

# DOCUMENTO DE VISÃO DE PROJETO



## Histórico de Versões

Data	Versão	Descrição	Autor	Revisor
15/09/2018	1.0	Modelagem e desenvolvimento	Marcelo Andrade	
27/11/2018	1.1	Revisão do Documento	Marcelo Andrade	
28/11/2018	1.2	Revisão do Documento	Marcelo Andrade	
01/12/2018	1.3	Revisão de diagramas do documento	Marcelo Andrade	

**Cliente** Rodrigo Rocha Silva – FATEC MOGI DAS CRUZES - Interno

**Documento** Documento de Visão de Projeto: *HelpMe!*

**Data** 15 de setembro de 2018

**Autor** **Marcelo Andrade**  
marcelo.andrae@fatec.sp.gov.br

## Sumário

1. Objetivo .....	4
1.1. Escopo .....	4
1.2. Referências .....	4
2. Necessidades de negócio .....	4
3. Objetivo do projeto .....	4
4. Declaração preliminar de escopo .....	5
4.1. Descrição .....	5
4.2. Produtos a serem entregues .....	5
4.3. Requisitos .....	5
4.3.1. Requisitos Funcionais .....	5
4.3.2. Requisitos Não Funcionais .....	6
4.3.3. Regras de Negócio .....	7
5. Premissas .....	9
6. Influência das partes interessadas .....	9
7. Representação Arquitetural .....	9
7.1. Restrições arquiteturais .....	10
7.2. Objetivos e restrições Arquiteturais .....	10
8. Visão de Casos de Uso .....	10
8.1. Diagrama de caso de uso UC01 – Gerenciar chamados .....	11
8.2. Diagrama de caso de uso UC02 – Gerenciar Usuários .....	12
8.3. Descrição dos casos de usos Arquiteturalmente Significativos .....	12
8.3.1. Gerenciar chamados .....	12
8.3.2. Gerenciar usuários .....	12
8.4. Realização de caso de uso .....	12
8.4.1. Gerenciar Chamados – Abertura do Chamado .....	12
9. Visão de lógica .....	14
9.1. Camada de Apresentação – Frontend.dll .....	14
9.2. Camada de Negócio – Dominio.dll .....	15
9.3. Camada de Persistência – Core.dll – Namespace DAOs .....	17
9.4. Camada de Negócio – Core.dll .....	17
10. Visão de implementação .....	19
11. Visão de Dados .....	19
12. Cronograma .....	21
13. Índice de ilustrações .....	22
14. Referências .....	23

# 1. Objetivo

Este documento trata principalmente das necessidades de negócios da justificativa do projeto, do entendimento atual das necessidades do cliente e descreve resumidamente o novo produto, serviço ou resultado que deve satisfazer esses requisitos.

Tem o objetivo de alinhar as expectativas dos interessados e formalizar o início do projeto. Apresentar uma visão arquitetural do sistema *HelpMe!*. O intuito é salientar diferentes aspectos deste produto obtidos a partir de decisões arquiteturais realizadas no âmbito do sistema *HelpMe!*.

## 1.1. Escopo

O escopo deste documento trata do desenvolvimento de um sistema que atenda às necessidades de uma empresa que deseja implementar um sistema de gestão de chamados técnicos, abrangendo quaisquer setor/áreas de atendimento da empresa.

O escopo deste documento é documentar as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, mostra sua divisão em classes e utilitários de classe.

## 1.2. Referências

Para a construção deste documento foram utilizadas as seguintes referências:

- Apresentações e discussão sobre o tema proposto em Laboratório de Engenharia de Software

Este documento influencia os seguintes documentos:

- Documento de Requisitos

# 2. Necessidades de negócio

Um sistema informatizado para controle de atendimento de chamados técnicos, é necessário para que a empresa/instituição consiga controlar a quantidade de atendimentos realizados por suas áreas técnicas, a quantidade de insumos utilizados no atendimento, a qualidade do atendimento e do serviço prestado. E através de relatórios desses atendimentos identificar uma falha atual ou futura de acordo com a quantidade de chamados abertos por setor para determinado problema.

# 3. Objetivo do projeto

Desenvolver uma plataforma para soluções WEB capaz de:

- Armazenar informações em uma base de dados;
- Utilizar protocolo HTTP;
- Ser executado em qualquer navegador;
- Controlar serviços realizados pelos setores técnicos;
- Controlar saída de material em estoque utilizados em atendimentos prestados.

Minimizando a burocracia e melhorando o controle de entrada e saída de materiais para atendimento de serviços técnicos e facilitando a produção de relatórios com base em quantidade de chamados abertos e atendidos.

## 4. Declaração preliminar de escopo

Esta seção descreve, em alto nível, o escopo do projeto. Os requisitos serão melhor detalhados nos documentos de Requisitos.

### 4.1. Descrição

A plataforma *HelpMe!* possibilita o controle de chamados técnicos abertos e atendidos na instituição. Possibilitando saber qual tipo de serviço foi atendido, quais atividades foram realizadas e a quantidade de insumos gastos e cada atendimento.

### 4.2. Produtos a serem entregues

Os seguintes itens são considerados produtos do projeto em sua entrega:

- Sistema de atendimentos de chamados, implementado de acordo com a especificação feita na fase de análise.
- Documentos de especificação do sistema, concebidos na fase de elaboração,
- Hospedagem do sistema em ambiente 24x7
- 

### 4.3. Requisitos

É importante salientar que o documento de visão estará sendo feito no momento de iniciação do projeto. No qual, ainda não se tem todo o detalhamento. Portanto, os requisitos devem descritos em alto nível e depois detalhados em outro documento, caso o projeto seja aprovado.

#### 4.3.1. Requisitos Funcionais

Grupo: Cadastro de Usuários		
RF001	Cadastrar usuário	O Sistema deve manter um cadastro único para usuários
RF002	Inativação de usuário	O Sistema deve possibilitar a inativação de usuários quando solicitado
RF003	Alterar senha	O Sistema deverá permitir a alteração da senha do usuário sem alterar os dados cadastrais
RF004	Consulta de usuário	O Sistema deverá permitir a consulta de um usuário. Todos os campos do cadastro podem ser utilizados como filtro.
RF005	Atualização de dados cadastrais	O sistema deverá permitir a alteração de dados cadastrais.

Grupo: Cadastro de Tipo de Serviço		
RF006	Cadastrar Tipo de Serviço	O sistema deverá permitir e manter o cadastro de categorias de atendimento.

<b>RF007</b>	Cadastro de “Tipo” de atendimento	O sistema deverá permitir o cadastro de “Tipos” de atendimento. Estes, devem estar vinculados a apenas uma subcategoria
<b>RF008</b>	Desativação de categoria	O sistema deverá permitir a desativação uma categoria quando solicitado.

#### Grupo: Controle de Estoque

<b>RF010</b>	Cadastrar produto	O sistema deverá permitir e manter o cadastro de produtos em estoque
<b>RF011</b>	Dar baixa de item em estoque	O sistema deverá permitir que um produto tenha baixa quando utilizado em um atendimento via ticket.

#### Grupo: Tickets de Atendimento

<b>RF012</b>	Cadastrar Ticket	O sistema deverá permitir que o usuário cadastre um ticket para atendimento técnico independente de sua função e departamento cadastrada no sistema.
<b>RF013</b>	Atualizar Ticket	O Sistema deverá permitir que o usuário que cadastrou e o técnico de atendimento, tenha acesso a visualizar e atualizar com comentários o seu pedido de atendimento
<b>RF014</b>	Encerrar Ticket	O sistema deverá permitir que o técnico encerre o ticket independente de sua fase de atendimento (com sucesso ou não). Se finalizado com sucesso, deverá conter os serviços realizados selecionados em uma lista preenchida automaticamente de acordo com a categoria do chamado.
<b>RF015</b>	Reabertura de ticket	O sistema deverá permitir a reabertura do ticket pelo usuário em 3 dias após a atribuição do status “Resolvido” caso não concorde com a resolução da equipe técnica

#### 4.3.2. Requisitos Não Funcionais

##### Grupo: Geral

<b>RNF001</b>	Tempo de resposta para consultas	O sistema deverá ter resposta de no máximo 1 segundo.

##### Grupo: Cadastro de usuários

<b>RNF002</b>	Senha forte	A senha cadastrada pelo usuário deve ser composta de pelo menos 8 caracteres, ter letras maiúsculas e minúsculas e numerais
<b>RNF003</b>	Confirmação de senha	O usuário obrigatoriamente deve digitar duas vezes a mesma senha no momento do registro da mesma.
<b>RNF004</b>	Código de cliente	Todo cliente cadastrado deve receber um código único no sistema.

##### Grupo: Cadastro de Tipos de Serviço

<b>RNF005</b>	Cadastro de Domínio	Os departamentos responsáveis pelo tipo de serviço a ser cadastrado, devem ser cadastrados através de um script de implantação
<b>RNF006</b>	Código de Tipo de Serviço	Todo tipo de serviço deve ter um código único no sistema

#### Grupo: Cadastro de Estoque

<b>RNF007</b>	Cadastro de Domínio	Os departamentos responsáveis pelo tipo de serviço a ser cadastrado, devem ser cadastrados através de um script de implantação
<b>RNF008</b>	Código de Estoque	Todo item de estoque cadastrado deve receber um código único no sistema.

#### Grupo: Gerenciamento de Chamados

<b>RNF010</b>	Diferenciar status por cores	No sinótico de chamados, o sistema deverá exibir com cores diferentes a contagem de status de ticket. Vermelho – Violados Azul – Resolvidos Verde – Não Violados Preto – Total de chamados abertos
---------------	------------------------------	--

#### Grupo: Notificações

<b>RNF011</b>	Diferenciar notificações por cores	Em todas as telas quando uma notificação for exibida, deverá ser diferenciada de um erro e mensagem de sucesso com as seguintes cores: Verde – Sucesso Vermelho - Erro
---------------	------------------------------------	--

### 4.3.3. Regras de Negócio

#### Grupo: Cadastro de usuários

RN001	Gestão de acesso	Usuários do sistemas deverão ter uma permissão configurada no perfil. O utilizador poderá ter uma das seguintes funções:						
		Função		Abrir chamado	Atendimento Técnico	Criar Categoria	Gerenciar Usuário	Visualizar Relatórios
		Administrador	Sim	Sim	Sim	Sim	Sim	
		Técnico	Sim	Sim	Não	Não	Não	
Usuário	Sim	Não	Não	Não	Não			
RN002	Dados obrigatórios	Para todo usuário cadastrado, é obrigatório dos seguintes dados: Nome, sobrenome, gênero, rg, departamento, função, email, senha e o endereço composto de CEP, rua, número, bairro, complemento, cidade e estado.						
RN003	Senha forte	A senha cadastrada pelo usuário deve ser composta de pelo menos 8 caracteres, ter letras maiúsculas e minúsculas além de conter caracteres especiais.						
RN005	Confirmação de senha	O usuário obrigatoriamente deve digitar duas vezes a mesma senha no momento do registro da mesma.						
RN006	Código de cliente	Todo cliente cadastrado deve receber um código único no sistema.						

#### Grupo: Cadastro de Categorias

<b>RN007</b>	Definição de prioridade	Para cada tipo de atendimento, deverá ter uma prioridade cadastrado em sua composição.
<b>RN008</b>	Definição de tipo de atendimento	Ao cadastrar uma categoria, o sistema deverá receber a informação se o atendimento será um incidente ou uma requisição

## Grupo: Gerenciamento de Tickets

<b>RN010</b>	Definir status do Ticket	<p>O ticket deverá um dos seguintes status:</p> <p><b>Atribuído</b> – Quando um ticket for aberto ou encaminhado entre equipes, deverá ficar como “Atribuído”</p> <p><b>Em Resolução</b> – Quando um técnico passar a atender o ticket, deverá ter recebido o status “Em resolução”</p> <p><b>Pendente</b> – Ao passar para esse status, deverá ser incluído um motivo para essa alteração, onde deverá ser exibido nos comentários do chamado. Podendo ser reativado qualquer momento pelo usuário/técnico.</p> <p><b>Resolvido</b> – O ticket deverá ter esse status ao ser finalizado pelo técnico, podendo ser reaberto pelo utilizador e voltará no status “Atribuído”</p> <p><b>Fechado Resolvido</b> – O ticket será fechado quando passar o prazo de 3 dias e não tenha sido reativado pelo utilizador após “Resolvido”.</p> <p><b>Cancelado</b> – O ticket foi cancelado pelo técnico ou <u>utilizador</u>. Não podendo ser reaberto.</p>																		
<b>RN011</b>	Feedback de atendimento	Para cada chamado finalizado pelo usuário técnico, o usuário que abriu o ticket, deverá receber um formulário de feedback. O sistema deverá solicitar uma nota para o atendimento de 0 a 10 e um comentário para o atendimento.																		
<b>RN012</b>	Cálculo de SLA	<p>O SLA deverá ser calculado de acordo com a Prioridade do chamado aberto.</p> <table border="1"> <thead> <tr> <th>Código de Prioridade</th><th>Descrição</th><th>Prazo para solução</th></tr> </thead> <tbody> <tr> <td>1</td><td>Crítico</td><td>8 Hora</td></tr> <tr> <td>2</td><td>Alto</td><td>16 Horas</td></tr> <tr> <td>3</td><td>Médio</td><td>24 Horas</td></tr> <tr> <td>4</td><td>Baixo</td><td>48 Horas</td></tr> <tr> <td>5</td><td>Planejado</td><td>Planejado</td></tr> </tbody> </table> <p>O sistema deverá exibir no rodapé do ticket aberto, a data prevista para violação do SLA. Também deverá ser atualizado automaticamente o status de “OK” para “Violado” quando a data de previsão ter sido alcançada e o chamado não estar com o status de “Resolvido”, “Cancelado” ou “Fechado Resolvido”.</p>	Código de Prioridade	Descrição	Prazo para solução	1	Crítico	8 Hora	2	Alto	16 Horas	3	Médio	24 Horas	4	Baixo	48 Horas	5	Planejado	Planejado
Código de Prioridade	Descrição	Prazo para solução																		
1	Crítico	8 Hora																		
2	Alto	16 Horas																		
3	Médio	24 Horas																		
4	Baixo	48 Horas																		
5	Planejado	Planejado																		
<b>RN013</b>	Histórico de alterações	<p>O sistema deverá manter em cada chamado um histórico de acordo com suas alterações.</p> <p>Ex: Houve a mudança de status de atribuído para em resolução, o sistema deverá registrar a hora e o utilizador que realizou essa alteração.</p>																		
<b>RN014</b>	Obrigatoriedade de comentários	Só será permitida a mudança de status caso haja algum comentário inserido no sistema.																		
<b>RN015</b>	Justificativa para tickets violados	Para todo ticket que tiver o SLA violado, deverá ser obrigatório a justificativa.																		



<b>RN016</b>	Código único de ticket	Todo ticket cadastrado deverá ter um código único associado a ele.
<b>RN017</b>	Encerramento de tickets resolvidos	Após 3 dias com o chamado no status “Resolvido” o sistema deverá fechar automaticamente o ticket e atualizar seu status para “Fechado Resolvido”, não podendo ser mais editado pelo utilizador do sistema.

## 5. Premissas

- O projeto será orientado pelo professor Rodrigo Rocha;
- As documentações devem ser revisadas e atualizadas constantemente.

## 6. Influência das partes interessadas

- Professor Rodrigo Rocha Silva, acompanhará o desenvolvimento do sistema, indicando pontos de correção e melhorias;
- O discente Marcelo de Moraes Andrade responsável pelo desenvolvimento do sistema, deverá realizar as alterações necessárias e se possível implementar novas funcionalidades ou melhorias.

## 7. Representação Arquitetural

Os sistemas serão desenvolvidos tendo como base a arquitetura ilustrada na Figura 1. Toda a arquitetura será baseada nos padrões de projetos tradicionais do GoF e também nos padrões *.Net Application* sendo executados dentro de um Servidor de Aplicações.

A Figura 1 representa o modelo arquitetural genérico utilizado no desenvolvimento desse sistema:

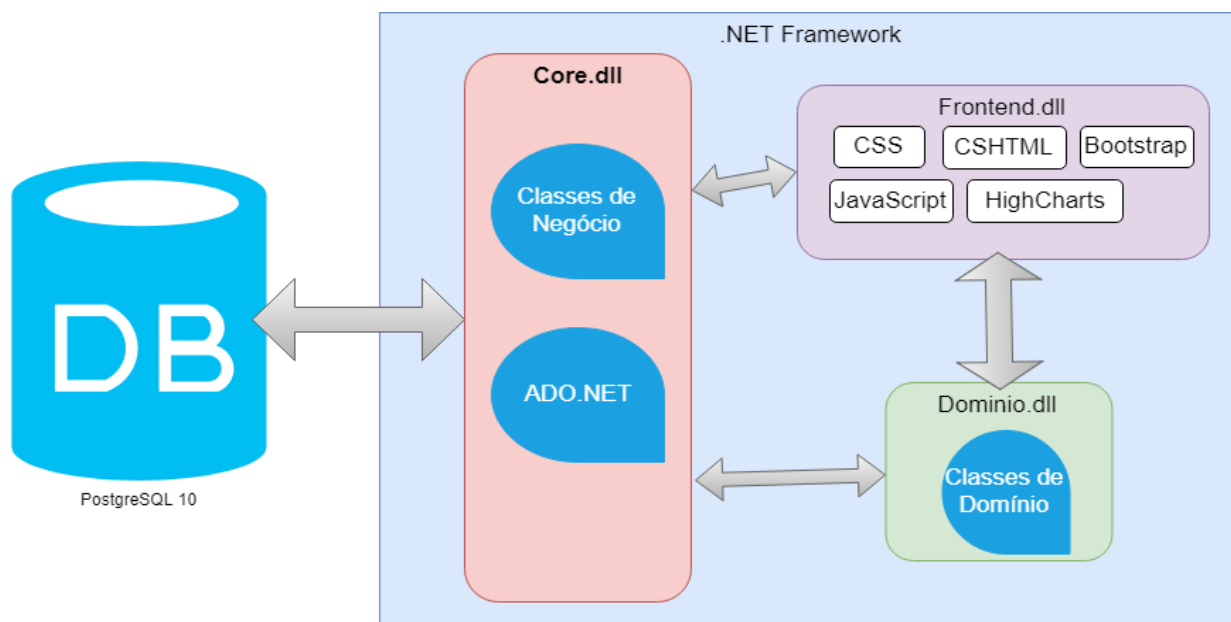


Figura 1 Modelo Arquitetural Genérico

A dll Frontend contém as classes responsáveis por exibir e possibilitar a interação com o usuário. Nessa camada são utilizadas as tecnologias CSHTML com Razor Framework, CSS, Bootstrap para estilização das telas, Javascript para dinamismo e HighCharts para exibição de análises realizadas em modo gráfico;

A dll Domínio contém as classes de representação do domínio da aplicação, contendo apenas construtores e atributos de acesso a dados;

A dll Core contém toda a lógica dos sistema, com regras para validação, as classes para manipulação dos dados inseridos pelo usuário, controle de acesso aos dados e gerenciamento das conexões ao banco de dados;

## 7.1. Restrições arquiteturais

Fora identificadas restrições pertinentes ao desenvolvimento dessa aplicação:

- Utilização de C# na versão do Framework 4.6.1;
- Utilização do SGBD PostgreSQL 10;
- Utilização de páginas web utilizando CSHTML com o Framework Razor;
- Utilização do Servidor Web IIS;

## 7.2. Objetivos e restrições Arquiteturais

Considerando os requisitos não funcionais descritos, pode-se citar as seguintes restrições:

Utilização da linguagem C#;

Considerar a utilização de Software livre quando possível;

# 8. Visão de Casos de Uso

Esta seção apresenta os Casos de Uso arquiteturalmente significativos, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema *HelpMe!*

## 8.1. Diagrama de caso de uso UC01 – Gerenciar chamados

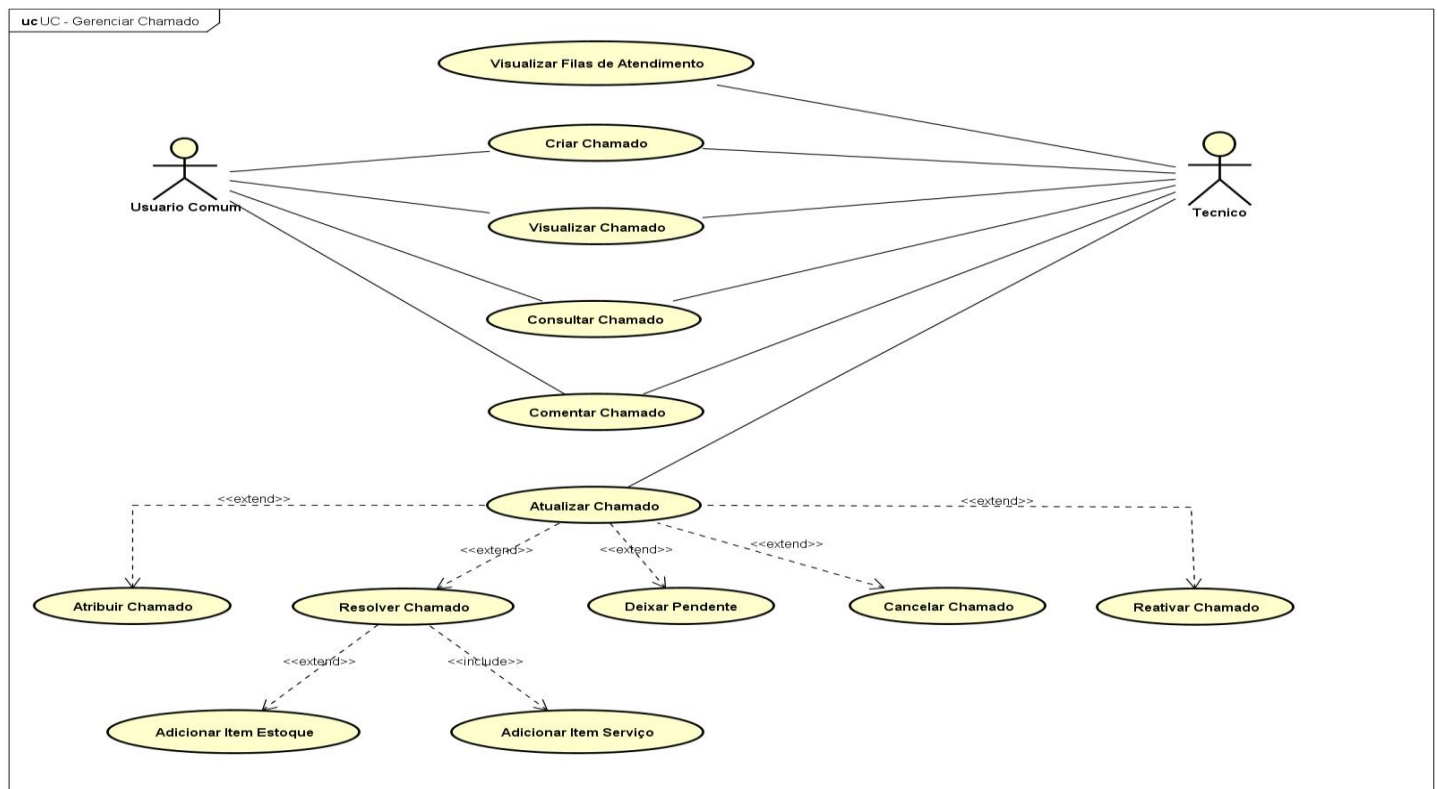


Figura 2 UC01 - Gerenciar Chamados

## 8.2. Diagrama de caso de uso UC02 – Gerenciar Usuários

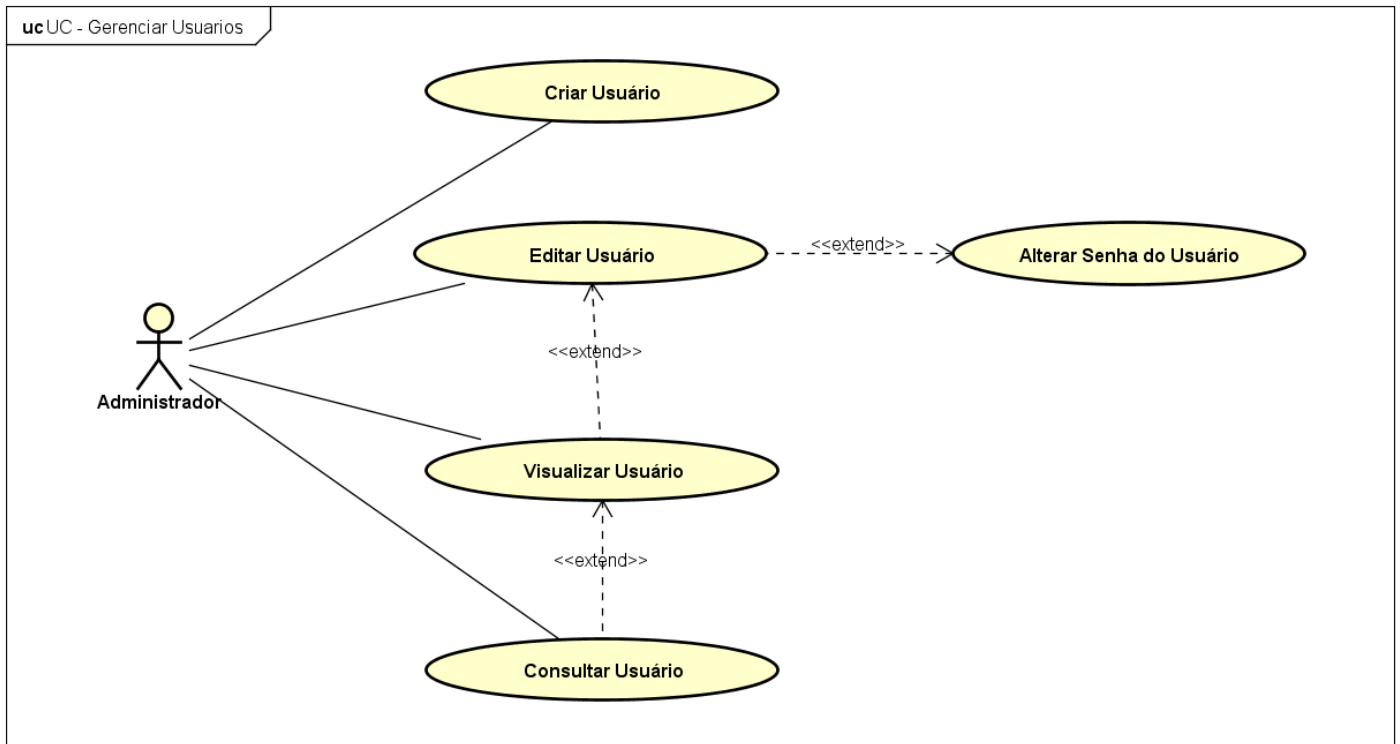


Figura 3 UC02 - Gerenciar Usuários

## 8.3. Descrição dos casos de usos Arquiteturalmente Significativos

### 8.3.1. Gerenciar chamados

Este caso de uso se inicia após o usuário logado na plataforma, iniciar um pedido de serviço. Passando pela suas fases de atendimento (novo, em resolução, pendente, resolvido e cancelado)

### 8.3.2. Gerenciar usuários

Este caso de uso se inicia após o usuário logado na plataforma, solicitar uma alteração de senha, cadastrar, consultar e/ou editar um novo usuário na plataforma.

## 8.4. Realização de caso de uso

### 8.4.1. Gerenciar Chamados – Abertura do Chamado

A figura 4 mostra a realização do caso de uso Gerenciar Chamado na atividade de “Abrir um chamado”



## 9. Visão de lógica

Esta visão apresenta elementos de design significativos do ponto de vista da arquitetura, descrevendo a organização do Sistema *HelpMe!* em pacotes, bem como a organização desses pacotes em camadas. A figura 5 representa a visão da organização dos pacotes .

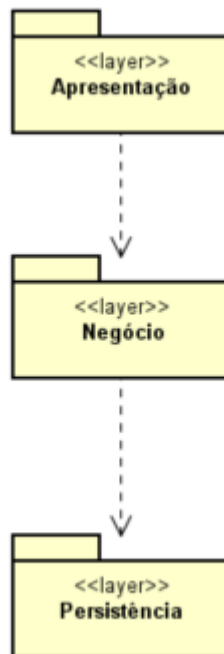


Figura 5 Organização dos pacotes do HelpMe!

**Apresentação:** Contém classes para as interfaces gráficas com os usuários (GUI). Através destas interfaces os usuários conseguem interagir com o *HelpMe!*, com o intuito de incluir, alterar e excluir produtos.

**Negócio:** Contém classes que controlam a execução das funcionalidades do *HelpMe!*.

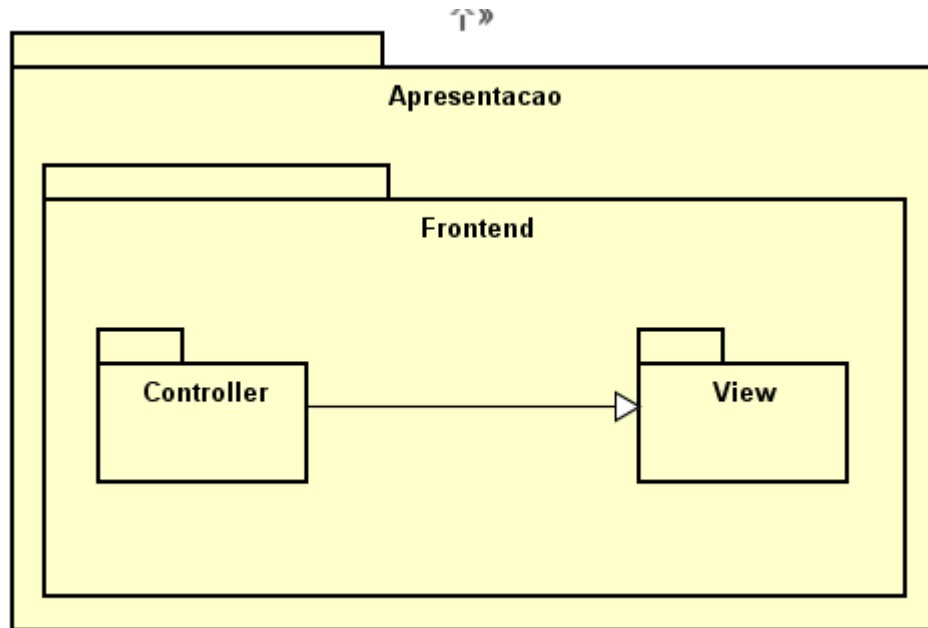
**Persistência:** Contém classes responsáveis por persistir as entidades de modelo. Por exemplo, contém as classes que permitem ler e gravar os objetos no banco de dados relacional.

Neste momento, é importante ressaltar que a camada de apresentação envolve componentes que são executados na porção servidora e na porção cliente. Na porção servidora, são executados os componentes que montam as páginas html/cshtml de resposta e controlam o fluxo de interação com o usuário. Assim, envolve interpretação de páginas cshtml, utilização de helpers para montagem das páginas de resposta, delegação de solicitação para a camada de negócio, obtenção de respostas da camada de negócio, dentre outros. Já na porção cliente, estão os códigos javascript que são executados nos navegadores dos usuários com o intuito de facilitar a interação. Por exemplo, podem fazer uma pré-verificação dos dados digitados (como o preenchimento de campos obrigatórios), solicitação de páginas via javascript, de modo a tornar mais dinâmica a interação com o usuário, dentre outros.

### 9.1. Camada de Apresentação – Frontend.dll

Nesta camada, temos o pacote form que contém todos os arquivos relacionados à exibição de informações para usuário, o que engloba páginas CSHTML, imagens, javascript, dentre outros. Já o pacote

de controle desta camada, contém as Actions responsáveis pela comunicação com as classes da camada de negócio. Estes pacotes podem ser vistos na figura 6



*Figura 6 - Representação do relacionamento dos pacotes da camada de apresentação*

## 9.2. Camada de Negócio – Dominio.dll

Nesta camada, temos o pacote dominio que contém as classes responsáveis por representar os objetos mapeados. A figura 7 exibe os relacionamentos das classes de modelo do sistema entre si. A principal entidade do sistema é a entidade "Ticket":

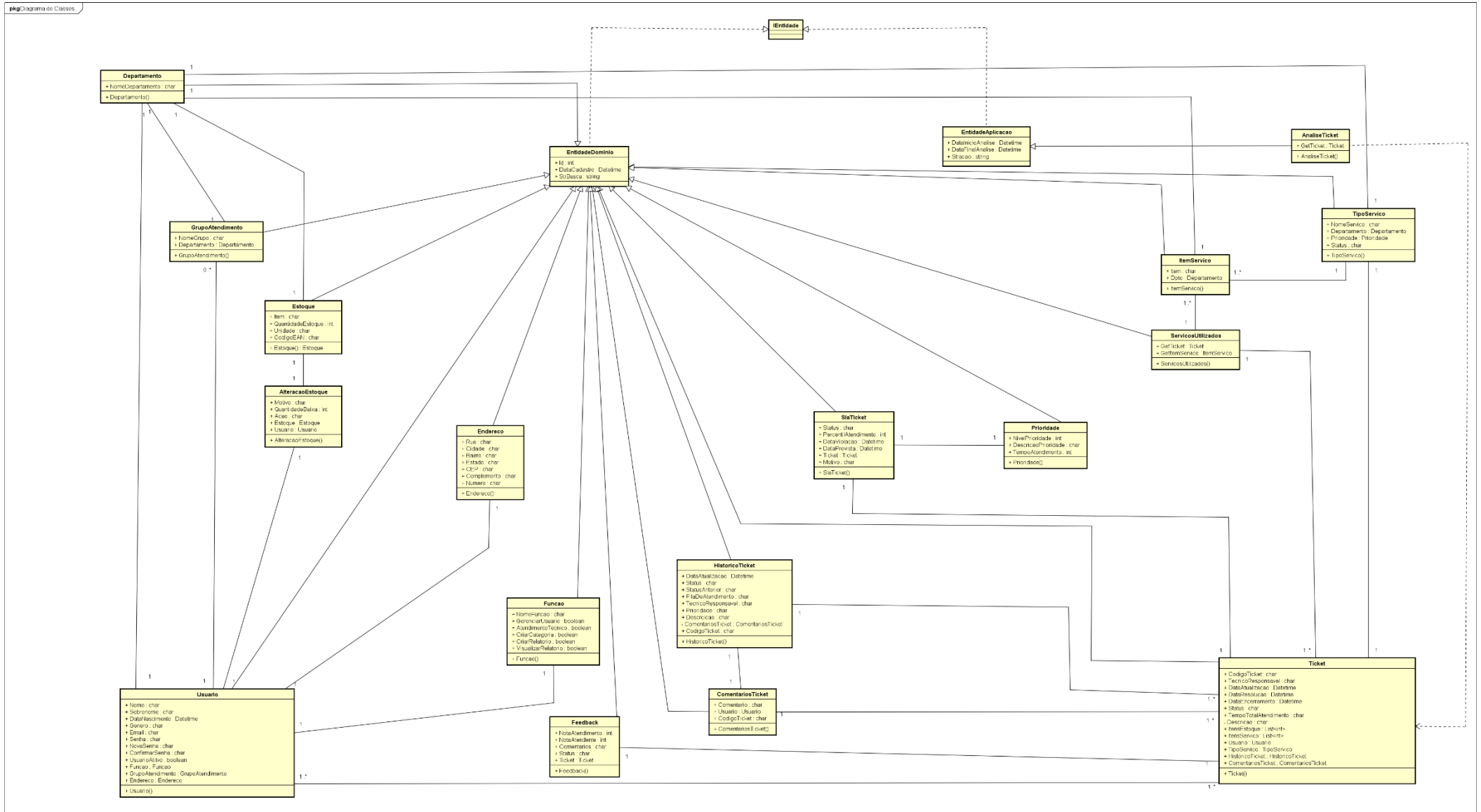
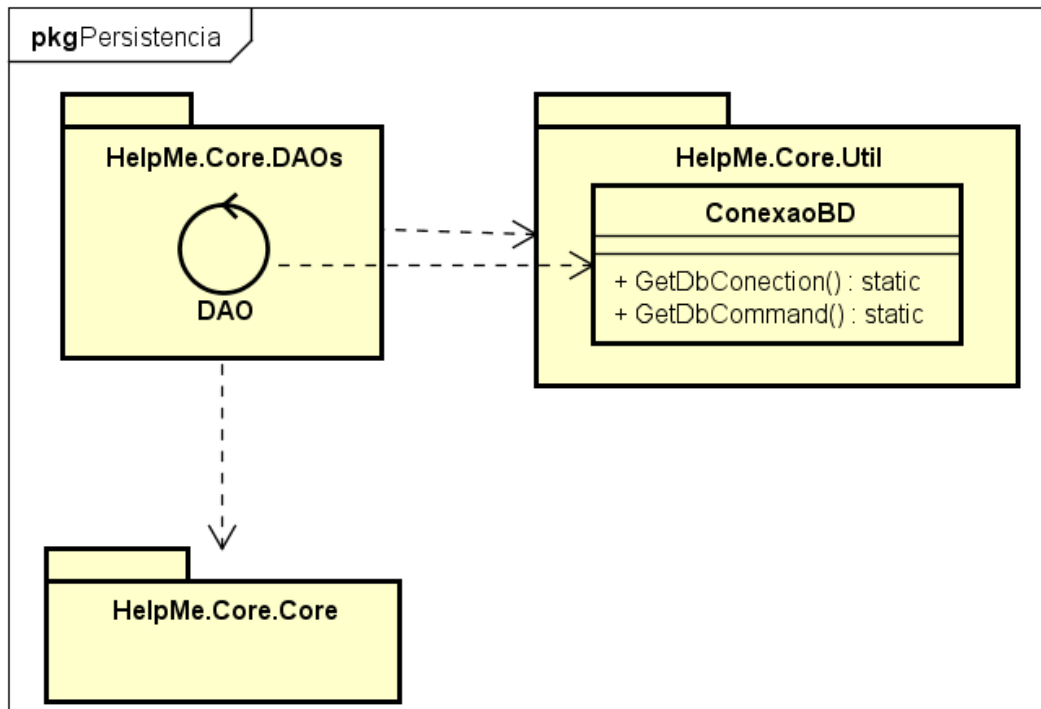


Figura 7 Diagrama de Classes de domínio - Helpme



### 9.3. Camada de Persistência – Core.dll – Namespace DAOs

Essa camada, é responsável por acesso aos dados. Na figura 8, é exibida a representação da comunicação entre os pacotes do sistema.



powered by Astah

Figura 8 Comunicação entre pacotes

### 9.4. Camada de Negócio – Core.dll

O pacote core contém as classes de controle do negócio. Na figura 9 temos as classe de implementação e suas dependências:

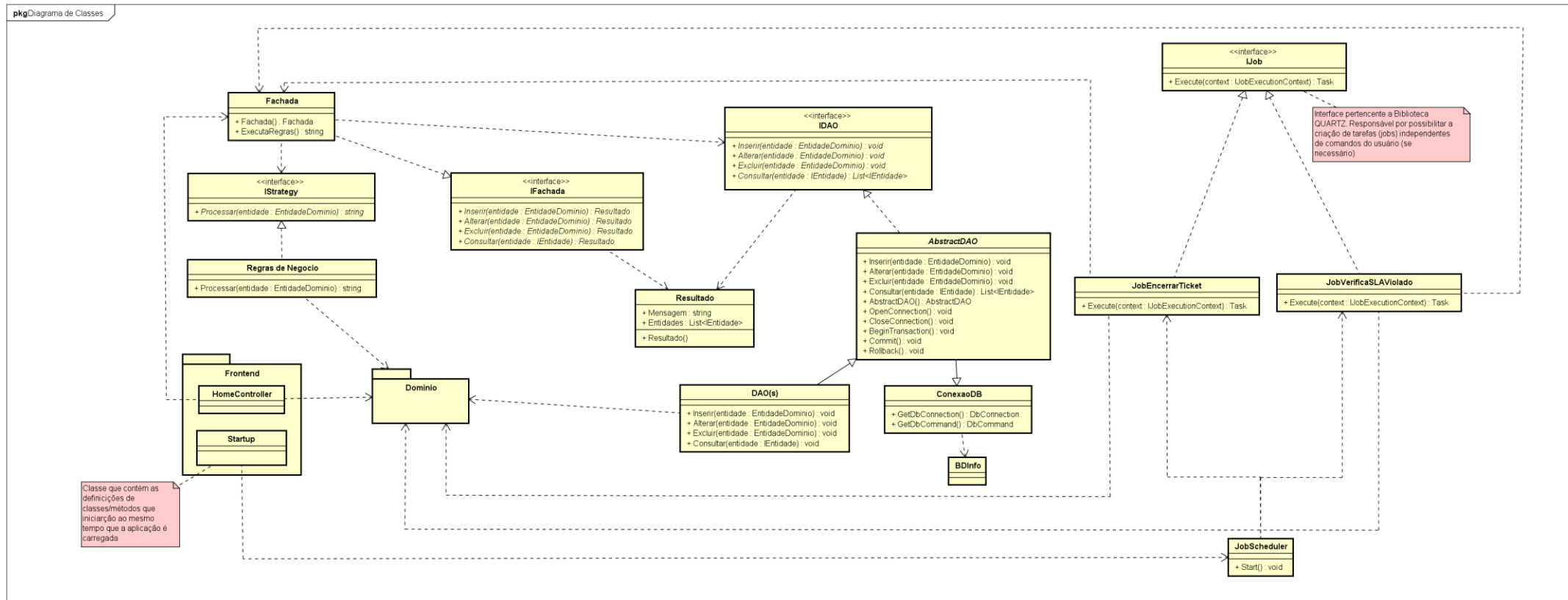
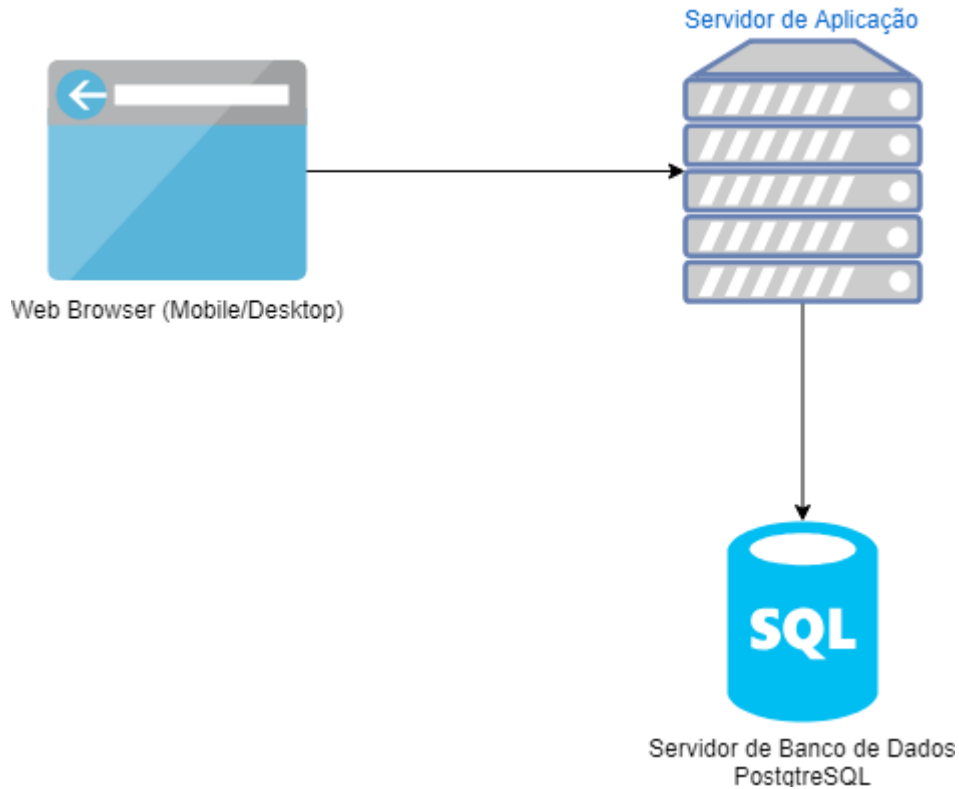


Figura 9 Diagrama de classes de implementação

## 10. Visão de implementação

Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

A figura 10, ilustra como é realizada a implantação do HelpMe, representando como é feito seu acesso via um navegador, a necessidade de um servidor de aplicação e um servidor de banco de dados.



*Figura 10 Implantação do HelpMe!*

## 11. Visão de Dados

O mecanismo de persistência utilizado no sistema HelpMe utiliza o banco de dados Relacional PostgreSQL juntamente com o framework de acesso a dados ADO.NET.

A figura demonstra o modelo físico do banco de dados do sistema.

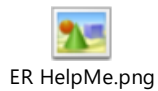
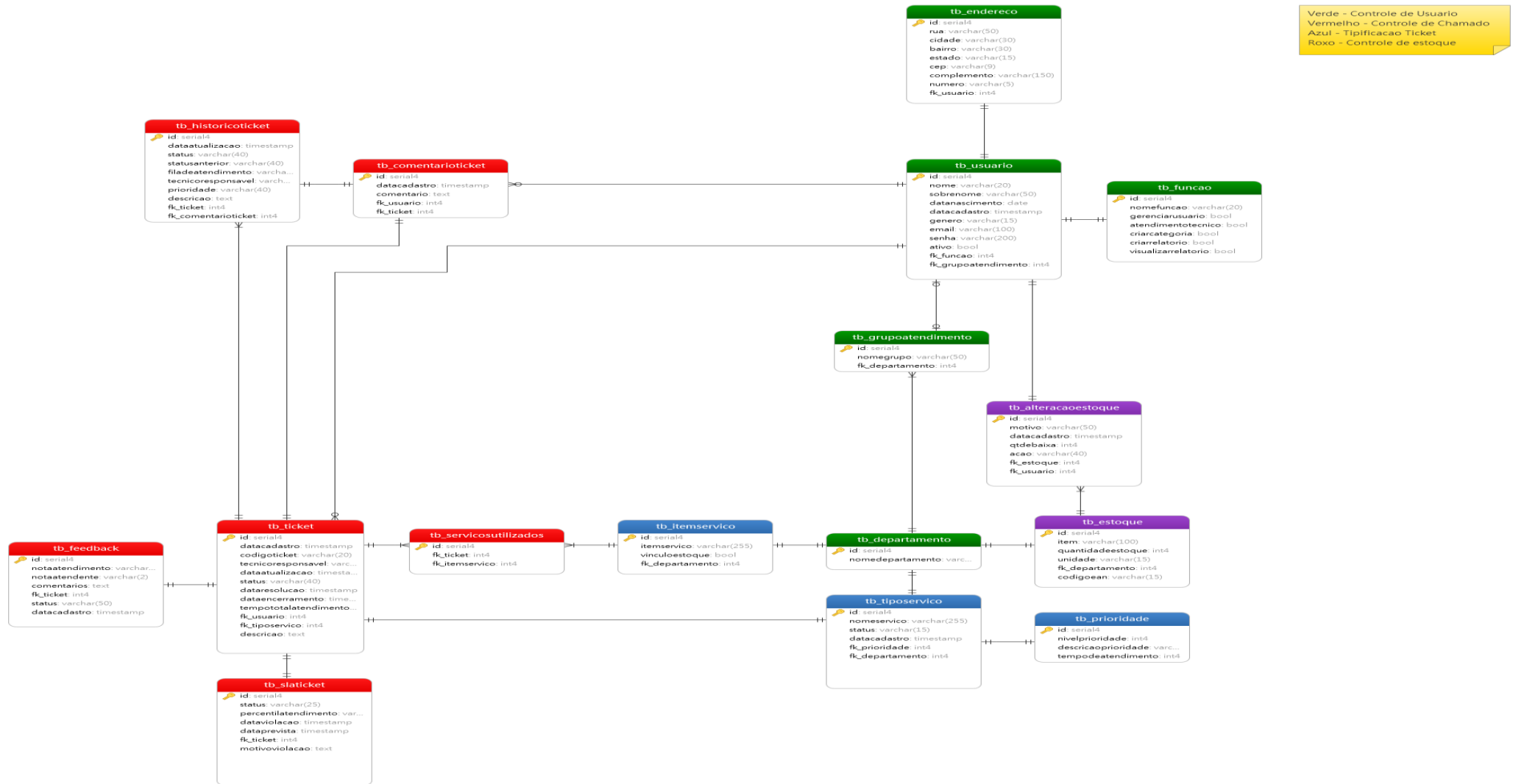


Figura 11 - Modelo Físico – HelpMe!

← Clique no PNG para abrir em melhor resolução

## 12. Cronograma

Data	Descrição da Atividade
20/08/2018	DRS - Documento de Requisito de Software
27/08/2018	Implementação de CRUD na arquitetura definida
03/09/2018	Protótipo de todo o sistema, com navegabilidade e deve ser incluído gráficos de análise
17/09/2018	Implementação de um novo CRUD na arquitetura definida
24/09/2018	DVP - Documento de Visão de Projeto
01/10/2018	Implementação de um novo CRUD na arquitetura
22/10/2018	Implementação da condução
12/11/2018	Especificação de caso de uso de condução e Especificação de Casos de Teste
12/11/2018	Apresentação do sistema como um todo e a Implementação da análise
26/11/2018	Apresentação final com proposta técnica-comercial
02/11/2018	Entrega da documentação final

## 13. Índice de ilustrações

Figura 1 Modelo Arquitetural Genérico .....	9
Figura 2 UC01 - Gerenciar Chamados .....	11
Figura 3 UC02 - Gerenciar Usuários .....	12
Figura 4 - Diagrama de Sequências - Abrir Chamado .....	13
Figura 5 Organização dos pacotes do HelpMe! .....	14
Figura 6 - Representação do relacionamento dos pacotes da camada de apresentação .....	15
Figura 7 Diagrama de Classes de domínio - Helpme.....	16
Figura 8 Comunicação entre pacotes .....	17
Figura 9 Diagrama de classes de implementação .....	18
Figura 10 Implantação do HelpMe!.....	19
Figura 11 - Modelo Físico – HelpMe!.....	20

## 14. Referências

Gerenciamento de Incidentes – ITIL, DevMedia - <https://www.devmedia.com.br/gerenciamento-de-incidentes-til/7174>, acesso em 21/08/2018 às 14:47;

Apostila básica ITIL Melhores Práticas – Training – Disponível em [http://www.training.com.br/download/Fundamentos%20ITIL%20Portugues%20Brasil%20BR%20Completo\(1\).doc](http://www.training.com.br/download/Fundamentos%20ITIL%20Portugues%20Brasil%20BR%20Completo(1).doc), acesso em 21/08/2018 às 15:00;

ITIL Service Management (ITSM) in a Nutshell – Oasis - Disponível em [https://www.oasis-open.org/committees/download.php/10737/DCML\\_ITIL\\_Service\\_Management\\_Summary.doc](https://www.oasis-open.org/committees/download.php/10737/DCML_ITIL_Service_Management_Summary.doc), acesso em 21/08/2018 às 15:00;

Ebook ITIL V3 Fundamentos. Filho, Felicio Cestari. –Escola Superior de Redes, Rio de Janeiro 2012. Disponível em <https://pt.scribd.com/doc/50809607/ITIL-v3-Fundamentos>, acesso em 21/08/2018 às 15:10;

Documento de Visão de Projeto – Disponibilizado pelo professor Rodrigo Rocha Silva;