

**DOCUMENTO DE REQUISITOS**  
SISTEMA DE MOBILIDADE URBANA  
VERSÃO 0.7

# Sumário

1	MODELAGEM DO DOMÍNIO DO PROBLEMA.....	3
1.1	REFERÊNCIAS .....	3
2	REQUISITOS .....	4
2.1	REQUISITOS FUNCIONAIS .....	4
2.2	REQUISITOS NÃO FUNCIONAIS .....	5
2.3	REGRAS DE NEGÓCIOS.....	6
3	DIAGRAMAS DE ATIVIDADE UML.....	7
4	DIAGRAMAS DE CASO DE USO .....	15
4.1	Diagrama de caso de uso - Cadastro e Login .....	15
4.2	Diagrama de Caso de Uso – Funcionalidades Gerais .....	16
5	DIAGRAMA DE CLASSES DE DOMÍNIO .....	17
6	DIAGRAMA DE CLASSES DE PROJETO .....	18
6.1	DESCRIÇÃO DE MÉTODOS.....	18
6.2	REPRESENTAÇÃO GRÁFICA .....	19
7	DIAGRAMA DE OBJETOS .....	20

# 1 MODELAGEM DO DOMÍNIO DO PROBLEMA

## 1.1 REFERÊNCIAS

Waze, Google maps e 99.

## 1.2 LINGUAGEM

Esse documento será apresentado em linguagem coloquial sem gírias, dado que motoristas e usuários são stakeholders essenciais, e por questão de idade podem não ter o mesmo tipo de gírias.

## 1.3 DESCRIÇÃO DO SISTEMA

Esse sistema é um serviço que conecta passageiros que precisam fazer viagens de forma rápida, a motoristas cadastrados disponíveis nas proximidades. Por apresentar uma função semelhante à dos táxis.

Para utilizar esse sistema, pela primeira vez, é necessário fazer o download do aplicativo, disponível para Android e iOS, e fazer um cadastro, informando endereço de e-mail e número de telefone válidos. O novo usuário também terá que criar uma senha de acesso e aceitar os Termos e Condições e a Declaração de Privacidade. Após inserir o código de validação enviado via SMS, o passageiro precisa selecionar uma forma de pagamento (cartão de crédito ou débito, PIX ou dinheiro). Em seguida, a pessoa receberá um e-mail para confirmar o registro. Concluído o cadastro, já é possível solicitar um carro, informando o local onde gostaria de ser encontrado.

Motoristas próximos são informados e podem aceitar a solicitação. Quando a corrida é aceita, o app informa quanto tempo o veículo irá demorar para chegar. O serviço também indica nome do motorista, marca, modelo e placa do carro, como uma forma de tornar a operação mais segura. A tarifa é calculada automaticamente e cobrada ao final da corrida através do método de pagamento escolhido (além das preferências cadastradas, também é possível escolher a opção de pagamento por dinheiro). Concluída a corrida, o passageiro poderá avaliar o motorista e vice-versa, como uma forma de tornar a rede cada vez mais confiável para ambos.

## 2 REQUISITOS

### 2.1 REQUISITOS FUNCIONAIS

**[RF1]** O sistema deve iniciar com uma tela de cadastro/login para que o usuário crie ou entre em sua conta. Esse requisito funcional deve estar de acordo com a RN1.

**[RF2]** O sistema permitirá o usuário Incluir/alterar/excluir informações pessoais. Esse requisito funcional deve estar de acordo com a RN2.

**[RF3]** O sistema deve conter um mapa para que o usuário possa mandar sua localização para o motorista, para que ele possa chegar ao local. Esse requisito funcional deve estar de acordo com a RN3.

**[RF4]** O sistema permitirá que o usuário poderá escolher o motorista, na qual lhe informará marca, modelo e placa do carro, que o mesmo poderá analisar o perfil do motorista.

**[RF5]** O sistema irá permitir ao motorista que ele possa fazer relatórios sobre algo da corrida, relatando se teve problemas ou não.

**[RF6]** O sistema irá automaticamente perguntar ao usuário logo depois que ele solicitar a corrida qual será a forma de pagamento, sendo elas PIX/crédito/débito/dinheiro. Esse requisito funcional deve estar de acordo com a RN4.

**[RF7]** O sistema permitirá ao usuário que ele possa avaliar em estrelas de 1 a 5 o motorista logo depois da corrida. Esse requisito funcional deve estar de acordo com a RN5.

**[RF8]** O sistema permitirá ao motorista que ele possa avaliar o usuário de estrelas de 1 a 5 logo depois da corrida. Esse requisito funcional deve estar de acordo com a RN5.

## 2.2 REQUISITOS NÃO FUNCIONAIS

**[RNF1]** O sistema deve ser responsivo. (Requisito de facilidade de uso)

**[RNF2]** O sistema deve rodar nos sistemas operacionais android e ios. (Requisito de portabilidade)

**[RNF3]** O sistema deve se manter online 24/7. (Requisito de confiabilidade)

**[RNF4]** O sistema deve assegurar o controle de acesso adequado para prevenir acessos não autorizados. (Requisito de segurança)

**[RNF5]** O sistema não utilizará informações privadas sem o consentimento do usuário. (Requisito de privacidade)

**[RNF6]** Quando instalado, o sistema não deve ocupar mais do que 500MB de armazenamento interno. (Requisito de espaço)

**[RNF7]** O sistema não apresentará aos usuários quaisquer dados de cunho privativo. (Requisito ético)

**[RNF8]** O sistema deverá ser desenvolvido na linguagem Java. (Requisito de implementação)

**[RNF9]** O documento de requisitos deve seguir as normas do padrão ABNT. (Requisitos de padrões)

**[RNF10]** O sistema deverá se comunicar com o SQL Server. (Requisito de interoperabilidade)

**[RNF11]** O sistema deve garantir um tempo de resposta médio inferior a 5 segundos para todas as transações e interações. (Requisito de eficiência)

**[RNF12]** Um relatório de acompanhamento do desenvolvimento do projeto deverá ser fornecido toda segunda-feira. (Requisito de entrega)

**[RNF13]** O sistema deve manter registros precisos e completos de todas as transações e interações realizadas na plataforma. (Requisito legal)

## 2.3 REGRAS DE NEGÓCIOS

**[RN1]** Para o cadastro, é necessário nome completo, cpf, senha e e-mail usuário.

**[RN2]** Para Incluir/alterar/excluir, o usuário poderá fazer isso com sua senha e seu e-mail.

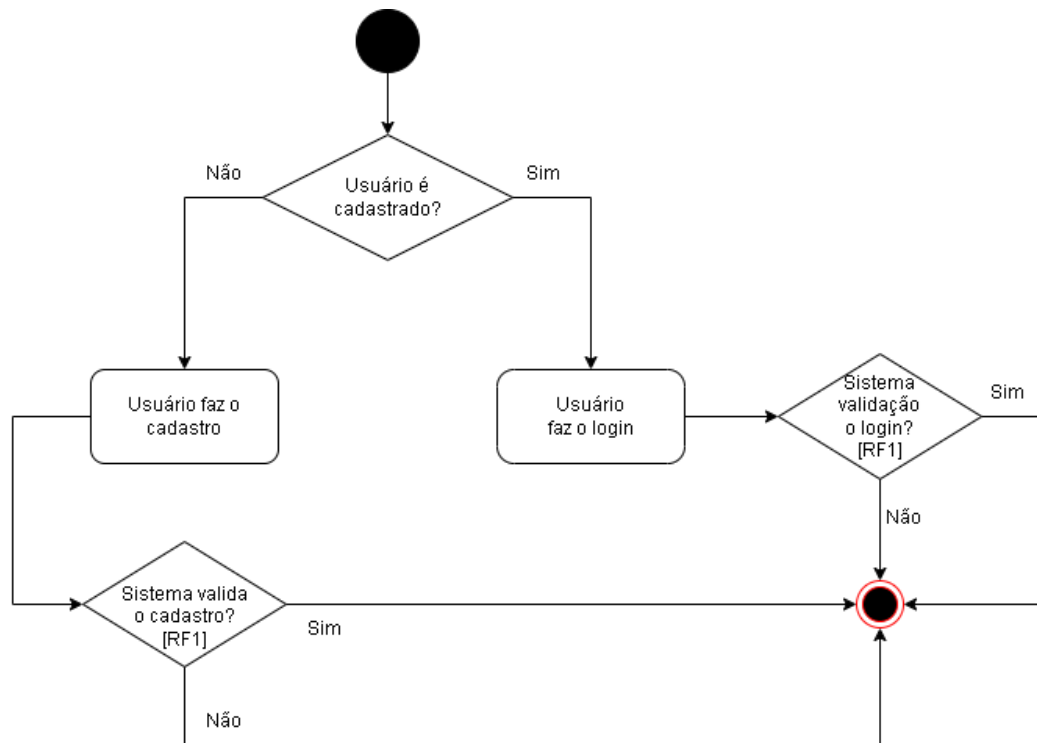
**[RN3]** O mapa do sistema mostra certas regiões da cidade nas cores amarelo, laranja e vermelho. Essas cores indicam regiões com procura mais alta por viagens no momento.

**[RN4]** O usuário define o pagamento antes de pedir o carro. Só é permitido usar dinheiro, cartão de crédito ou débito ou PIX. De acordo com o sistema, antes de o motorista aceitar a corrida de um usuário, não há indicação de que o valor da viagem será pago em dinheiro de papel.

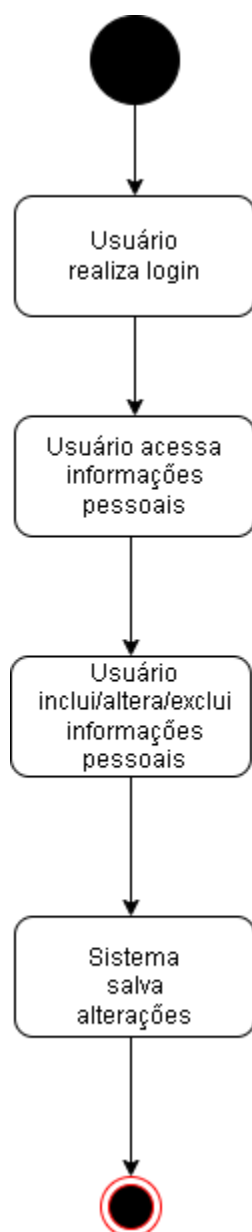
**[RN5]** Após cada viagem, os usuários e motoristas parceiros têm a oportunidade de avaliar um ao outro, com 1 a 5 estrelas, com base na experiência da viagem. As avaliações são anônimas. Você não verá as avaliações individuais vinculadas a uma viagem ou pessoa específica.

### 3 DIAGRAMAS DE ATIVIDADE UML

**[RF1]** O sistema deve iniciar com uma tela de cadastro/login para que o usuário crie ou entre em sua conta. Esse requisito funcional deve estar de acordo com a RN1.

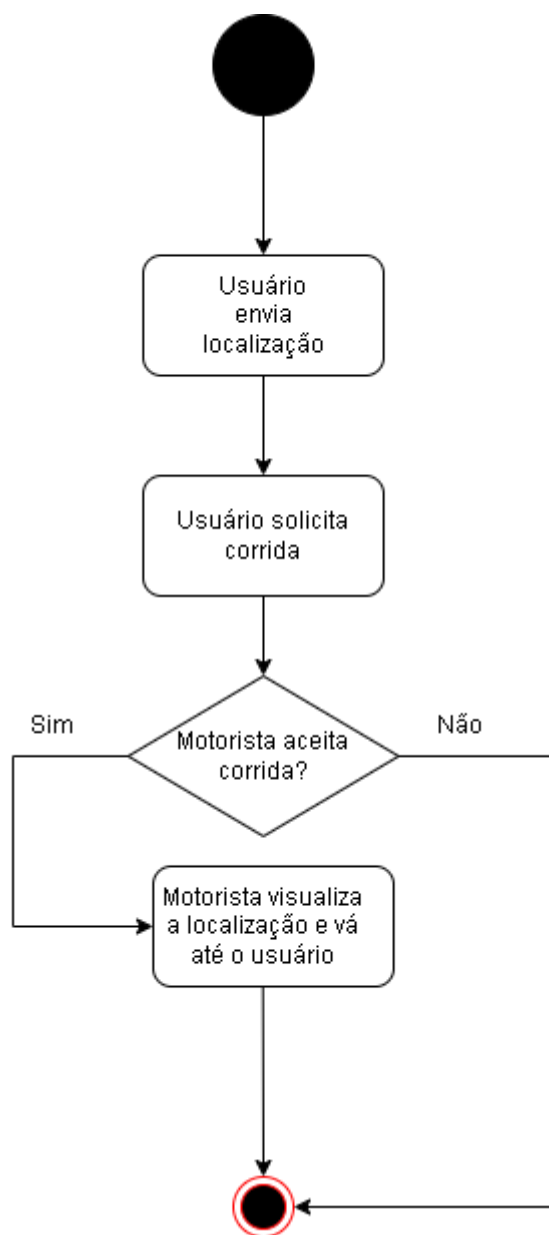


**[RF2]** O sistema permitirá o usuário Incluir/alterar/excluir informações pessoais. Esse requisito funcional deve estar de acordo com a RN2.

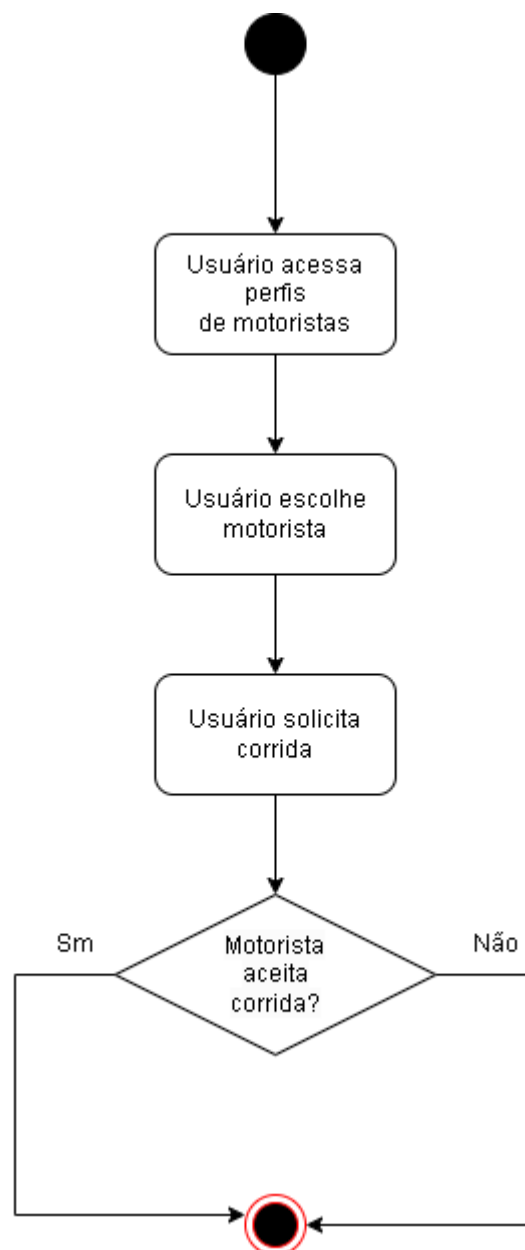




**[RF3]** O sistema deve conter um mapa para que o usuário possa mandar sua localização para o motorista, para que ele possa chegar ao local. Esse requisito funcional deve estar de acordo com a RN3.



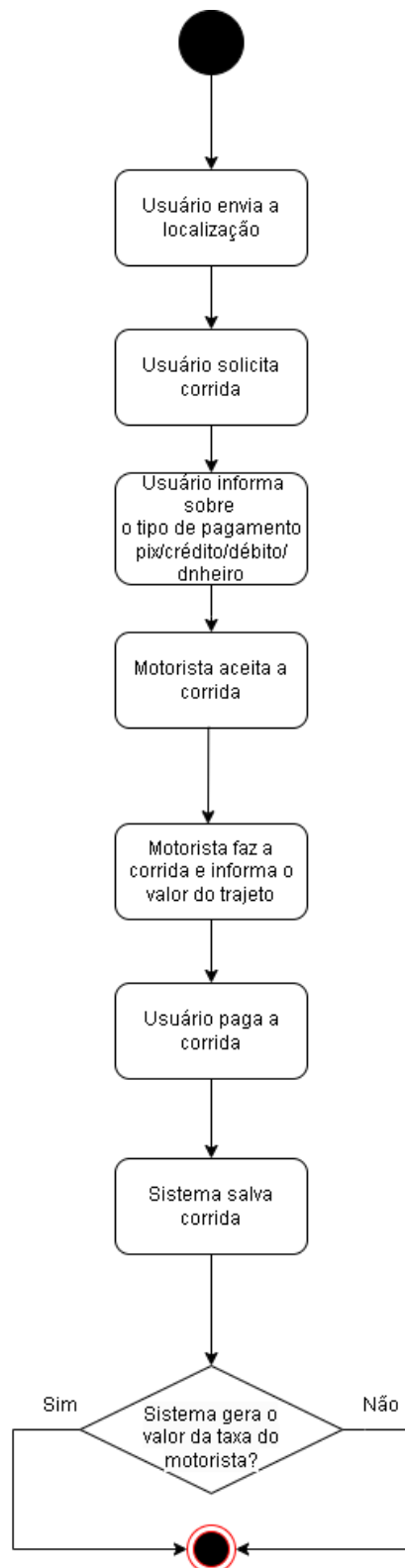
**[RF4]** O sistema permitirá que o usuário poderá escolher o motorista, na qual lhe informará marca, modelo e placa do carro, que o mesmo poderá analisar o perfil do motorista.



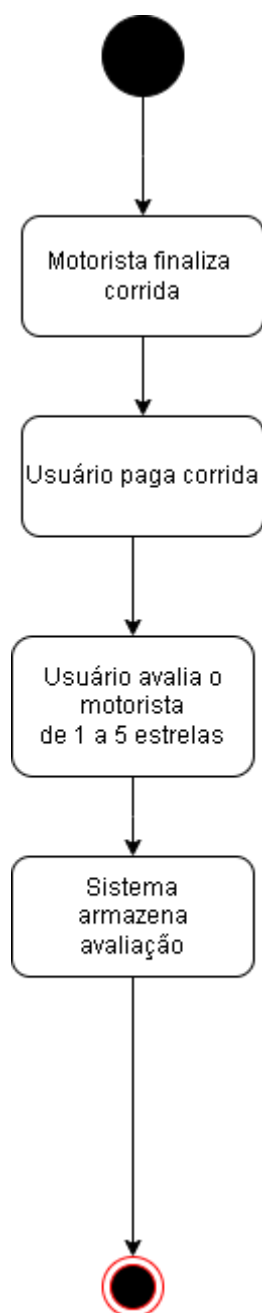
**[RF5]** O sistema irá permitir ao motorista que ele possa fazer relatórios sobre algo da corrida, relatando se teve problemas ou não.



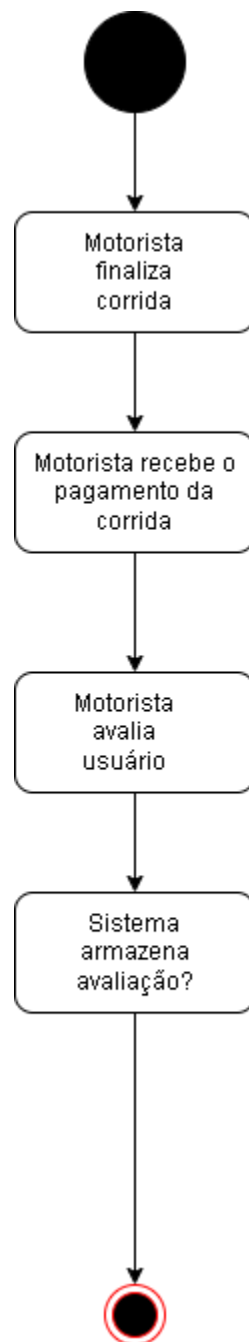
**[RF6]** O sistema irá automaticamente perguntar ao usuário logo depois que ele solicitar a corrida qual será a forma de pagamento, sendo elas PIX/crédito/débito/dinheiro. Esse requisito funcional deve estar de acordo com a RN4.



**[RF7]** O sistema permitirá ao usuário que ele possa avaliar em estrelas de 1 a 5 o motorista logo depois da corrida. Esse requisito funcional deve estar de acordo com a RN5.



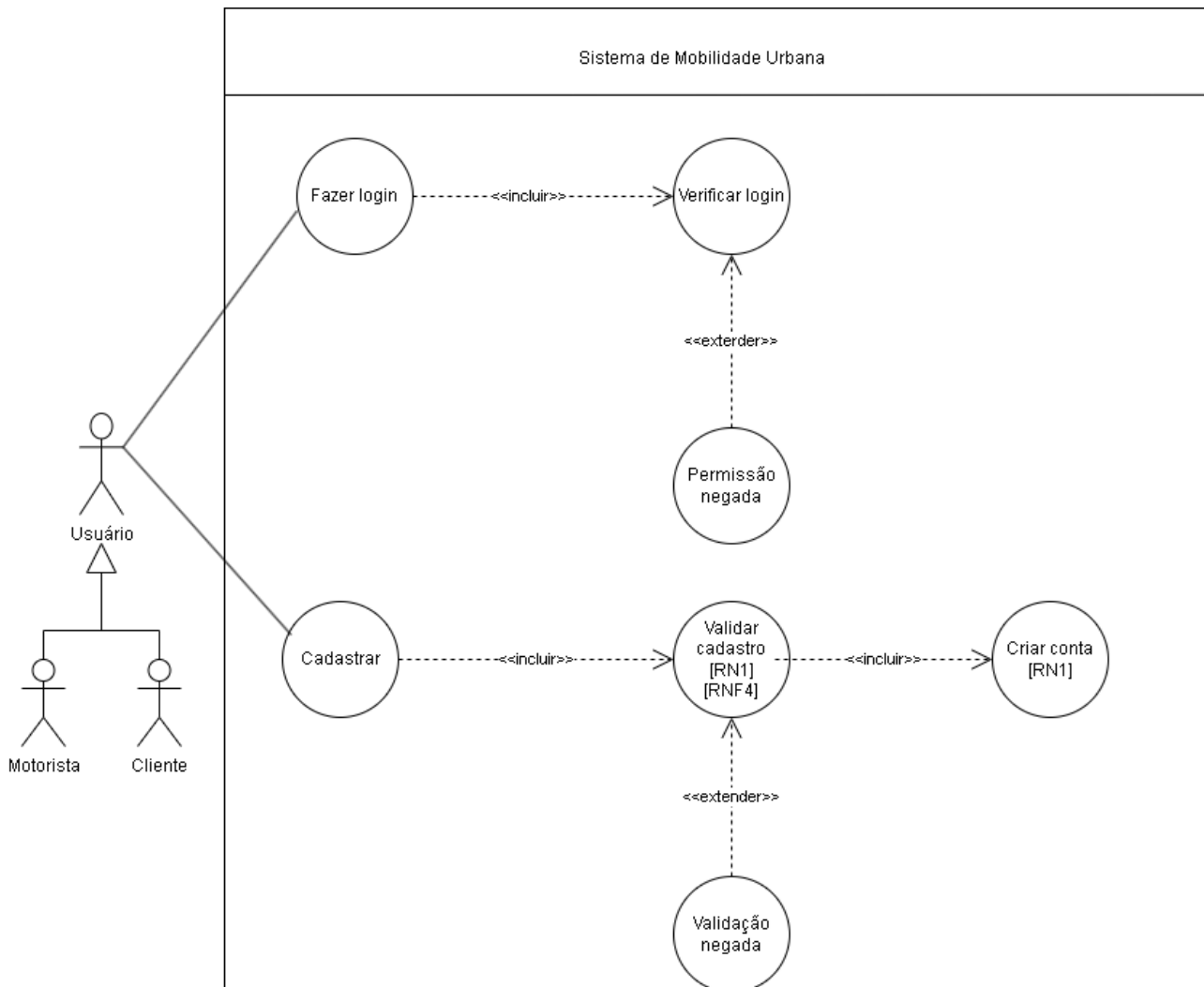
**[RF8]** O sistema permitirá ao motorista que ele possa avaliar o usuário de estrelas de 1 a 5 logo depois da corrida. Esse requisito funcional deve estar de acordo com a RN5.



## 4 DIAGRAMAS DE CASO DE USO

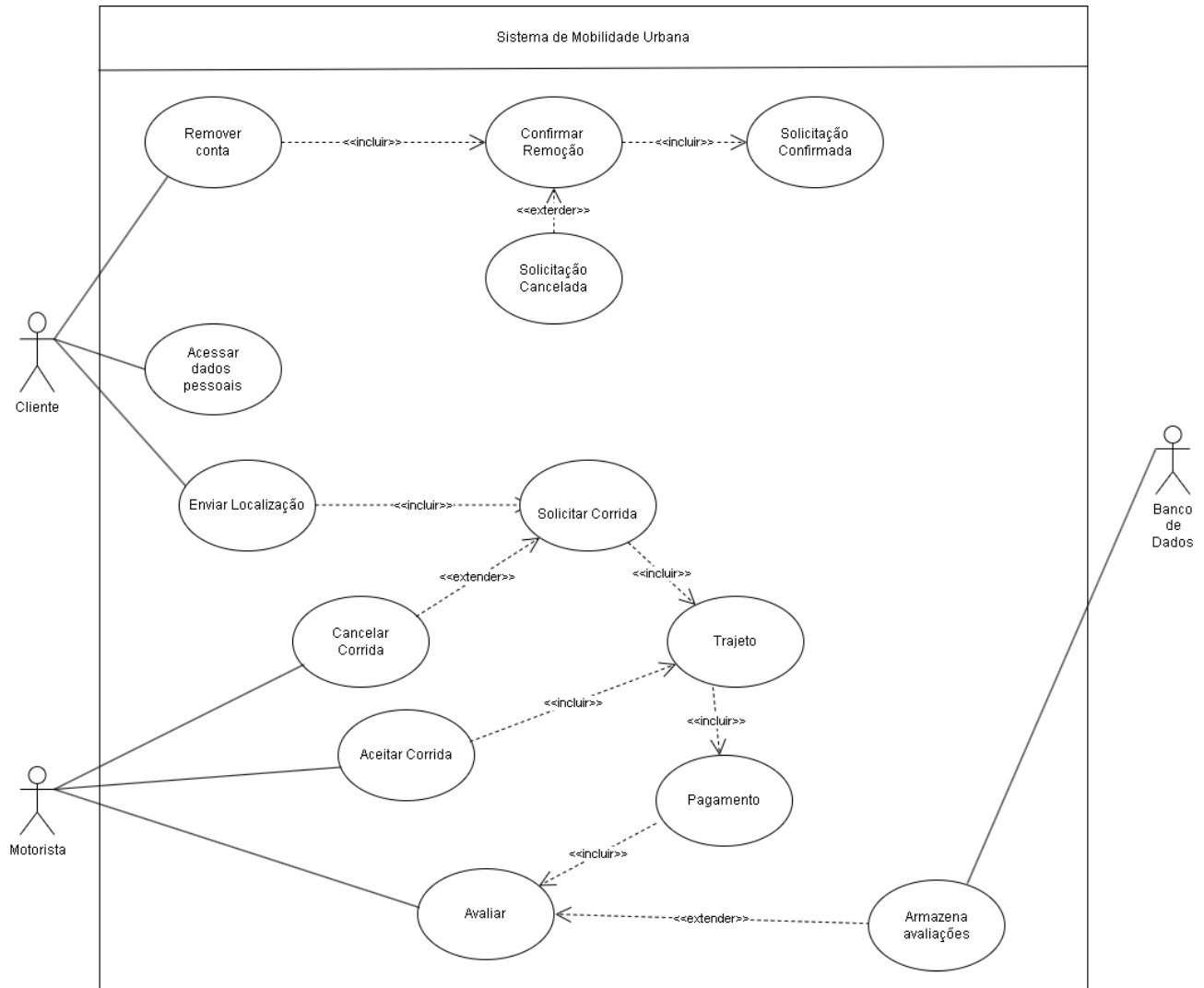
### 4.1 Diagrama de caso de uso - Cadastro e Login

O seguinte diagrama contempla os casos de uso relacionados a cadastro e login.



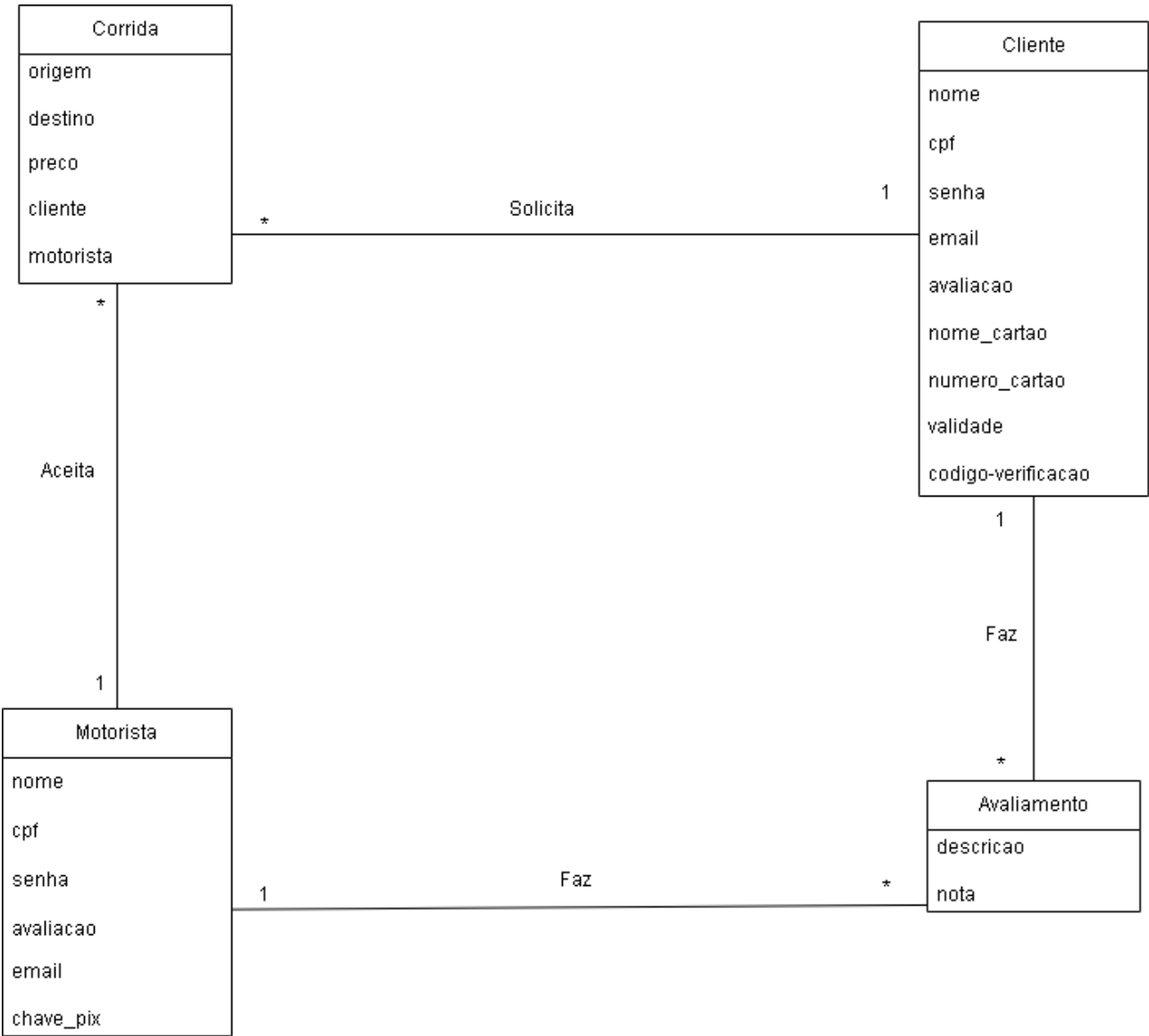
## 4.2 Diagrama de Caso de Uso – Funcionalidades Gerais

Já considerado que os usuários, tanto passageiros como motoristas, já tenham se autenticado corretamente no sistema, esse diagrama de caso de uso demonstra a funcionalidade geral do sistema de mobilidade urbana.





# 5 DIAGRAMA DE CLASSES DE DOMÍNIO



## 6 DIAGRAMA DE CLASSES DE PROJETO

### 6.1 DESCRIÇÃO DE MÉTODOS

Métodos da classe Corrida

**emitirPreco()** : Permite emitir o preço do trajeto da corrida.

Métodos da classe Usuário

**gerarSenha()** : Gera uma senha para o usuário.

**alterarSenha()** : Alterar a senha de acesso para o usuário.

**avaliar()** : Permite ao usuário poder avaliar.

Métodos da classe Motorista

**acessarLocalização()** : Permite ao motorista acessar a atual localização do cliente.

**aceitarCorrida()** : Permite ao motorista aceitar a corrida que foi solicitada.

**cancelarCorrida()** : Permite ao motorista cancelar a corrida solicitada.

Métodos da classe Cliente

**solicitarCorrida()** : Permite ao cliente solicitar uma corrida.

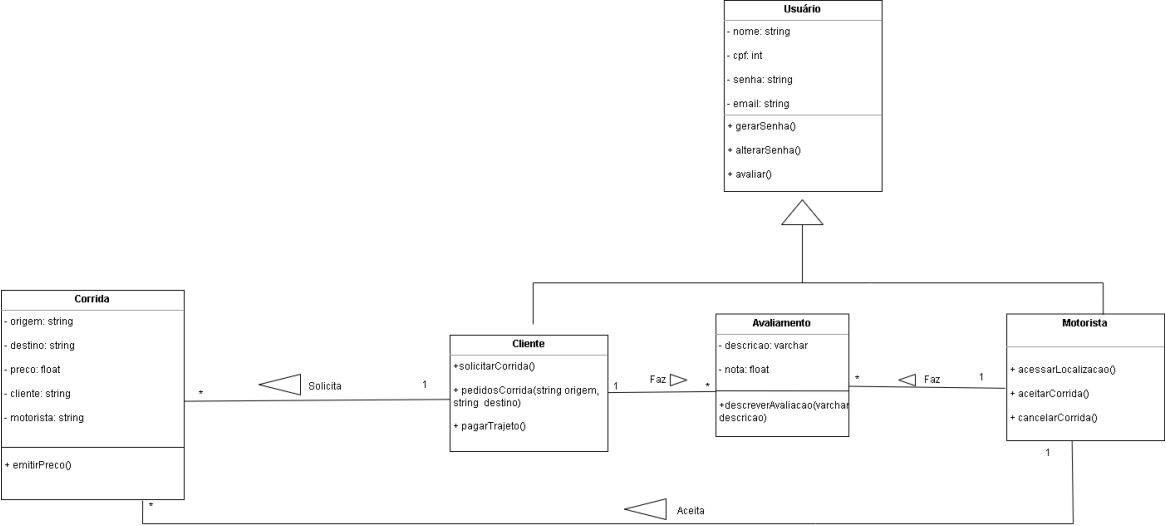
**pagarTrajeto()** : Permite ao cliente pagar o valor da corrida.

**pedidosCorrida(string origem, string destino)** : Processa os pedidos de corridas a partir de uma origem e destino específicos.

Métodos da classe Avaliação

**descreverAvaliação(varchar descrição)** : Permite ao cliente fazer uma descrição de sua avaliação sobre o motorista e vice e versa.

# 6.2 REPRESENTAÇÃO GRÁFICA



# 7 DIAGRAMA DE OBJETOS

