

Versão 2.0

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 02/04/2015 | 1.0 | Tópico 1: Criação inicial do documento introdução, Finalidade, Escopo, Referências e Visão Geral itens criado e revisados (1, 1.1, 1.2, 1.3, 1.4) | Jefferson Oliveira |
| 06/04/2015 | 1.1 | Tópico 2: Continuação do documento Representação Arquitetural, itens criado e revisados ( 2 ) | Jefferson Oliveira |
| 08/042015 | 1.2 | Tópico 3: Metas e Restrições da Arquitetura itens criado e revisados ( 3 e 3.1) | Jefferson Oliveira |
| 09/04/2015 | 1.3 | Tópico 4: Requisitos funcionais, Requisitos não funcionais, Diagrama de caso de uso itens criado e revisados (4, 4,1) | Jefferson Oliveira |
| 13/04/2015 | 1.4 | Tópico 5: Visão lógica, Visão Geral, Pacotes de Design Significativos do Ponto de Vista da Arquitetur itens criado e revisados (5, 5.1 e 5.2) | Jefferson Oliveira |
| 14/04/2015 | 1.5 | Tópico 6: Visão de Processos iten criado e revisado ( 6 ) | Jefferson Oliveira |
| 15/04/20115 | 1.6 | Tópico 7: Visão de Implantação iten criado e revisado ( 7 ) | Jefferson Oliveira |
| 16/04/2015 | 1.7 | Tópico 8: Visão da Implementação iten criado e revisado ( 8 ) | Jefferson Oliveira |
| 17/04/2015 | 1.8 | Tópico 9: Visão de Dados, Modelo lógico da Base de Dados, Criação Física do Modelo de Dados, Dicionário de Dados iten criado e revisado (9, 9.1, 9.2 e 9.3) | Jefferson Oliveira |
| 18/042015 | 1.9 | Tópico 10: Tamanho e Desempenho, iten criado e revisado (10) | Jefferson Oliveira |
| 19/04/2015 | 2.0 | Tópico 11: Qualidade, Usabilidade, Condiabilidade e Desempenho itens criado e revisado. (11, 11.1, 11.2, 11.3 | Jefferson Oliveira |

Índice Analítico

1. Introdução 5

1.1 Finalidade 5

1.2 Escopo 5

1.3 Referências 5

1.4 Visão Geral 5

2. Representação Arquitetural 5

3. Metas e Restrições da Arquitetura 5

7.1 Restrições 5

4. Visão de Casos de Uso 5

4.1 Realizações de Casos de Uso 5

5. Visão Lógica 5

5.1 Visão Geral 5

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 5

6. Visão de Processos 5

7. Visão de Implantação 5

8. Visão da Implementação 5

8.1 Camadas 5

9. Visão de Dados 5

### 9.1 Modelo lógico da Base de Dados 5

9.2 Criação Física do Modelo de Dados 5

### 9.3 Dicionário de Dados 5

10. Tamanho e Desempenho 5

11. Qualidade 5

11.1 Usabilidade 5

11.2 Confiabilidade 5

11.3 Desempenho 5

# Introdução

O “Documento de Arquitetura de Software - TransporLine” é definir um modelo arquitetural para ser aplicado ao desenvolvimento do Sistema de Gerenciamento de Moto-taxistas, bem como reunir todas as informações necessárias ao controle das atividades de Arquitetura, oferecendo uma visão macro dos requisitos arquiteturais e não funcionais para suportar o desenvolvimento do mesmo.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

## Escopo

O projeto visa desenvolver um sistema Web para proporcionar aos usuários, acesso aos moto-taxistas, além de facilitar buscas relacionadas a localização. Também manter organizados todas as corridas .Notando-se a necessidade de uma organização mais rápida e precisa do Moto-táxi dentro do sistema nas cooperativas, este trabalho tem como sua principal função o desenvolvimento de um sistema que disponibilize uma base de dados online. O sistema está habilitado para ser utilizado nos principais navegadores web e também em dispositivos móveis.

## Referências

MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 4. ed. Rio de Janeiro: LTC, 2012.

SOMMERVILLE, Ian. **Engenharia de Software. Tradução**: Ivan Bosnic e Kalinka G. de O. Gonçalves. 9. ed. São Paulo: Pearson Pratice Hall, 2009.

ABREU, Maurício Pereira de; MACHADO, Felipe Nery Rodrigues. **Projeto de Banco de Dados: Uma Visão Prátrica**. 16. Ed. rev. E atual. São Paulo: Érica, 2009.

GAMMA, Erich et al. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objeto**. Tradução: Luiz A. Meirelles Salgado. Porto Alegre: Bookman, 2000.

GUEDES, Gilleanes T.A. **UML 2: Uma Abordagem Prática**. São Paulo: Novatec Editora, 2009.

ARTIGO. ***Requisitos não funcionais.*** *Disponível em:* <http://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>

BANCO DE DADOS. ***Tecnologia de projeto.****Disponível em:* <http://www.tecnologiadeprojetos.com.br/banco_objetos/%7B228E90E8-4727-4F53-9ABD-601EE7C5AE50%7D_modelo%20de%20planejamento.pdf>

FIGUEIREDO.  ***Requisitos funcionais.****Disponível em:* <http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf>

LINHA DE CODIGO. ***Plano de testes mestre.*** *Disponível em:* <http://www.linhadecodigo.com.br/artigo/3333/tmap-nexttest-management-approach-processo-plano-de-testes-mestremtp-planejamento-e-controle-parte-8-2.aspx>

METODOLOGIA.  ***Arquitetura de informação.****Disponível em:* <http://arquiteturadeinformacao.com/recursos/metodologia/dicas-para-um-prototipo-eficaz/>

PDF. ***UML Diagrama caso de uso.*** *Disponível em:* <http://www.les.inf.puc-rio.br/wiki/images/6/61/Aula01-diagrama_casos_uso.pdf>

UFCG. ***Diagrama de interação e seqüência.*** *Disponível em:* <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/interacao/sequencia.htm>

## Visão Geral

Este documento está organizado em seções e subseções, onde descrevem as características do sistema, e estar organizado da seguinte forma:

Na seção 1, a Introdução, onde fala informações necessárias ao controle das atividades de Arquitetura.

Na seção 2, a Representação da Arquitetura descreve qual é a arquitetura de software do sistema atual e como ela é representada.

Na seção 3, a Metas e Restrições de Arquitetura descreve os requisitos de software e os objetivos que têm um impacto significativo na arquitetura.

Na seção 4, a Visão de Casos de Uso apresenta uma lista dos casos de uso ou cenários do modelo de casos de uso.

Na seção 5, a Visão Lógica descreve as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes.

Na seção 6, a Visão de Processos descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves).

Na seção 7, a Visão de Implantação descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado.

Na seção 8, a Visão de Implementação descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.

Na seção 9, a Visão de Dados descrição da perspectiva de armazenamento de dados persistentes do sistema.

Na seção 10, o Tamanho e Desempenho apresentam uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.

Na seção 11, a Qualidade apresenta uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema:, confiabilidade, Usabilidade, Desempenho e assim por diante.

# Representação Arquitetural

O sistema será baseado na arquitetura cliente-servidor, no qual por ser web poderá se conectar via qualquer plataforma usando um browser, se conectando a internet em um servidor remoto, atualizando suas informações semanalmente (ou sempre que for requisitado).

As informações serão inseridas no servidor pelos administradores pelo site na web que acessará diretamente o SGBD (Sistema de Gerenciamento de Banco de Dados) no servidor.

Utilizando um servidor com configurações moderadas, ele atenderá todas as requisições com fluidez, com até 500 pesquisas simultâneas e colocando em filas as que surgirem no mesmo instante, utilizando também de recursos de gerenciamento mais novos e atualizados para não depender somente do hardware, como o sistema de cache guardando informações também no dispositivo móvel.

O servidor funcionará em um Sistema operacional linux e um servidor apache, com configurações de firewall IPTables e banco de dados MySQL.

O Servidor contara com 1 Mbps(Mega bites por segundo) dedicado no inicio tendo possíveis alterações de acordo com a demanda, e na parte do usuário uma rede 3G de 128Kbps em pleno funcionamento suprirá as necessidades.

# Metas e Restrições da Arquitetura

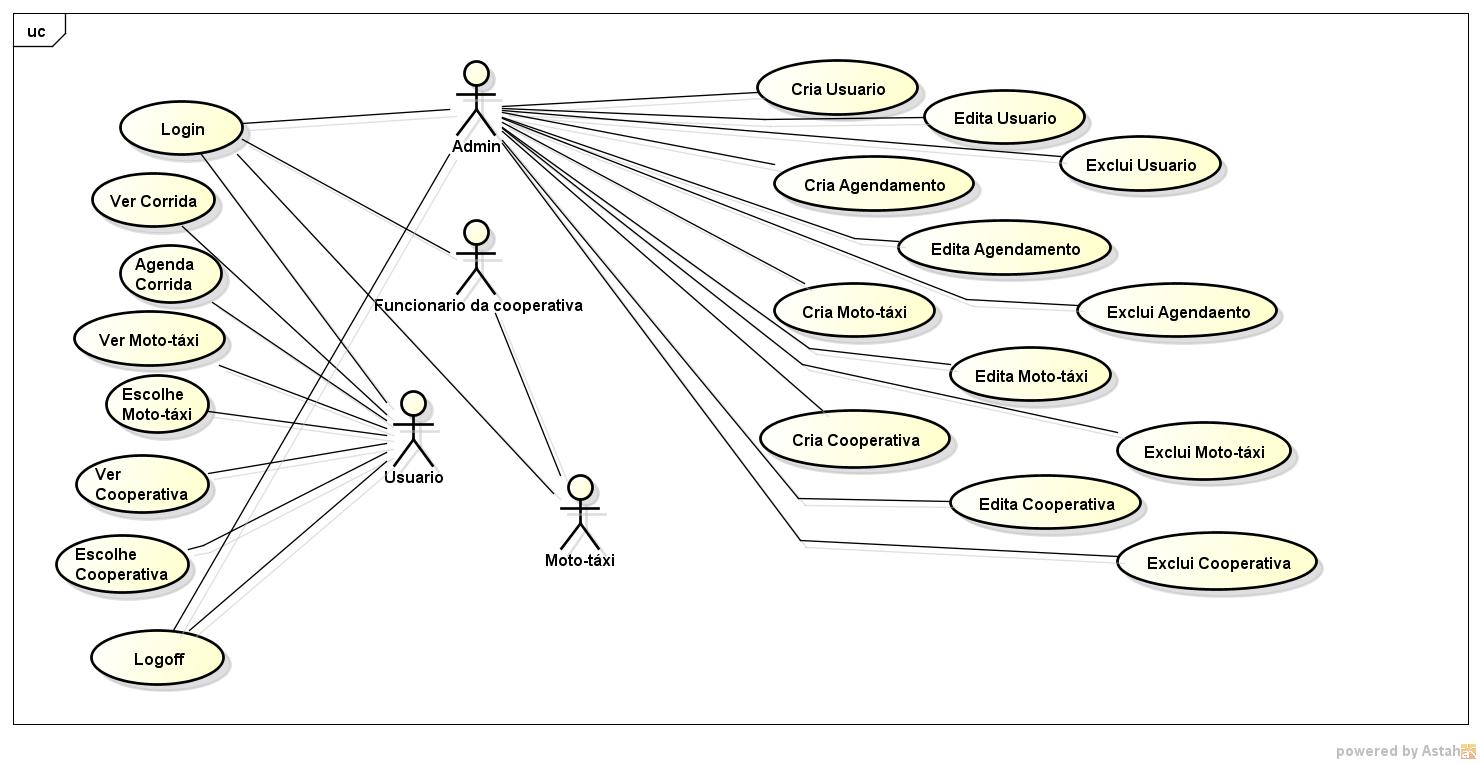
* 1. **Restrições**

Para que o sistema funcione de forma correta há a necessidade de mantê-lo hospedado em um servidor web. Também o registro de um nome de domínio. Para poder obter acesso ao sistema deverá ser realizado via browser a partir de um dispositivo conectado à internet.

# Visão de Casos de Uso

## Requisitos Funcionais

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Editar usuários | O administrador edita os usuários cadastrados. |
| Criar usuários | O administrador cria usuários para o sistema. |
| Excluir usuários | O administrador exclui usuários do sistema. |
| Editar moto-táxi | O administrador edita moto-táxi cadastrados. |
| Criar moto-táxi | O administrador cadastra moto-táxi no sistema. |
| Excluir moto-táxi | O administrador exclui moto-táxi do sistema. |
| Criar corrida | O administrador cadastra agendamento. |
| Editar corrida | O administrador edita agendamento. |
| Excluir corrida | O administrador exclui agendamento. |
| Excluir cooperativa | O administrador exclui cooperativa. |
| Criar cooperativa | O administrador cria cooperativa. |
| Efetuar login | O administrador efetua o login no servidor para ter acesso aos dados do sistema |
| Efetuar logoff | O administrador efetua o logoff para sair do sistema. |



## Requisitos Não Funcionais

|  |  |
| --- | --- |
| RNF DE PRODUTO | |
| USABILIDADE | Terá uma interface limpa, amigável e intuitiva projetada para usuários de todos os níveis intelectuais com ícones representando suas funções. |
| EFICIÊNCIA | O aplicativo deverá ocupar no máximo 30MB de espaço de armazenamento e fornecerá acesso ao banco de dados com no máximo 3 segundos de espera. |
| DISPONIBILIDADE | O sistema TransporLine, deverá esta disponível 24 horas por dias, 7 dias por semana.  O banco de dados deve ser atualizado em tempo real.  O hardware do sistema deve ser assegurado por um gerador. |
| CONFIABILIDADE | A área administrativa do sistema deverá ser restrito, apenas o administrador terá acesso através de login e senha.  Não deve revelar aos operadores todos os dados pessoais do cliente ou administrador.  O sistema será seguro. |

## Realizações de Casos de Uso

|  |  |
| --- | --- |
| **CU-01** | Visualizar Corridas/Rota |
| **CU-02** | Visualizar Cooperativas e Moto-Táxi |
| **CU-03** | Escolher Cooperativas e Moto-Táxi |
| **CU-04** | Criar, Editar e Excluir Usuários do Sistema |
| **CU-05** | Criar, Editar e Excluir Moto-Táxi e Cooperativa do Sistema |
| **CU-06** | Criar, Editar e Excluir Agendamento |
| **CU-07** | Efetuar Login para Acesso ao Sistema |
| **CU-08** | Efetuar Logoff para Sair do Sistema |

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-01** Visualizar Corridas/Rota |
| **Breve Descrição** | O usuário estará habilitado para ver as corridas |
| **Atores Envolvidos** | Usuário |
| **Pré-Condições** | Usuário deve estar cadastrado no sistema. |
| **Fluxo Principal** | P1 - O caso de uso é iniciado quando o usuário acessa o sistema.  P2 - O sistema apresenta interface de consulta de corridas.  P3 - O usuário seleciona a corrida.  P4 - O usuário escolhe a corrida.  P5 - O usuário clica em sair. |
| **Pós-Condições** | O usuário é mantido no sistema. |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1. No P1 caso o usuário não esteja cadastrado.  F2. O sistema volta a tela inicial. |

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-02** Ver Cooperativa e Moto-Táxi |
| **Breve Descrição** | O usuário poderá ver o Cooperativa escolhida para consultar suas corridas. |
| **Atores Envolvidos** | Usuário**.** |
| **Pré-Condições** | Usuário deve estar cadastrado no sistema. |
| **Fluxo Principal** | P1 - O caso de uso é iniciado quando o usuário acessa o sistema e seleciona a opção “Cooperativa” no menu principal.  P2 - O sistema apresenta interface de consulta de Cooperativa.  P3 - O usuário seleciona o Cooperativa.  P4 - O usuário clica em sair. |
| **Pós-Condições** | Cadastro do usuário é mantido no sistema. |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1. No P1 caso o usuário não esteja cadastrado.  F2. O sistema volta à tela inicial.  Fluxo Alternativo B:  F1. No P3 o usuário poderá selecionar outro Cooperativa. |
|  | |
| **Nome do Caso de Uso** | **CU-03** Escolher Cooperativas e Moto-Táxi |
| **Breve Descrição** | O usuário poderá Classificar as suas principais corridas para poder fazer um balanço total do preço.  E também classificar as cooperativas de acordo com sua escolha. |
| **Atores Envolvidos** | Usuário. |
| **Pré-Condições** | Usuário deve estar cadastrado no sistema |
| **Fluxo Principal** | P1- O caso de uso é iniciado quando o usuário acessa o sistema e seleciona a opção “Classificar corridas” ou “Classificar cooperativas” no menu principal.  P2 - O sistema apresenta interface para classificar a corridas e cooperativas.  P3 - O usuário escolhe o que deseja classificar.  P4 - O usuário seleciona as corridas.  P5 - O usuário seleciona as cooperativas.  P6 - O usuário clica em sair |
| **Pós-Condições** | As corridas estarão classificadas |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1. No P1 caso o usuário não esteja cadastrado.  F2. O sistema volta a tela inicial.  Fluxo Alternativo B:  F1. No P4 o usuário poderá desmarcar e selecionar outras corridas e cooperativas. |

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-04**  Criar,Editar e Excluir Usuários do Sistema |
| **Breve Descrição** | O administrador tem permissão para criar, editar e excluir os usuários do sistema quando necessário. |
| **Atores Envolvidos** | Administrador; Usuário. |
| **Pré-Condições** | O usuário deverá estar devidamente cadastrado no sistema com perfil de Administrador. |
| **Fluxo Principal** | P1 - O caso de uso é iniciado quando o usuário acessa o sistema e seleciona a opção “usuários” no menu principal.  P2 - Em seguida seleciona qual tarefa deseja Executar. Criar, Editar e Excluir.  P3 - Após a seleção da atividade o sistema apresenta a interface para realizar a tarefa de sua escolha.  P4 - O administrador recebe os dados e modifica os dados.  P5 - O administrador ver os dados novos e confirma para salvar.  P6 - O usuário clica em sair |
| **Pós-Condições** | A execução do administrador é feita com sucesso |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1. No P1 caso o usuário não esteja cadastrado.  F2. O sistema volta a tela inicial.  Fluxo Alternativo B:  F1. Caso escolha criar usuário. Em seguida preenche um formulário com seus dados.  F2. Os dados do usuário podem está errados no sistema.  F3. Se a escolha for excluir usuário será apresentada a confirmação de exclusão.  F4. O usuário faz um novo cadastro.  Fluxo Alternativo C:  F1. P5 a conexão da internet pode cair, assim terá que enviar os dados novamente.  Fluxo Alternativo D:  F1. P6 os dados novos do usuário podem ser feito da maneira errada.  F2. Assim tendo que o administrador informar novamente os dados. |

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-05** Criar, Editar e Excluir Moto-Táxi e Cooperativa do Sistema |
| **Breve Descrição** | O administrador estar habilitado a criar, editar e excluir os moto-táxi e cooperativa do sistema. |
| **Atores Envolvidos** | Administrador. |
| **Pré-Condições** | O ator deverá estar devidamente cadastrado no sistema com perfil de Administrador e o moto-táxi e a cooperativa já ter sido criado. |
| **Fluxo Principal** | P1. O administrador efetuar o login do sistema.  P2. O caso de uso é iniciado quando o administrador acessa o sistema e seleciona a opção “moto-táxi ou cooperativa” no menu.  P3. Em seguida seleciona qual tarefa deseja Executar. Criar, Editar e Excluir moto-táxi ou cooperativa.  P4. Após a seleção da atividade o sistema apresenta a interface para realizar a tarefa de sua escolha.  P5. O administrador confirma e salva.  P6. O Administrador clica em sair. |
| **Pós-Condições** | A execução está realizada |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1. No P1 caso administrador não esteja cadastrado.  F2. O sistema volta a tela inicial.  Fluxo Alternativo B:  F1. o administrador pode informar seus dados errados.  F2. O sistema volta a tela inicial.  Fluxo Alternativo C:  F1. P5 o administrador pode executar o processo de maneira errada  F2. Assim o administrador irá repetir todo o processo. |

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-06** Criar, Editar e Excluir Agendamento |
| **Breve Descrição** | O administrador estar habilitado a criar, editar e excluir os Agendamento e detalhar em qual cooperativa estará o moto-táxi apresentado na corrida. |
| **Atores Envolvidos** | Administrado |
| **Pré-Condições** | O usuário ser administrador, possuir um cooperativa cadastrado e que possua o moto táxi. |
| **Fluxo Principal** | P1: efetuar login.  P2: é iniciado quando o administrador seleciona a opção de sua escolha.  Criar, editar e excluir Agendamento  P3: Após a seleção da atividade o sistema apresenta a interface para realizar a tarefa de sua escolha.  P4: realizada a tarefa de acordo com a escolha o administrador confirma a tarefa. |
| **Pós-Condições** | Apenas as corridas e as cooperativas cadastrados serão vistas pelos os usuários. |
| **Fluxo Alternativo** | Fluxo Alternativo A  F1: no P2, Se a escolha for editar agendamento, caso seja procurado e não encontrado, ele pode ter sido excluído.  F2: no P2, se a escolha for criar agendamento o moto táxi terá todos os seus atributos descritos.  Fluxo Alternativo B  F1: no P1, caso o usuário ou senha estiver incorreta, uma mensagem será apresentada, "usuário ou senha estão incorretos".  F2: no P3, se o moto táxi já estiver agendado, será enviada uma mensagem, moto táxi agendado, se for o mesmo agendamento mantenha cadastrado, caso não, alterar para agendamento atual.  F3: no P4, caso ocorra erro na hora de confirmar a agendamento será possível voltar e realizar uma nova execução. |

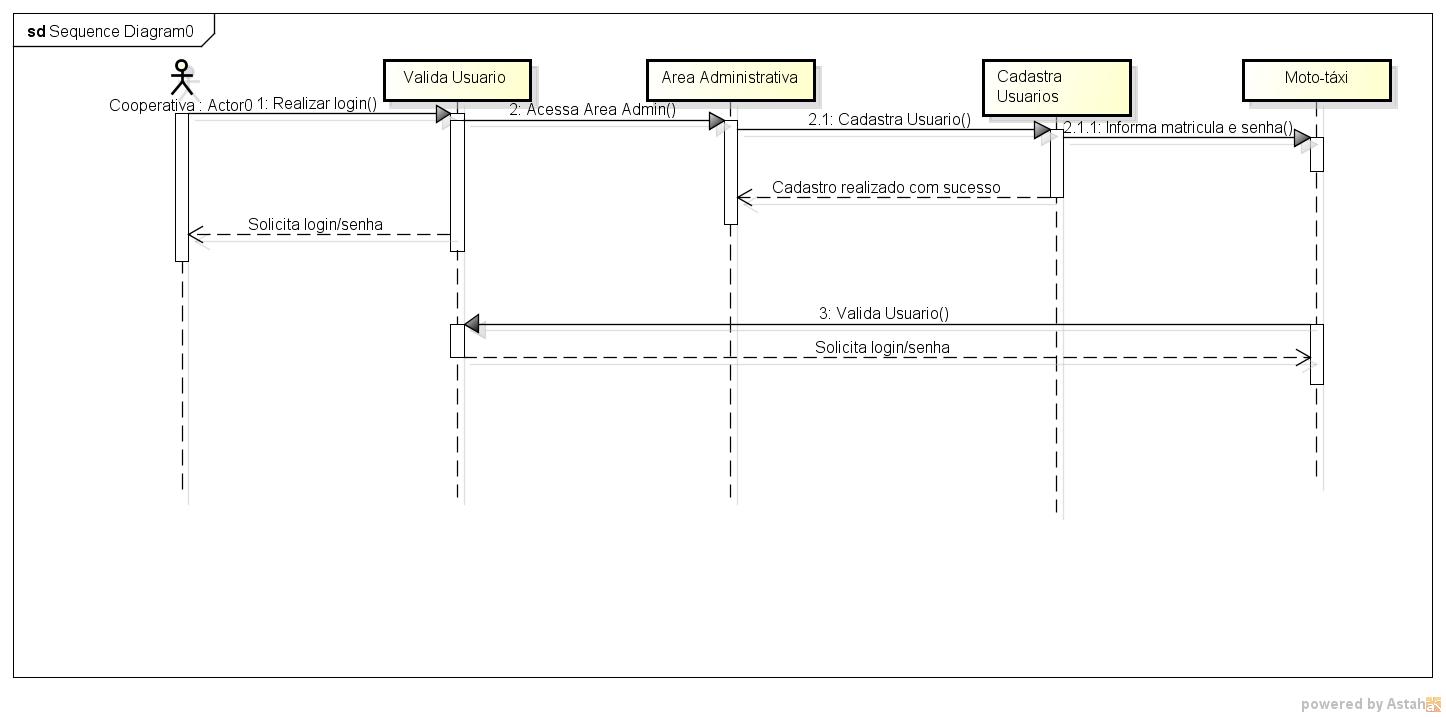
|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-07** - Efetuar login. |
| **Breve Descrição** | O administrador efetua o login para ter acesso ao sistema. |
| **Atores Envolvidos** | Administrador, usuário. |
| **Pré-Condições** | Estar cadastrado no sistema como administrador ou usuário. |
| **Fluxo Principal** | P1: digitar usuário.  P2: digitar senha.  P3: confirmar o login. |
| **Pós-Condições** | Após o login o administrador estará habilitado as funções do sistema. |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1: no P3 uma mensagem é apresentada no sistema "login efetuado".  F2: após confirmar o login, o sistema vai validar usuário e senha, se estiver incorreta será apresentada a mensagem "usuário ou senha inválida". |

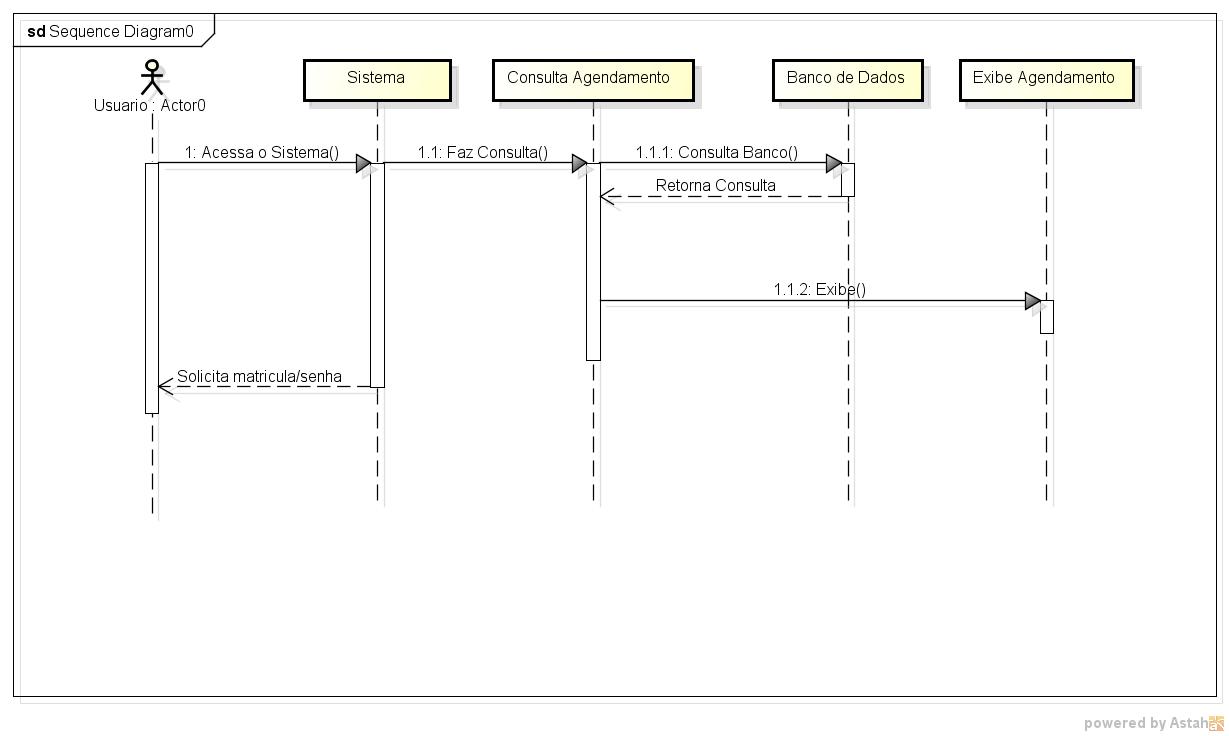
|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **CU-08**-Efetuar logoff |
| **Breve Descrição** | O administrador efetua o logoff para sair do sistema. |
| **Atores Envolvidos** | Administrador, usuário. |
| **Pré-Condições** | O usuário ter logado no sistema |
| **Fluxo Principal** | P1: o usuário utiliza o botão sair.  P2: confirmar o logof. |
| **Pós-Condições** | Ao confirmar o logof, o sistema bloqueia as funções e só é possível ter acesso efetuando o login. |
| **Fluxo Alternativo** | Fluxo Alternativo A:  F1: no P1 o usuário pode ter que confirmar algum caso para sair do sistema  F2: caso exista um erro no sistema que mantenha o usuário conectado, à exceção será tratada com uma mensagem "falha no logof". |

## Histórias de Usuários

|  |
| --- |
| Como usuário eu gostaria de pesquisar por corridas de moto-táxi |
| Como usuário eu gostaria de ver as cooperativas que estão ofertando corridas de moto-táxi |
| Como usuário eu gostaria de ver os moto-táxi da cooperativa X |
| Como administrador eu devo informar login e senha para ter acesso ao sistema administrativo |
| Como administrador eu devo cadastrar cooperativas no sistema para mostrar os agendamentos e moto-taxistas |
| Como administrador eu devo cadastrar moto-táxi de cooperativa X para exibi-los no sistema |
| Como administrador eu quero excluir cooperativa(s) do sistema para não mais exibi-lo |
| Como administrador eu quero excluir moto-táxi(s) da cooperativa X para não mais exibi-las no sistema |
| Como administrador eu preciso editar cooperativa X para atualizar seu dados cadastrais/informações |
| Como administrador eu preciso editar moto-táxi X para atualizar suas informações |

# Visão Lógica





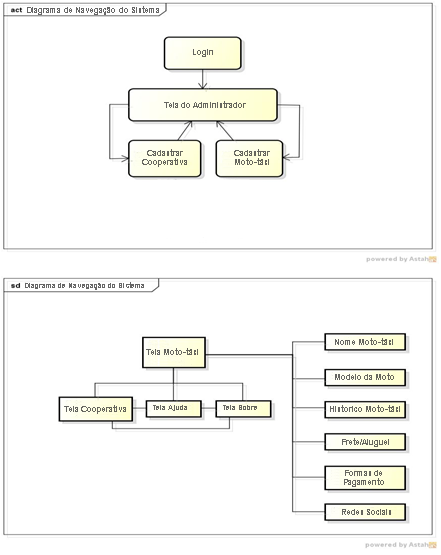
## Visão Geral

A visão lógica define a estrutura lógica da arquitetura. Ela consiste em descrever as camadas que compõem a aplicação definindo as responsabilidades de cada camada. Nesta seção, essa definição será ampliada para mencionar os frameworks que serão utilizados em cada uma das camadas e descrever a estrutura da arquitetura em relação aos padrões (design patterns) que devem ser utilizados e como deverão ser os seus relacionamentos.

Essas diferentes visões ajudarão a melhor entender a arquitetura ao mesmo tempo em que estabelecerão um forte padrão em termos de camadas, frameworks e design patterns que compõem a aplicação.

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

**Diagrama de navegação**



# Visão de Processos

O processo adotado para o desenvolvimento do software consistiu no estudo das metodologias ágeis de desenvolvimento de software, analisou se um conjunto de atividades e técnicas bem definidas e organizadas, cujo objetivo é transformar as necessidades do usuário em um Sistema de Software, o qual deve representar, de maneira precisa, soluções para essas necessidades. Com base em modelos de sequencia e metodologias ágeis de processo de software; Scrum, fundamentou-se em pontos que demonstram o fluxo de ações decorrentes para elaboração do sistema TransporLine, processo com uma abordagem especifica para o desenvolvimento de nosso sistema que se propõe a organizar e agilizar o agendamento das corridas.

Implementação

Plano de Projeto

Fase de elicitação e análise de requisitos

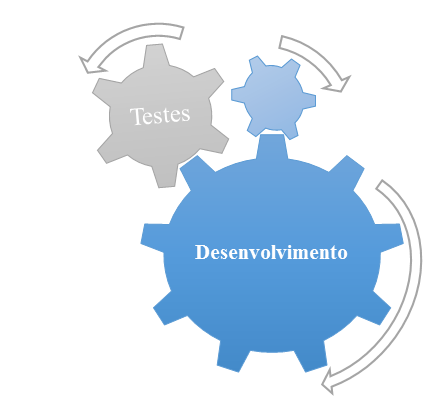
Fase de planejamento de atividades

Fase de teste de usabilidade

Entrega do sistema

**Modelagem do sistema**

**Validação de requisitos**



Documento da fase de análise

Documento da fase de Testes

**Release**

Documento de Projeto

# Visão de Implantação

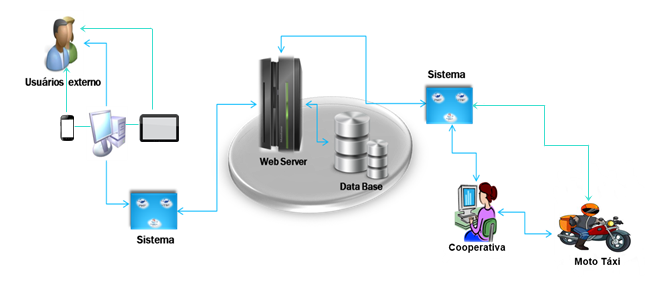
O sistema será baseado na arquitetura cliente-servidor como mostra a figura, no qual o sistema utilizando a plataforma Windows, se conecta utilizando a internet em um servidor remoto, atualizando suas informações semanalmente (ou sempre que for requisitado).

As informações serão inseridas no servidor pelos administradores pelo site na web que acessará diretamente o SGBD (Sistema de Gerenciamento de Banco de Dados) no servidor.

Utilizando um servidor com configurações moderadas, ele atenderá todas as requisições com fluidez, com até 500 pesquisas simultâneas e colocando em filas as que surgirem no mesmo instante, utilizando também de recursos de gerenciamento mais novos e atualizados para não depender somente do hardware, como o sistema de cache guardando informações também no dispositivo móvel.

O servidor funcionará em um Sistema operacional linux e um servidor apache, com configurações de firewall IPTables e banco de dados MySQL.

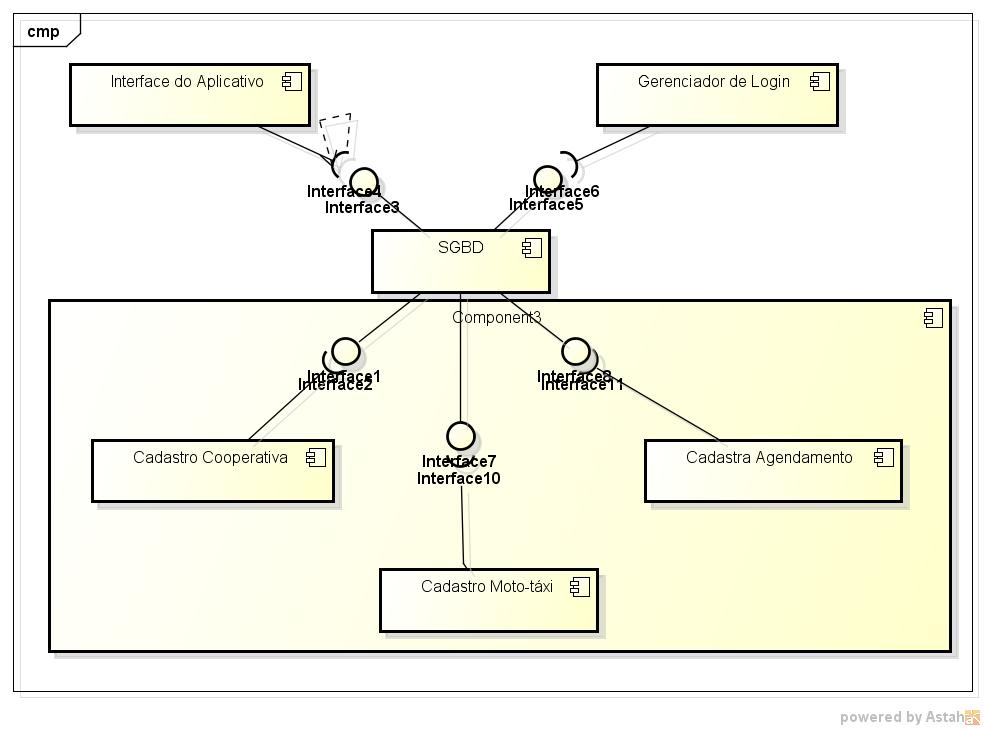
O Servidor contara com 1 Mbps (Mega bites por segundo) dedicado no inicio tendo possíveis alterações de acordo com a demanda, e na parte do usuário uma rede 3G de 128Kbps em pleno funcionamento suprirá as necessidades.



# Visão da Implementação

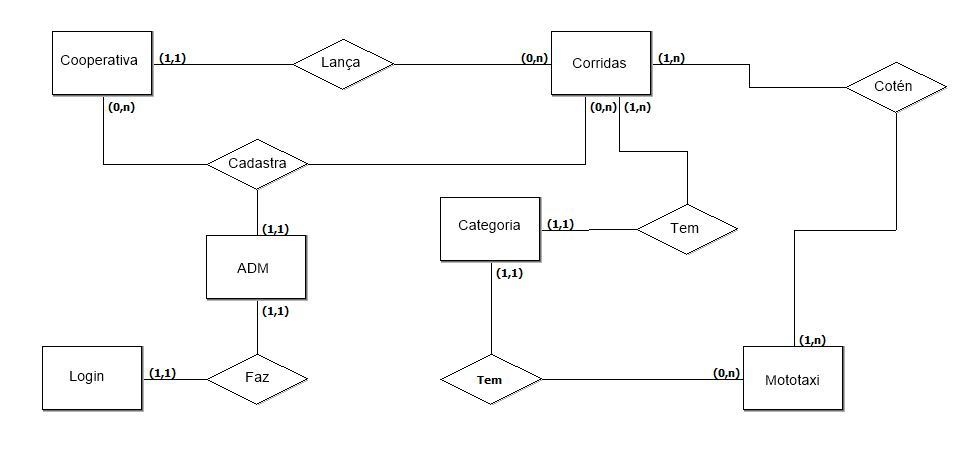
Nessa etapa será aplicado a documentação gerada da analise de requisitos onde o programador iniciara a fase de desenvolvimento do sistema (codificação), o desenvolvedor aplicara a linguagem PHP com aplicações de bootstrap no desenvolvimento da aplicação onde se tornara possível a compilação e geração do código-executável para o sistemas e fazer alguns testes, para realização dessa tarefa o desenvolvedor se utilizara da plataforma de desenvolvimento Netbeans com o auxilio de outras ferramentas de domínio do programador. Após iniciado o desenvolvimento, o banco de dados Mysql definido nas fases anteriores conter características que melhor atende a esse sistema é implantado estabelecendo uma conexão com os métodos codificados e os dados armazenados no banco, de forma a colaborar para que os processos do sistema sejam mais efetivos.

## Camadas

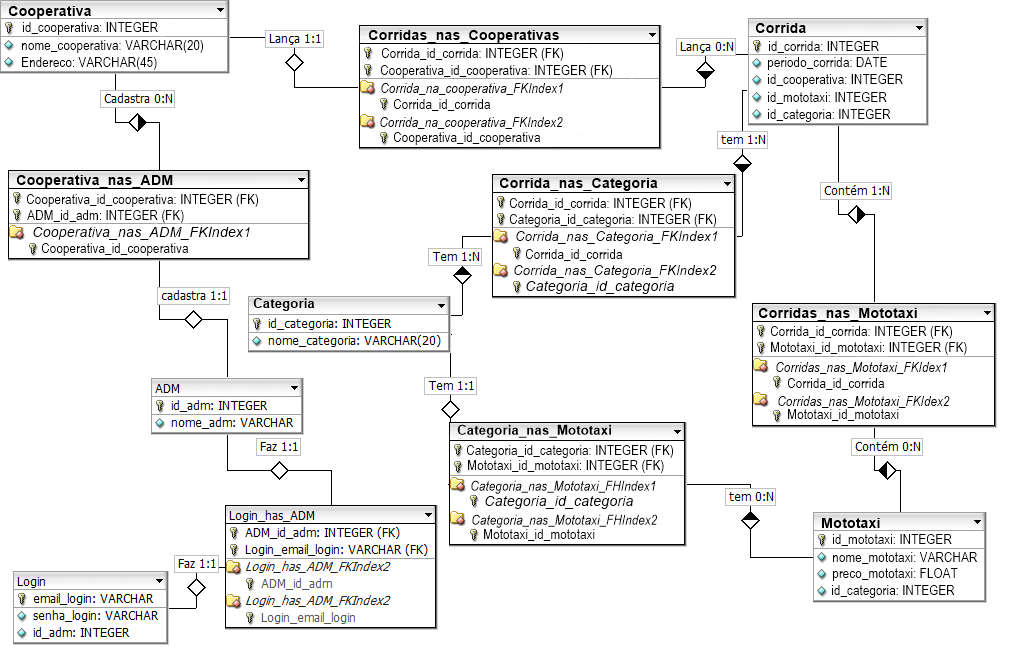


# Visão de Dados

### Modelo lógico da Base de Dados



**Modelo Lógico - TransporLine**



**Modelo Lógico – TransporLine**

### Criação Física do Modelo de Dados

CREATE TABLE ADM (

 id\_adm INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

 nome\_adm VARCHAR NULL,

 PRIMARY KEY(id\_adm) );

CREATE TABLE Categoria (

 id\_categoria INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

 nome\_categoria VARCHAR(20) NULL,

 PRIMARY KEY(id\_categoria) );

CREATE TABLE Categoria\_has\_Mototaxi (

 Categoria\_id\_categoria INTEGER UNSIGNED NOT NULL,

 Produto\_id\_produto INTEGER UNSIGNED NOT NULL,

 PRIMARY KEY(Categoria\_id\_categoria, Mototaxi\_id\_mototaxi),

 INDEX Categoria\_has\_Mototaxi\_FKIndex1(Categoria\_id\_categoria),

 INDEX Categoria\_has\_Mototaxi\_FKIndex2(Mototaxi\_id\_mototaxi) );

CREATE TABLE Login (

 email\_login VARCHAR NOT NULL AUTO\_INCREMENT,

 senha\_login VARCHAR NULL,

 id\_adm INTEGER UNSIGNED NULL,

 PRIMARY KEY(email\_login) );

CREATE TABLE Login\_has\_ADM (

 ADM\_id\_adm INTEGER UNSIGNED NOT NULL,

 Login\_email\_login VARCHAR NOT NULL,

 PRIMARY KEY(ADM\_id\_adm, Login\_email\_login),

 INDEX Login\_has\_ADM\_FKIndex2(ADM\_id\_adm),

 INDEX Login\_has\_ADM\_FKIndex2(Login\_email\_login) );

CREATE TABLE Mototaxi (

 id\_mototaxi INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

 nome\_mototaxi VARCHAR NULL,

 preco\_mototaxi FLOAT NULL,

 id\_categoria INTEGER UNSIGNED NULL,

 PRIMARY KEY(id\_mototaxi) );

CREATE TABLE Corrida (

 id\_corrida INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

 periodo\_corrida DATE NULL,

 id\_cooperativa INTEGER UNSIGNED NULL,

 id\_mototaxi INTEGER UNSIGNED NULL,

 id\_categoria INTEGER UNSIGNED NULL,

 PRIMARY KEY(id\_corrida) );

CREATE TABLE Corrida\_has\_Categoria (

 Corrida\_id\_corrida INTEGER UNSIGNED NOT NULL,

 Categoria\_id\_categoria INTEGER UNSIGNED NOT NULL,

 PRIMARY KEY(Corrida\_id\_corrida, Categoria\_id\_categoria),

 INDEX Corrida\_has\_Categoria\_FKIndex1(Corrida\_id\_corrida),

 INDEX Corrida\_has\_Categoria\_FKIndex2(Categoria\_id\_categoria) );

CREATE TABLE Corrida\_has\_Mototaxi (

 Corrida\_id\_corrida INTEGER UNSIGNED NOT NULL,

 Mototaxi\_id\_mototaxi INTEGER UNSIGNED NOT NULL,

 PRIMARY KEY(Corrida\_id\_corrida, Mototaxi\_id\_mototaxi),

 INDEX Corrida\_has\_Mototaxi\_FKIndex1(Corrida\_id\_corrida),

 INDEX Promocao\_has\_Produto\_FKIndex2(Mototaxi\_id\_mototaxi) );

CREATE TABLE Corrida\_has\_Cooperativa (

 Corrida\_id\_corrida INTEGER UNSIGNED NOT NULL,

 Cooperatva\_id\_cooperativa INTEGER UNSIGNED NOT NULL,

 PRIMARY KEY(Corrida\_id\_corrida, Cooperativa\_id\_cooperativa),

 INDEX Corrida\_has\_Cooperativa\_FKIndex1(Corrida\_id\_corrida),

 INDEX Corrida\_has\_Cooperativa\_FKIndex2(Cooperativa\_id\_Cooperativa) );

CREATE TABLE Cooperativa (

 id\_cooperativa INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT,

 nome\_cooperativa VARCHAR(20) NULL,

 Endereco VARCHAR(45) NULL,

 PRIMARY KEY(id\_cooperativa) );

CREATE TABLE Cooperativa\_has\_ADM (

 Cooperativa\_id\_cooperatva INTEGER UNSIGNED NOT NULL,

 ADM\_id\_adm INTEGER UNSIGNED NOT NULL,

 PRIMARY KEY(Cooperativa\_id\_cooperativa, ADM\_id\_adm),

 INDEX Cooperativa\_has\_ADM\_FKIndex1(Cooperativa\_id\_cooperativa) );

### Dicionário de Dados

|  |  |  |
| --- | --- | --- |
| **Entidade** | **Atributo** | **Relacionamento** |
| Cooperativa | Id\_cooperativa  Nome\_cooperativa  Endereço  Contato | - Com Corridas através de Lança 0:N  - com ADM através de cadastro 1:1 |
| Corrida | Id\_corrida  Período\_corrida  Id\_cooperativa  Id\_mototaxi  Id\_categoria | - com ADM através de cadastra 1:1  - com Categoria através de possui 1:N  - com Moto Táxi através de contém 1:N |
| Categoria | Id\_categoria  Nome\_categoria | - com Corrida através de possui 1:N  - com Moto Táxi através de possui 0:N |
| Mototaxi | Id\_mototaxi  Nome\_mototaxi  Preço\_mototaxi  Id\_categoria | - com Corrida através de contém 1:N  - com Categoria através possui 1:1 |
| ADM | Id\_ADM  Nome\_ADM | - com Corrida através de cadastra 0:N  - com Cooperativa através de cadastra 0:N  - com Login através de faz 1:! |
| Login | Email\_login  Senha\_login  Id\_ADM | - com ADM através de faz 1:1 |
| **Relacionamento** | **Atributo** | **Conexões** |
| Cadastra | Id\_ADM  Id\_corrida  Id\_cooperativa  Data\_cadastro | - Com Corrida 0:N  - Com ADM  1:1  - Com Cooperativa 0:N |

# Tamanho e Desempenho

A arquitetura de software escolhida suporta os principais requisitos de qualidade e prazo.

O sistema deve suportar até 1.000 usuários simultâneos utilizando o banco de dados central ao mesmo tempo e até 500 usuários simultâneos utilizando os servidores locais a qualquer momento.

O sistema fornecerá acesso ao banco de dados dos Moto-taxistas com um tempo de espera de até 10 segundos.

O sistema deve ser capaz de concluir 80% de todas as transações em 2 minutos.

A parte cliente deve requerer menos de 20 MB de espaço em disco e 32 MB de RAM.

A arquitetura selecionada suporta os requisitos de dimensionamento e velocidade pela implementação de uma arquitetura cliente/servidor. A parte cliente é implementada em PCs locais ou em PCs dial-up remotos. Os componentes foram projetados para assegurar que sejam necessários requisitos mínimos de disco e memória na parte cliente do PC.

# Qualidade

## Usabilidade

Terá uma interface limpa, amigável e intuitiva projetada para usuários de todos os níveis intelectuais com ícones representando suas funções.

## Confiabilidade

O hardware do sistema deve ser assegurado por um gerador.

A área administrativa do sistema deverá ser restrita, apenas o administrador terá acesso através de login e senha.

Não deve revelar aos operadores todos os dados pessoais do cliente ou administrador.

## Desempenho

O sistema deverá esta disponível 24 horas por dias, 7 dias por semana.