*, *findAll()*, *remove(Integer)*, *countREST()*, *findRange(Integer, Integer)* que servem para criar, editar, buscar, buscar todos, remover, contagem total e buscar por intervalo de código, respectivamente, estes métodos são os que receber e retornam informações via HTTP.

As classes de serviços *UsuarioFacadeREST*, *MototaxistaFacadeREST*, *LocalUserFacadeREST*, *ChamadoFacadeREST* e *CorridaFacadeREST* estendem à classe *AbstractFacade* que contém os métodos padrões que usam entidades abstratas para persistir as informações dentro do banco. Na classe *AbstractFacade* existe o método *findWithNamedQuery(String, Map<String, Object>, int)* que usa o conceito de JPQL para realizar buscas de informações mais específicas da aplicação.

A classe *ApplicationConfig* é a responsável por mapear o caminho principal para acessar as classes de serviços e adicionar todas elas ao escopo da aplicação deixando-as disponíveis para fazer transações HTTP.

## 4.5 Construção

A construção da aplicação como um todo utilizou três softwares sendo eles o gerenciador de banco de dados Workbench e os IDE's NetBeans e Intel XDK.

### 4.5.1 Construção do Banco de Dados

A criação do banco de dados do MySQL deu-se através da geração automática do *schema* baseado no diagrama entidade relacionamento mostrado na figura 4.5. Encontrada nas opções do gerenciador de bancos Workbench, a opção *Forward Engineer* permite essa criação. A figura 4.8 mostra o acesso à opção que fica localizada na aba do editor gráfico de diagramas no menu *Database*.

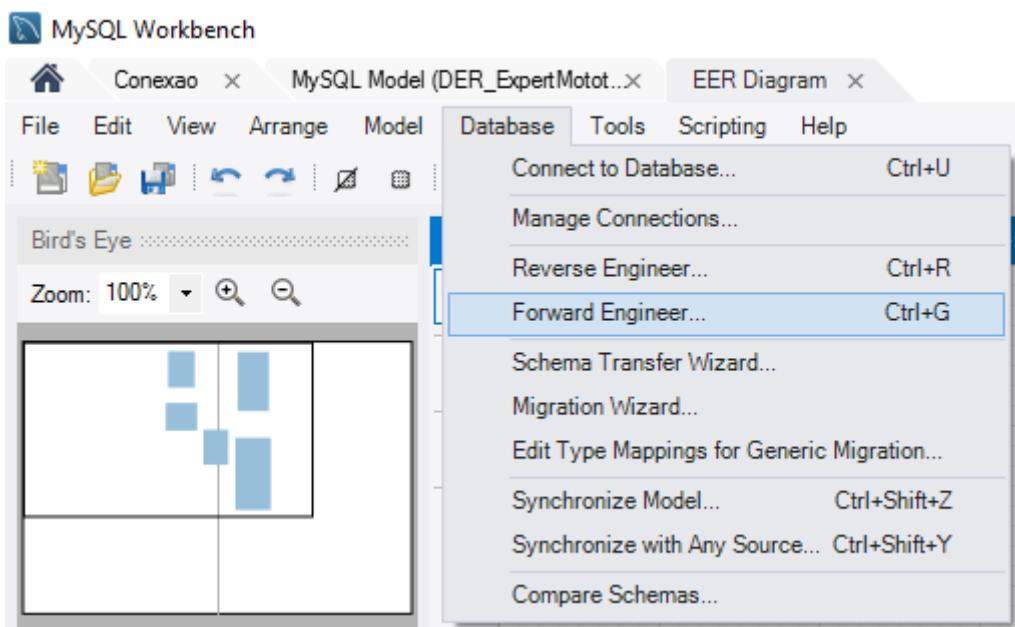


Figura 4.8: Geração automática do banco de dados