



Universidade Federal Rural de Pernambuco

Bacharelado em Sistemas de Informação

Disciplina: Análise e Projeto de Sistemas de Informação

Docente: Rodrigo

Aluna: Thays Melo de Moraes

Diagramas do Projeto Personal Virtual Stylist

Recife, Julho 2014

Sumário

Problema do Cliente	3
Descrição Geral do Sistema	3
Cenário Encontrado	3
Requisitos do Sistema	4
Requisitos Funcionais	4
<i>[RF01] Logar no Sistema</i>	4
<i>[RF02] Consultar Dicas</i>	5
<i>[RF03] Cadastrar Roupas</i>	6
<i>[RF04] Pesquisar Roupas</i>	7
<i>[RF05] Editar Informações da roupa</i>	7
<i>[RF06] Excluir Roupa</i>	8
<i>[RF07] Cadastrar Funcionário</i>	9
<i>[RF08] Pesquisar Funcionários</i>	10
<i>[RF09] Editar Informações da Funcionário</i>	11
<i>[RF10] Excluir Funcionário</i>	11
<i>[RF11] Cadastrar Notas das Roupas</i>	12
Requisitos Não-Funcionais	13
Diagrama de Casos de Uso	13
Diagrama das Classes de Análise	14
Diagrama de Sequência	23
Subsistemas	28
Camadas	31
Diagrama de Pacotes	32
MVC	35
Padrões de Projeto	36
Singleton	36
Façade.....	36
Oserver	36
Prototipação da Interface	39
Fluxo de Telas	39
Protótipos das Telas	40
SOA	47
Banco de Dados	49

Personal Virtual Stylist

Problema do Cliente

O cliente solicitou a solução do seguinte problema:

“Tenho notado que muitas pessoas perguntam para as outras pessoas próximas a ela se aquela roupa fica bem nela. Com isso tive a ideia: Porque não ter alguém que diga a essa pessoa se a roupa o que ela quer comprar combina com ela? Mas ter uma pessoa falando isso sairia caro, logo porque não tem um programa que faça isso?”

Eu quero um programa que diga pra pessoa se aquela roupa que ela deseja comprar combina com ela. ”

Descrição Geral do Sistema

O projeto Personal Virtual Stylist, é um sistema que oferece um aval quanto as peças de vestuários escolhidas pelo cliente, sendo assim um software interativo, tendo em vista que, o sistema terá como parâmetro o peso, a altura e o tipo corpóreo, não esquecendo o estilo que é determinado de acordo com a preferência de cada indivíduo. Propiciando uma experiência nova e agradável no momento das compras, maior confiança quanto ao grau de combinação das suas peças e bem-estar em favor do relacionamento cliente-loja.

Cenário Encontrado

Cena: O cliente procura por uma roupa que viu no catálogo.

Agentes: Cliente, atendente

Cliente: Eu gostaria de comprar uma roupa que vi no catálogo da loja.

Atendente: Você está com o catálogo?

Cliente: Não.

Atendente: Como é a roupa.

Cliente: É uma calça masculina não sei se Skinny ou reta, jeans, cor azul marinho ou preto. Será que tem tamanho 40?

O atendente procura pelo departamento de roupas masculinas, e busca a parte onde se encontra à venda as calças. A partir das calças ele busca pela calça que o cliente descreveu. Encontra uma parecida e mostra ao cliente. O cliente decide ficar com a calça.

Requisitos do Sistema

Essencial:

- *Logar no Sistema*
- *Consultar Dicas*
- *CRUD – Roupas*
- *CRUD Funcionário*
- *Adicionar Notas das Roupas*

Importante:

- *Interface gráfica simples*
- *Armazenamento de dados em nuvem*
- *Acesso de dados remotamente*

Desejável:

- *Sistema estável*
- *Sistema de rápido acesso*

Requisitos Funcionais

[RF01] Logar no Sistema

Prioridade: Essencial

Descrição do requisito funcional: O sistema deve apresentar ao usuário uma tela de “Login” para permitir acesso a janela principal do sistema.

Pré-condição: O usuário tem que inserir login e senha corretos. A autenticação do login só deve ocorrer para usuário já cadastrado.

Pós-condição: A janela principal será exibida para o usuário.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário informa os dados necessários para o login.
 - Login do usuário;
 - Senha do usuário;
2. O sistema verifica se os dados do login são válidos.

3. O sistema carrega a janela principal de acordo com as informações do usuário.
4. A janela de login é fechada.
5. A janela principal é exibida com as funções disponíveis.

Fluxo secundário:

1. No passo 2 do fluxo principal, se a verificação for inválida, o sistema deve apresentar uma caixa de diálogo com a mensagem: “*Usuário ou senha inválidos!*” e retorna para o passo 1 do fluxo principal de eventos.

[RF02] Consultar Dicas

Prioridade: Essencial

Descrição do requisito funcional: O sistema deve apresentar uma janela onde o usuário entrará com as informações consultar dicas de moda.

Pré-condição: Nenhuma

Pós-condição: As roupas indicadas para a pessoa aparecerão na tela do sistema.

Usuário: Usuário, Cliente, Funcionário

Fluxo principal do evento:

1. O usuário informa os dados para receber as dicas.
 - Altura;
 - Peso;
 - Tipo Corpóreo;
 - Estilo.
2. O sistema verifica as informações digitadas.
3. O sistema busca no sistema as roupas que são consideradas adequadas de acordo com os parâmetros informados.

4. O sistema retorna o resultado das roupas consideradas ideais para o usuário.

Fluxos secundários:

1. No passo 1 do fluxo principal, o usuário pode voltar e digitar novamente os parâmetros solicitados.
2. Em qualquer momento a operação pode ser cancelada.

[RF03] Cadastrar Roupas

Prioridade: Essencial

Descrição do requisito funcional: O sistema deve apresentar uma janela onde o usuário entrará com as informações para o cadastro da roupa.

Pré-condição: A roupa não deve ter sido cadastrada no sistema. O usuário deverá estar *logado* no sistema.

Pós-condição: Os dados da roupa é validada e inserida no sistema.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário informa os dados da roupa a ser cadastrado.

- Código;
- Descrição;
- Tamanhos disponíveis;
- Quantidade;

2. O sistema verifica as informações digitadas.

3. O sistema inicia o [RF04]

4. Os dados da roupa são inseridos no banco de dados do sistema.

Fluxos secundários:

1. No passo 3 do fluxo principal, se a roupa já existir no banco de dados, ou seja, já for cadastrado, o sistema mostrará a mensagem “*Roupa já cadastrada!*” e retornará ao passo 1 do fluxo principal.
2. Em qualquer momento a operação pode ser cancelada.

[RF04] Pesquisar Roupas

Prioridade: Essencial

Descrição do requisito funcional: O sistema verificará se determinada roupa está armazenada no banco de dados.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: A roupa é retornada ao caso de uso principal.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário entra com um dos seguintes dados da roupa:
 - a. Nome;
 - b. Código.
2. O sistema acessa o banco de dados.
3. O sistema verifica se há alguma roupa cadastrada a partir da informação digitada.
4. Caso seja encontrado algum registro no banco de dados, o sistema alerta o usuário informando que “*Roupa já cadastrada*”.

Fluxo secundário

1. No caso 2 do fluxo principal, se as informações não forem encontradas, o sistema mostrará a mensagem “*Roupa não cadastrado*”.

[RF05] Editar Informações da roupa

Prioridade: Essencial

Descrição do requisito funcional: O sistema realiza a modificação dos dados da roupa cadastrada no sistema.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: Os novos dados são inseridos no sistema (banco de dados).

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário inicia a [RF04].
2. O usuário seleciona a opção “*Editar dados*”.
3. O usuário informa os novos dados.
4. Os novos dados do usuário são inseridos no sistema.

Fluxo secundário

1. Em qualquer momento a operação pode ser cancelada.

[RF06] Excluir Roupa

Prioridade: Essencial

Descrição do requisito funcional: O sistema apresenta ao usuário uma janela onde ele pode selecionar sua exclusão cadastral do sistema.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: A roupa é excluída do sistema.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário inicia a [RF04].
2. O sistema mostrará uma mensagem com a pergunta “*Deseja realmente excluir este item?*”.

3. A roupa tem seu cadastro do sistema.

Fluxos secundários

1. No passo 2 do fluxo principal, se o usuário responder “*Não*” o sistema retorna para a sua tela principal.

[RF07] Cadastrar Funcionário

Prioridade: Essencial

Descrição do requisito funcional: O sistema deve apresentar uma janela onde o usuário entrará com as informações para o cadastro de um outro Funcionário.

Pré-condição: LO funcionário não deve ter sido cadastrado no sistema. O usuário deverá estar *logado* no sistema.

Pós-condição: Os dados do da Funcionário é validada e inserida no sistema.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário informa os dados da Funcionário a ser cadastrado.

- Código;
- Nome;
- Login
- Senha;
- Tipo

2. O sistema verifica as informações digitadas.

3. O sistema inicia o [RF08]

4. Os dados da Funcionário são inseridos no banco de dados do sistema.

Fluxos secundários:

1. No passo 3 do fluxo principal, se a Funcionário já existir no banco de dados, ou seja, já for cadastrado, o sistema mostrará a mensagem “*Funcionário já cadastrado!*” e retornará ao passo 1 do fluxo principal.
2. Em qualquer momento a operação pode ser cancelada.

[RF08] Pesquisar Funcionários

Prioridade: Essencial

Descrição do requisito funcional: O sistema verificará se determinado Funcionário está armazenado no banco de dados.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: A Funcionário é retornado ao caso de uso principal.

Usuário: Funcionário

Fluxo principal do evento:

5. O usuário entra com um dos seguintes dados da Funcionário:
 - a. Código;
 - b. Nome;
 - c. Código.
6. O sistema acessa o banco de dados.
7. O sistema verifica se há alguma Funcionário cadastrado a partir da informação digitada.
8. Caso seja encontrado algum registro no banco de dados, o sistema alerta o usuário informando que “*Funcionário já cadastrado*”.

Fluxo secundário

1. No caso 2 do fluxo principal, se as informações não forem encontradas, o sistema mostrará a mensagem “*Funcionário não cadastrado*”.

[RF09] Editar Informações da Funcionário

Prioridade: Essencial

Descrição do requisito funcional: O sistema realiza a modificação dos dados da Funcionário cadastrada no sistema.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: Os novos dados são inseridos no sistema (banco de dados).

Usuário: Funcionário

Fluxo principal do evento:

5. O usuário inicia a [RF08].
6. O usuário seleciona a opção “*Editar dados*”.
7. O usuário informa os novos dados.
8. Os novos dados do usuário são inseridos no sistema.

Fluxo secundário

1. Em qualquer momento a operação pode ser cancelada.

[RF10] Excluir Funcionário

Prioridade: Essencial

Descrição do requisito funcional: O sistema apresenta ao usuário uma janela onde ele pode selecionar sua exclusão cadastral do sistema.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: A Funcionário é excluído do sistema.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário inicia a [RF08].

2. O sistema mostrará uma mensagem com a pergunta “*Deseja realmente excluir este item?*”.
3. A Funcionário tem seu cadastro “excluído” do sistema.

Fluxos secundários

1. No passo 2 do fluxo principal, se o usuário responder “*Não*” o sistema retorna para a sua tela principal.

[RF11] Cadastrar Notas das Roupas

Prioridade: Essencial

Descrição do requisito funcional: O sistema apresenta ao usuário uma janela onde ele pode selecionar a roupa e adicionar uma pontuação a ela.

Pré-condição: O usuário deve estar *logado* no sistema.

Pós-condição: A pontuação é adicionada para a roupa no sistema.

Usuário: Funcionário

Fluxo principal do evento:

1. O usuário inicia a [RF08]
2. O usuário entra com o seguinte dado:
 - a. Nota
3. O sistema verifica se há alguma nota cadastrada.
4. A nota é cadastrada no sistema.

Fluxos secundários

1. No passo 3 do fluxo principal, se *for encontrado* uma nota já registrada no banco de dados, o sistema alerta o usuário informando que “*Nota já Adicionada*”.

Requisitos Não-Funcionais

- Usabilidade
- Confiabilidade
- Desempenho
- Segurança

Diagrama de Casos de Uso

A seguir será mostrado os diagramas de casos de uso de acordo com os requisitos funcionais citados anteriormente. Foi desenvolvido dois diagramas onde um o usuário é o Funcionário da loja e o outro é o Cliente da loja onde ele irá receber as dicas de moda.

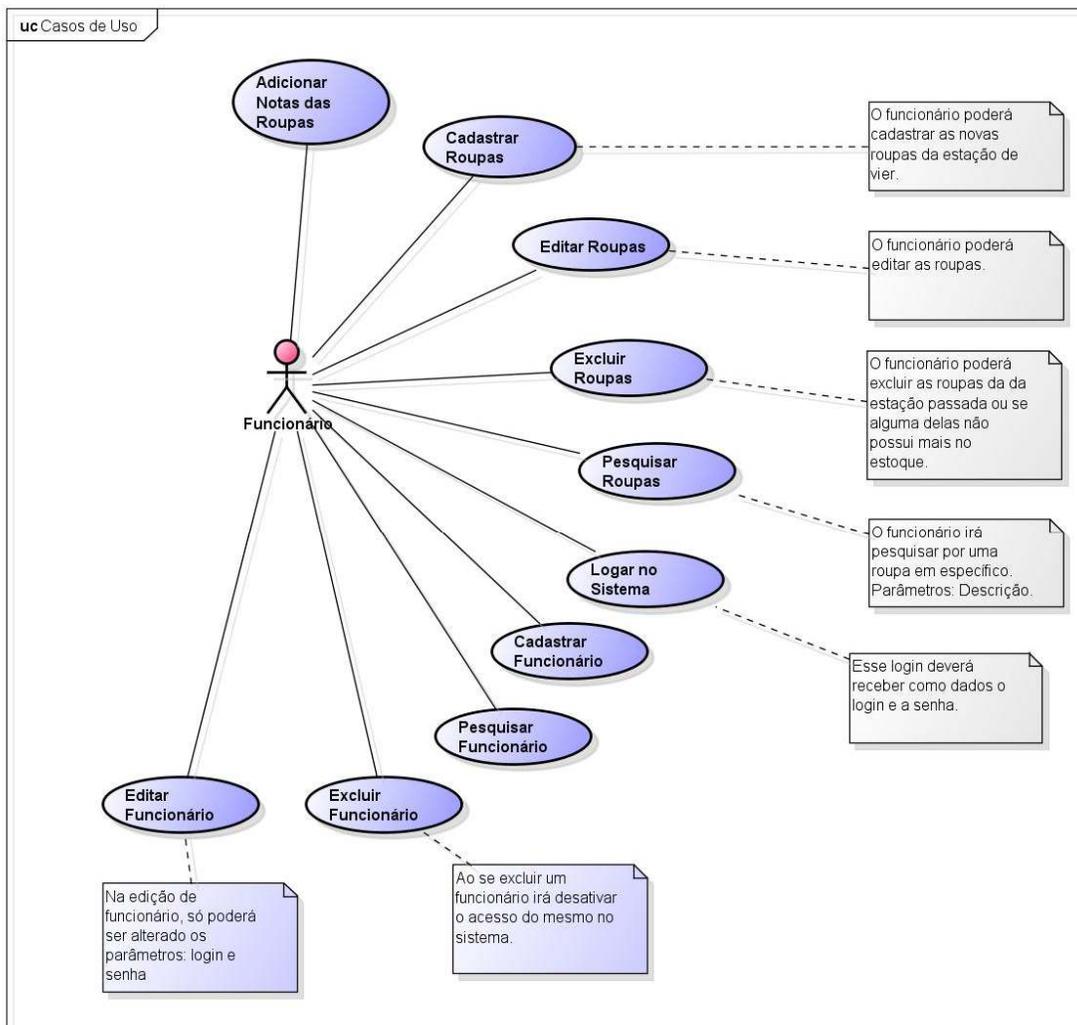
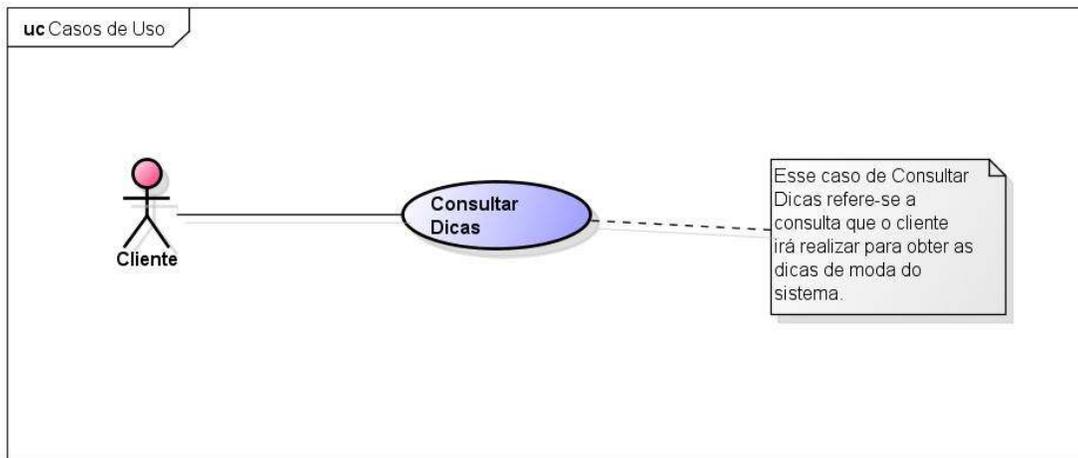


Figura 1 - Diagrama de Casos De Uso Usuário Funcionário



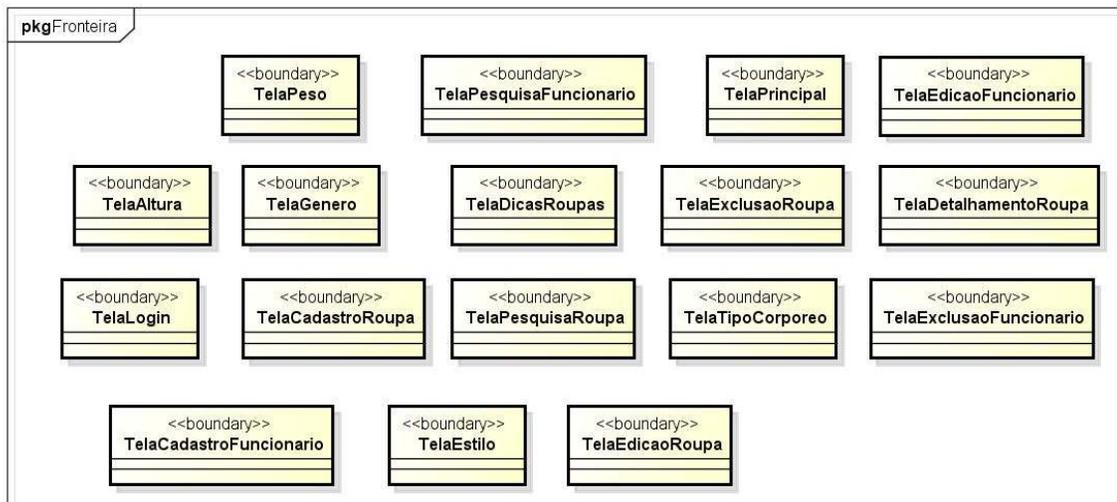
powered by Astah

Figura 2 - Diagrama de Casos De Uso Usuário Cliente

Diagrama das Classes de Análise

As classes de Análise é o primeiro passo para modelar um sistema. Ele representa a estrutura estática do sistema. Assim são em um nível abstrato mais alto e durante a fase de planejamento e modelagem do diagrama de classe do projeto geralmente sofrem modificações. Existem três tipos de classes de análise: Fronteira, Entidade e Controle. A seguir será mostrado a primeira etapa da modelagem dessas classes.

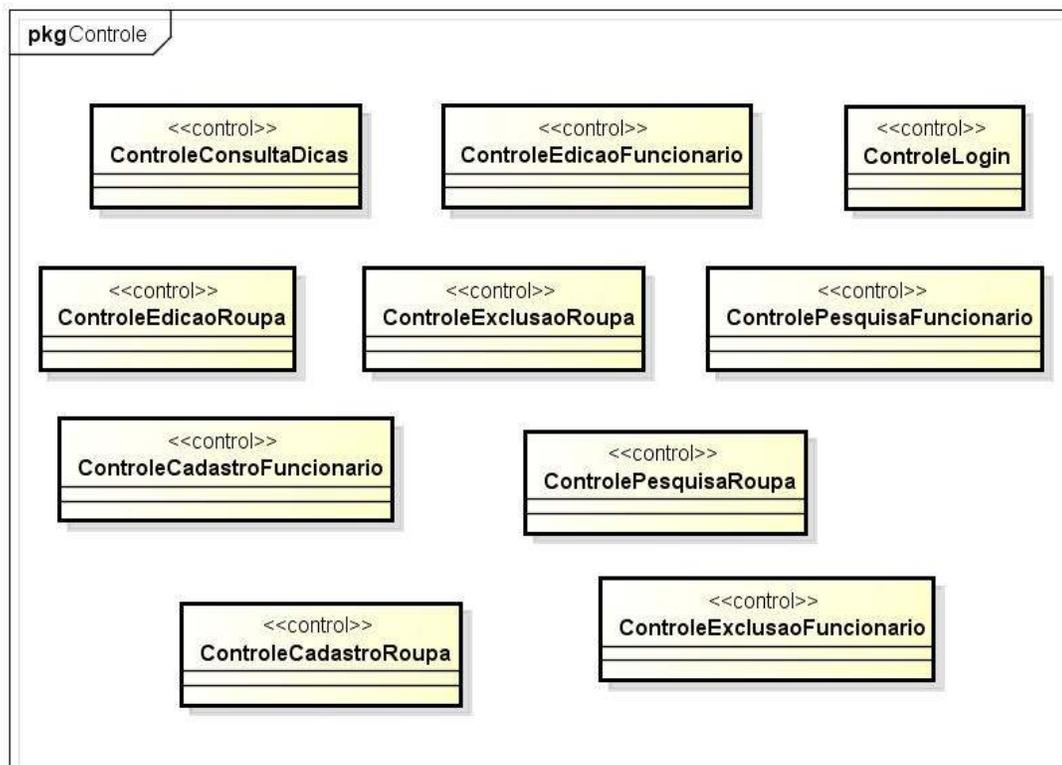
As classes de Fronteiras constituem classes que se comunicam diretamente com o usuário ou com algum subsistema existente no projeto. Elas são identificadas pelo estereótipo no diagrama UML como `<<boudary>>`.



powered by Astah

Figura 3 - Diagrama de Classe de Análise de Fronteira

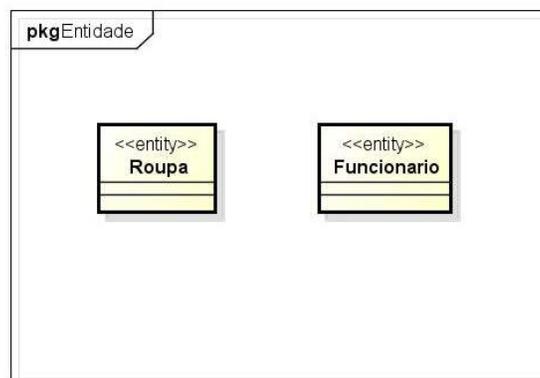
As classes de Controle são responsáveis pelo controle do sistema e pela comunicação entre as classes de fronteiras com as outras classes. Seu estereótipo no diagrama UML é definido como `<<control>>`.



powered by Astah

Figura 4 - Diagrama de Classes de Análise de Controle

As classes de Entidade representam o domínio do sistema e possuem informações importantes a respeito dela. Geralmente são fáceis de se identificar pois constituem em sua maioria os substantivos dos Casos de Uso. São representadas em UML por <<entity>>



powered by Astah

Figura 5 - Diagrama de Classes de Análise de Entidade

As classes destacadas como `<<entity collection>>` são denominadas classes de persistência. Esse tipo de classe é auxiliar das classes de análise e é responsável pela persistência dos dados.

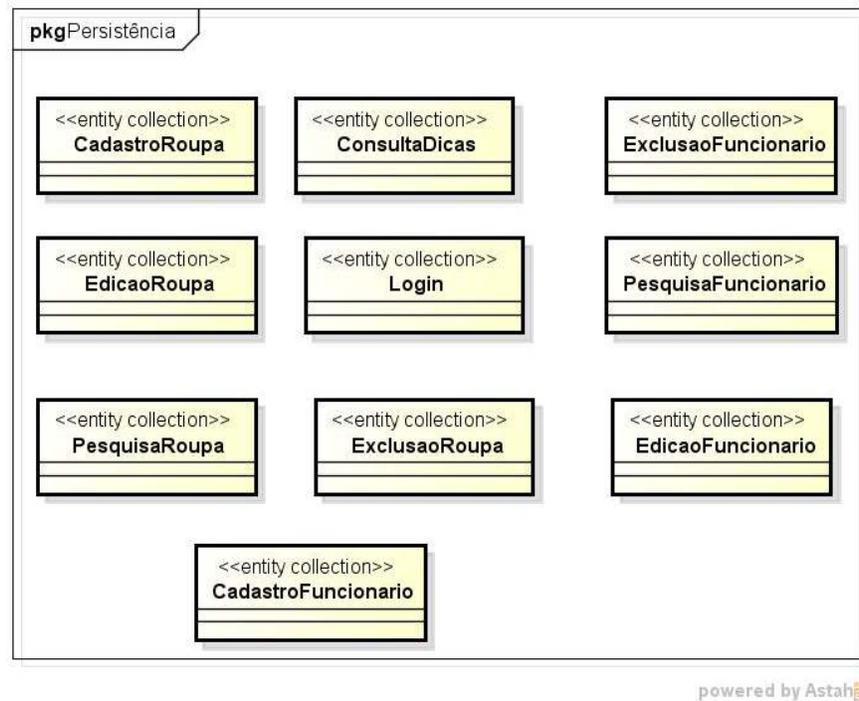
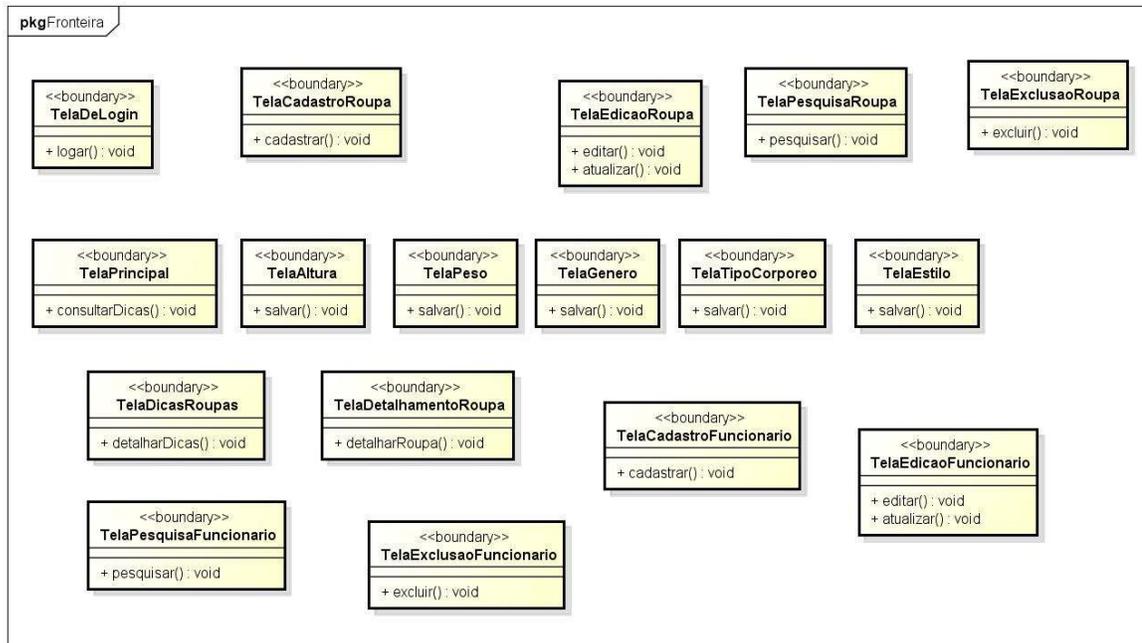


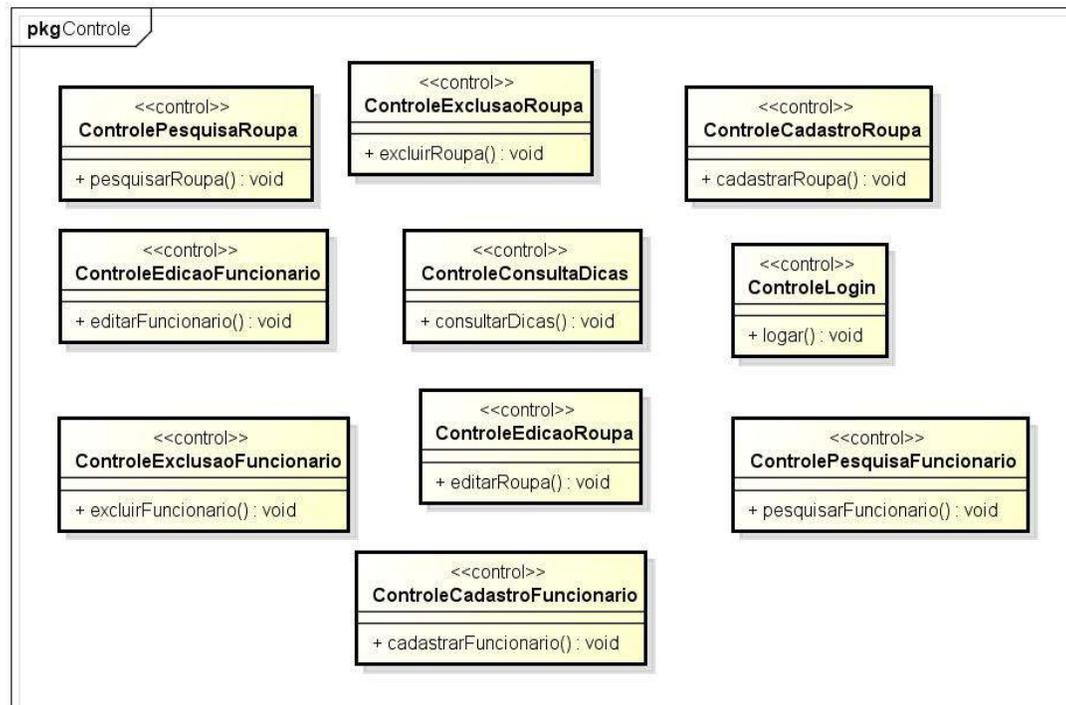
Figura 6 - Diagrama de Classes de Análise Auxiliar de Persistência

Depois de identificadas as classes de análise, pode-se passar para o próximo passo que é a identificação de operações e atributos que estas classes possuem. Nos diagramas a seguir foram identificadas as operações das classes de fronteira, controle e persistência. Nas classes de entidade foram identificados atributos que serão essenciais no sistema.



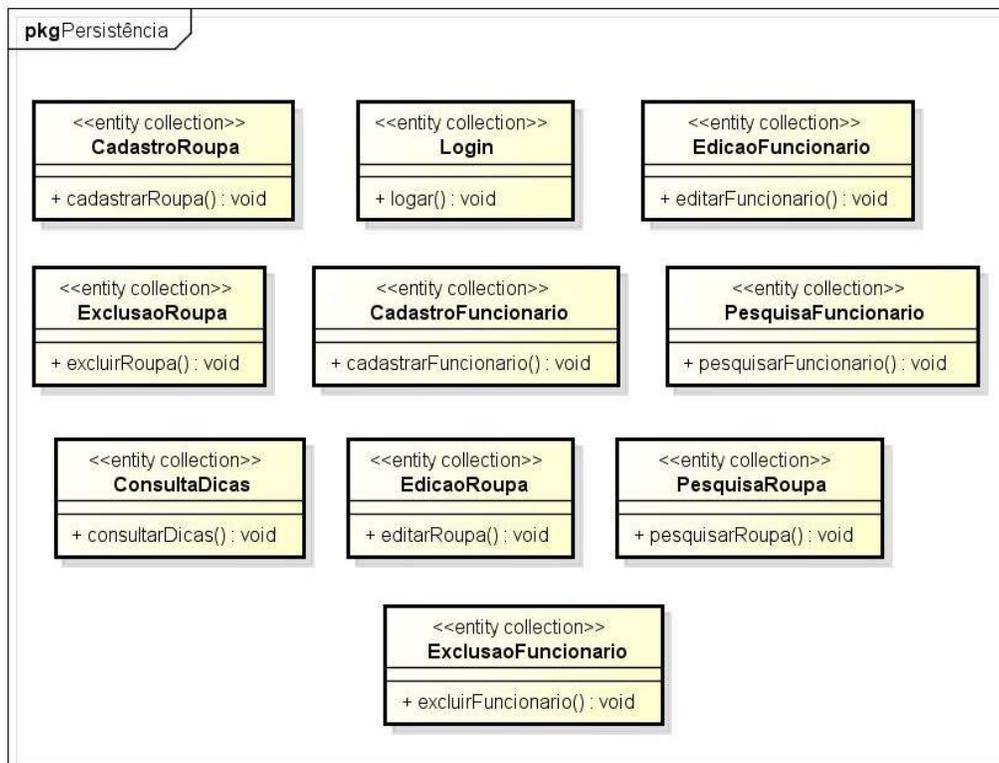
powered by Astah

Figura 7 - Diagrama de Classes de Análise de Fronteira com as operações



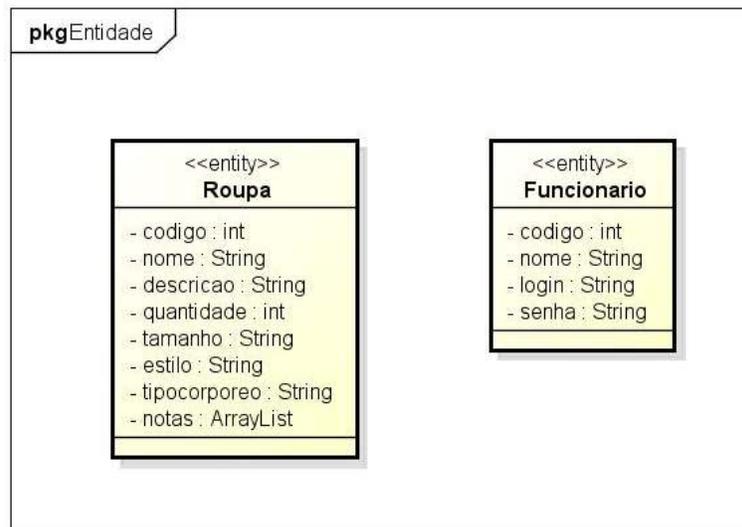
powered by Astah

Figura 8 - Diagrama de Classes de Análise de Controle com as operações



powered by Astah

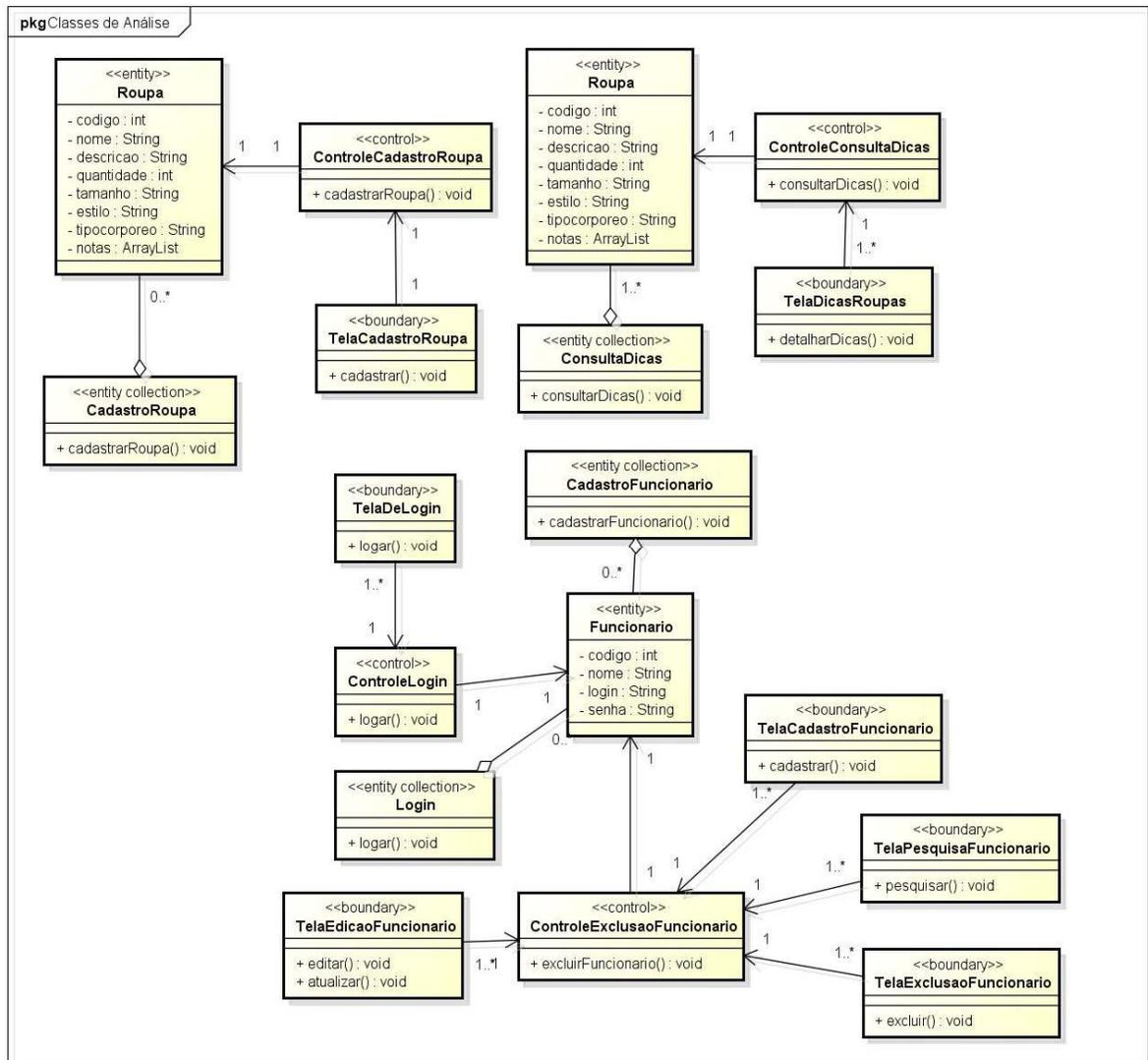
Figura 9 - Diagrama de Classes de Análise Auxiliar de Persistência com as operações



powered by Astah

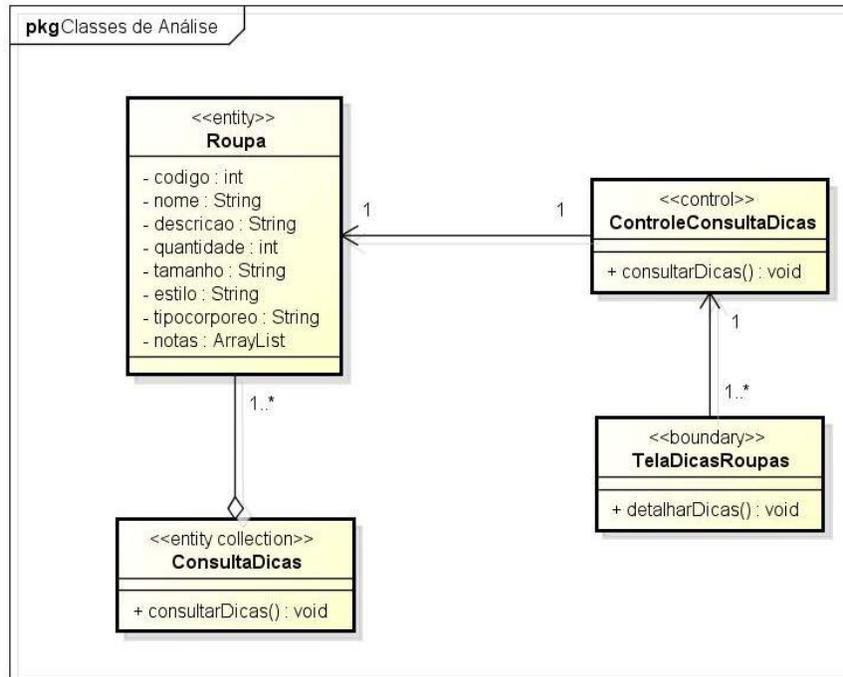
Figura 10 - Diagrama de Classes de Análise de Entidade com os atributos

Só após a identificação das operações e atributos, fica mais fácil relacionar as classes entre si e identificar seu comportamento e fluxo. Este constitui o próximo passo para a modelagem das classes de análise. Foi adicionado o relacionamento entre as classes e a cardinalidade delas.



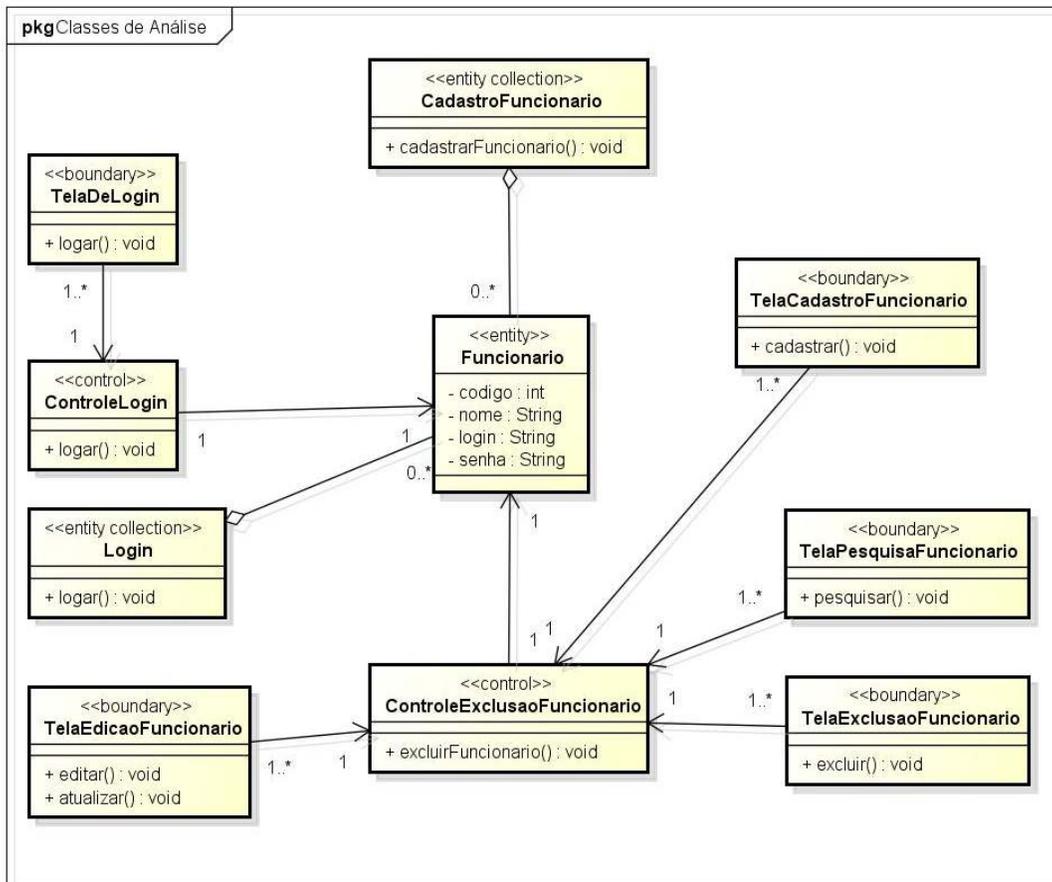
powered by Astah

Figura 11 - Diagrama de Classes de Análise com relacionamentos



powered by Astah

Figura 12 - Diagrama de Classes de Análise Parte 1



powered by Astah

Figura 13 - Diagrama de Classes de Análise Parte 2

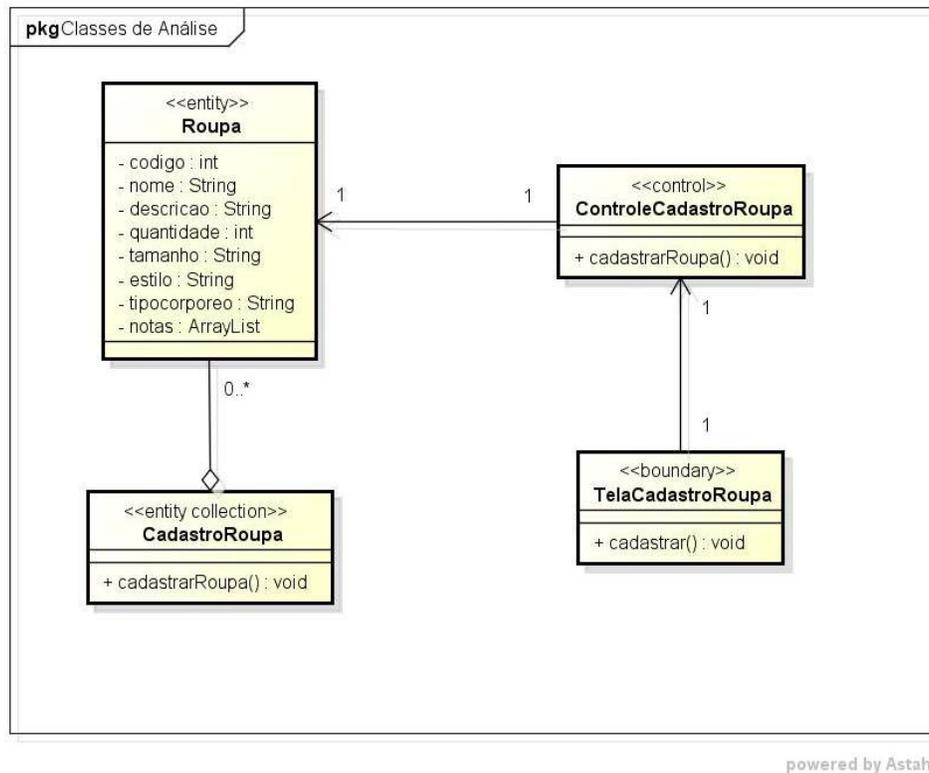
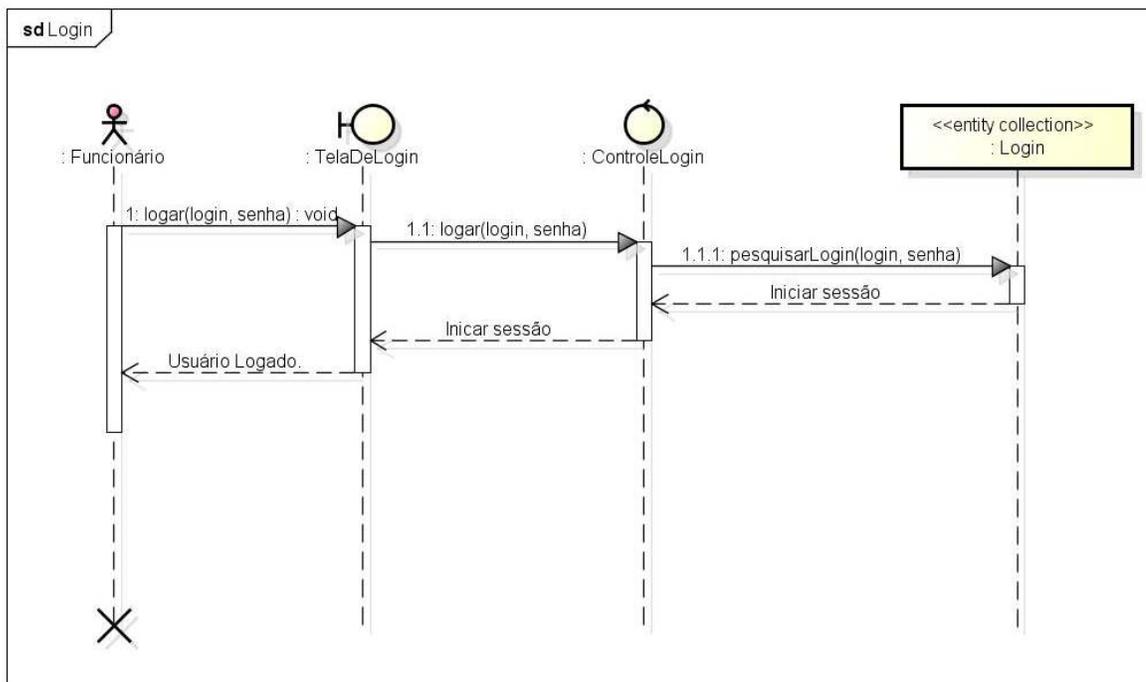


Figura 14 - Diagrama de Classes de Análise Parte 3

Diagrama de Sequência

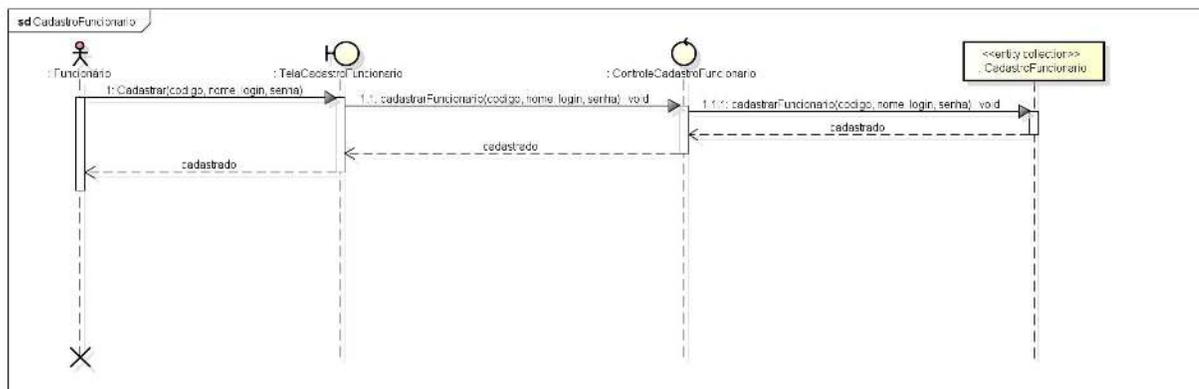
Diagrama de Sequência tem como finalidade identificar o fluxo das operações dos requisitos do sistema. Assim pode-se entender o funcionamento e a resposta dele a uma determinada solicitação do usuário. Geralmente torna-se necessário modelar um diagrama de sequência para cada requisito. Abaixo temos um diagrama de sequência para cada fluxo principal dos requisitos funcionais citados anteriormente neste documento.

Em cada diagrama de sequência a classe de Fronteira se comunica diretamente com a classe de Controle por meio de métodos identificados no último passo das Classes de Análise. A classe de fronteira se comunica por sua vez com as classes auxiliares de Persistência. Após a solicitação do usuário for efetuada a classe de Persistência retorna a classe de Controle uma mensagem ou o resultado (nos casos onde a solicitação seja relacionada a pesquisa). Esta por sua vez retorna o resultado a classe de Fronteira, para que o usuário receba a resposta da sua requisição.



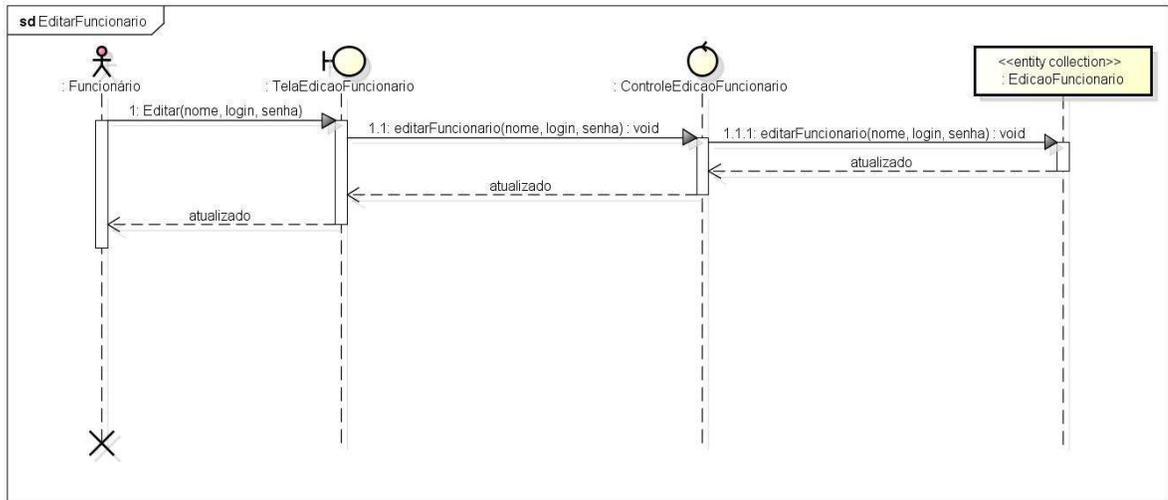
powered by Astah

Figura 15 - Diagrama de Sequência Requisito Logar no Sistema



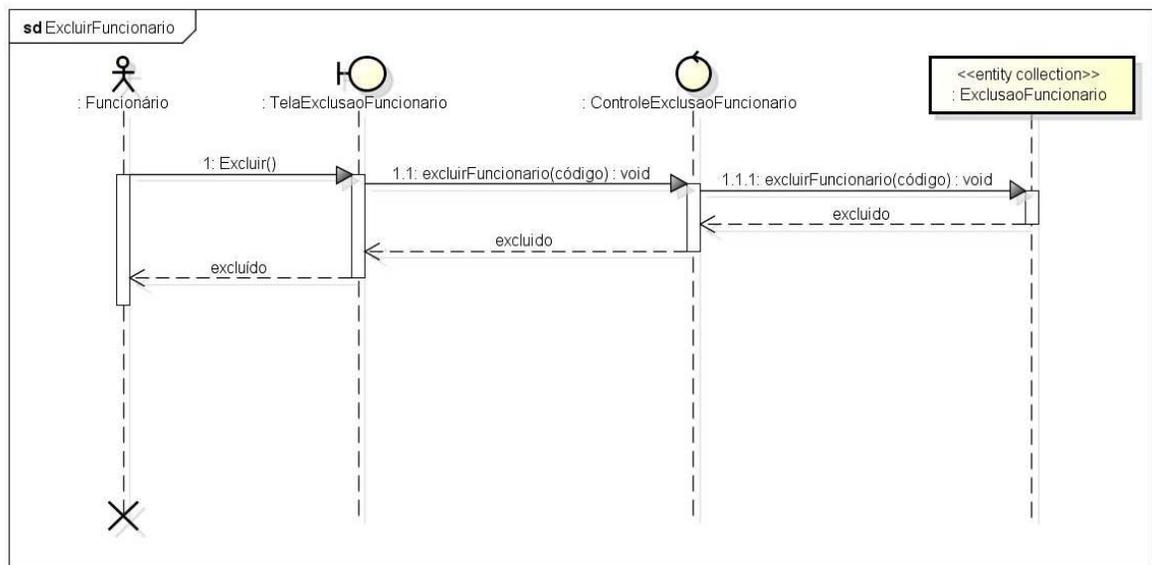
powered by Astah

Figura 16 - Diagrama de Sequência Requisito Cadastrar Funcionário



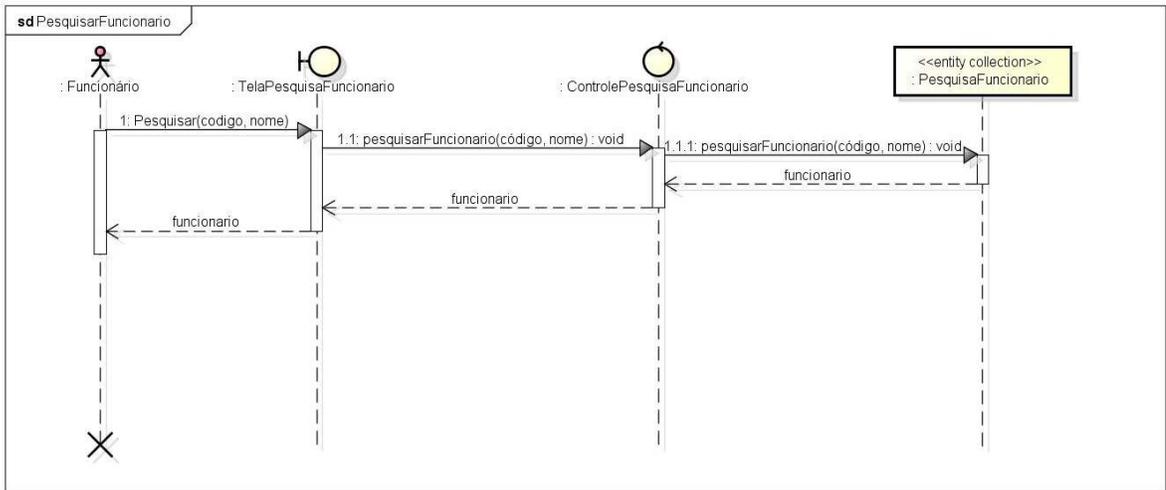
powered by Astah

Figura 17 - Diagrama de Sequência Requisito Editar Funcionário



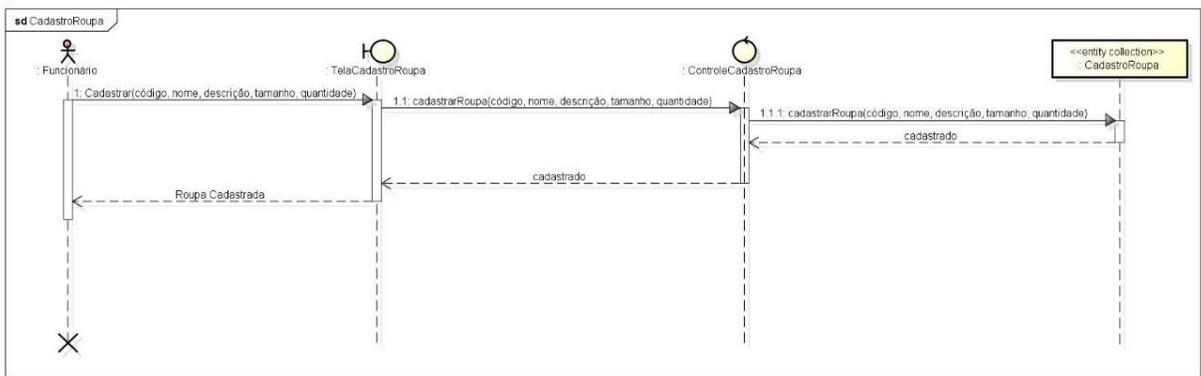
powered by Astah

Figura 18 - Diagrama de Sequência Requisito Excluir Funcionário



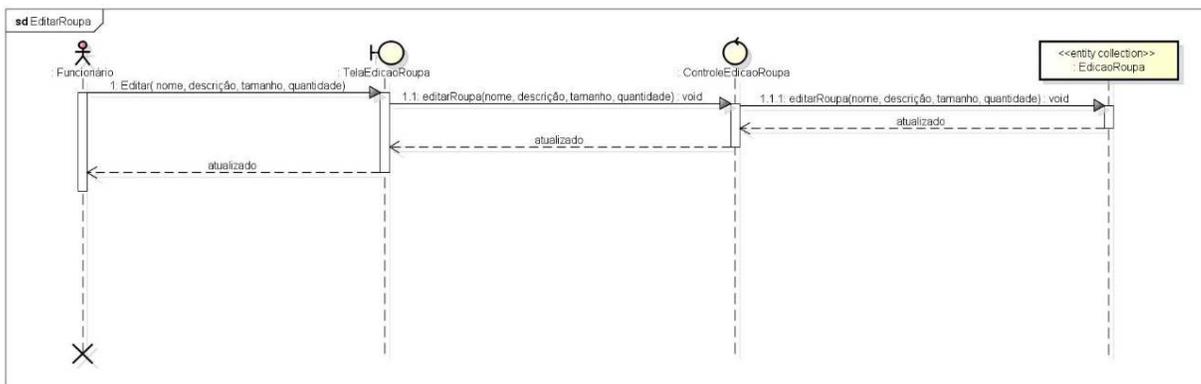
powered by Astah

Figura 19 - Diagrama de Sequência Requisito Pesquisar Funcionário



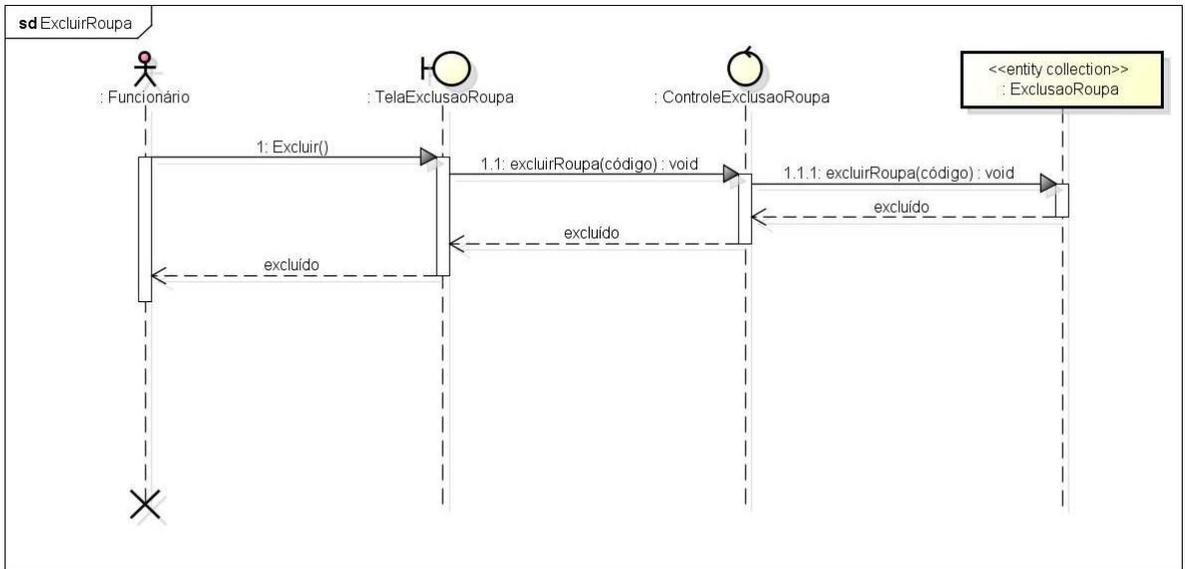
powered by Astah

Figura 20 - Diagrama de Sequência Requisito Cadastrar Roupa



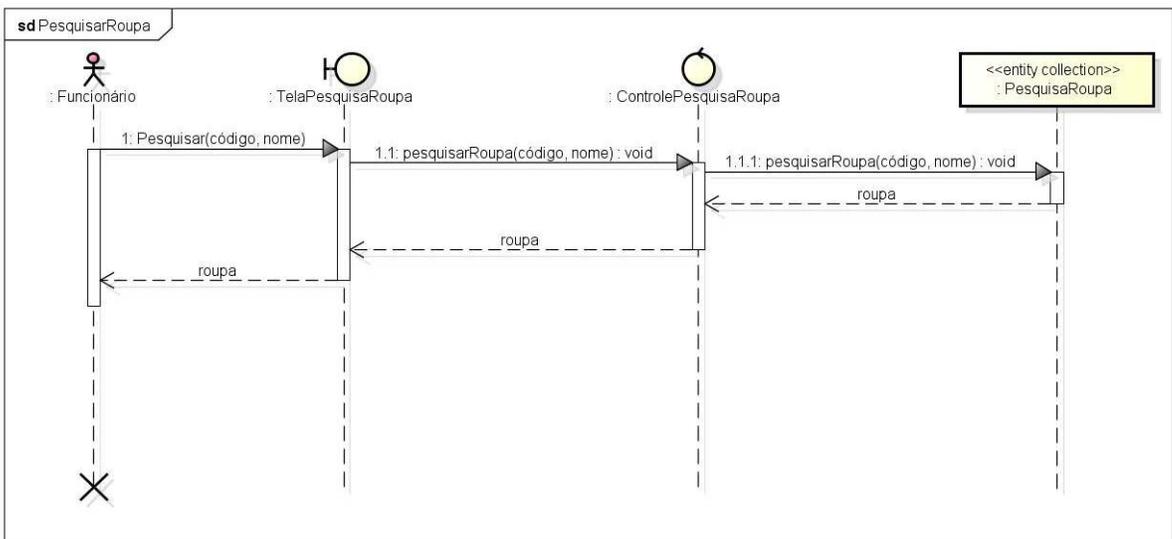
powered by Astah

Figura 21 - Diagrama de Sequência Requisito Editar Roupa



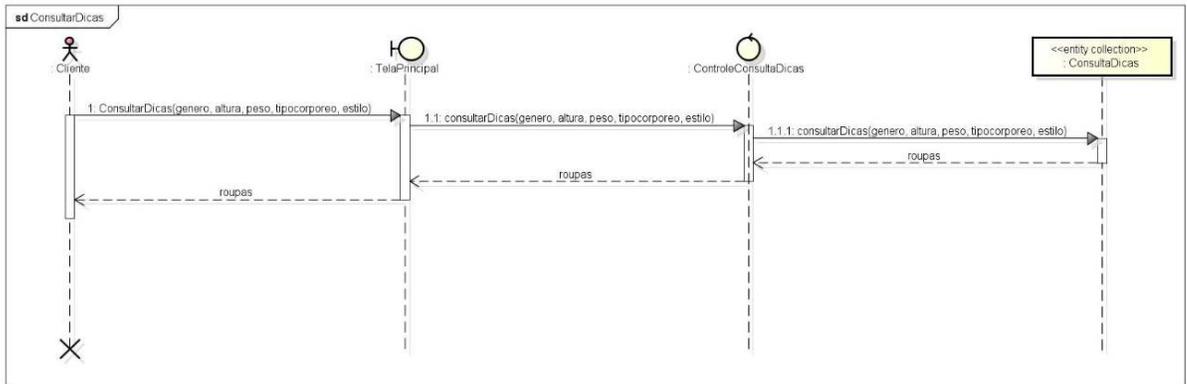
powered by Astah

Figura 22 - Diagrama de Sequência Requisito Excluir Roupa



powered by Astah

Figura 23 - Diagrama de Sequência Requisito Pesquisar Roupa

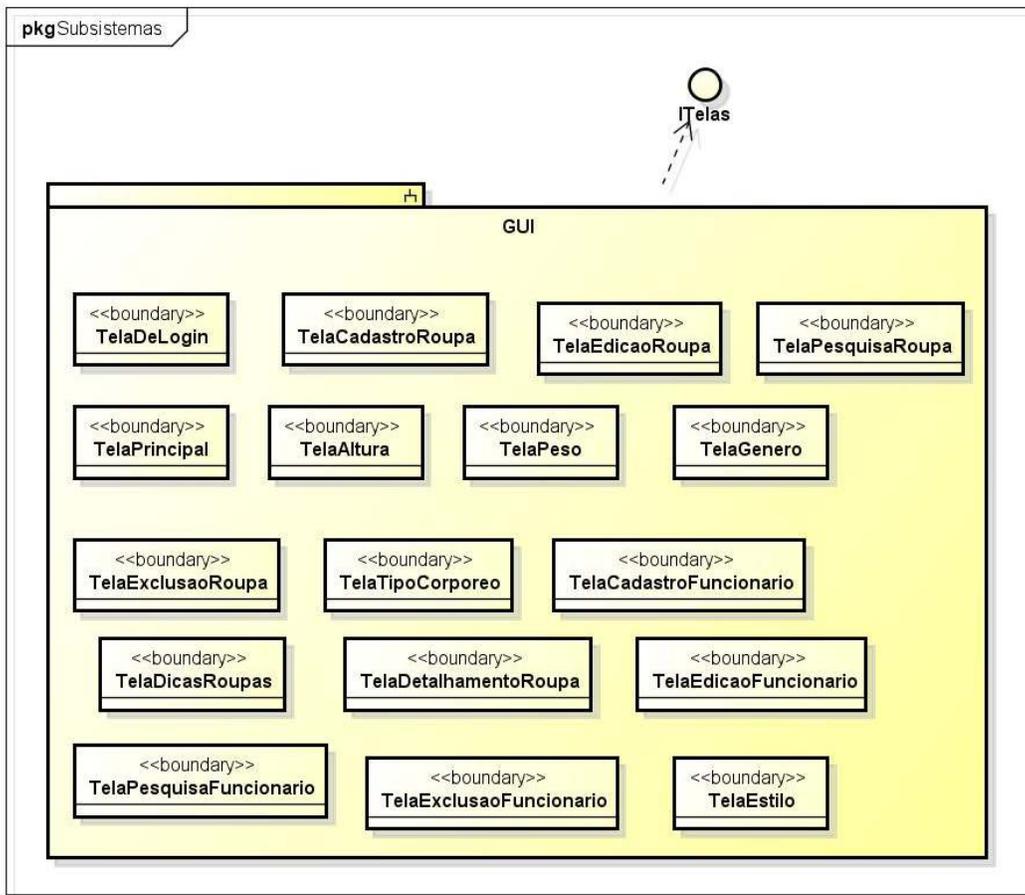


powered by Astah

Figura 24 - Diagrama de Sequência Requisito Consultar Dicas

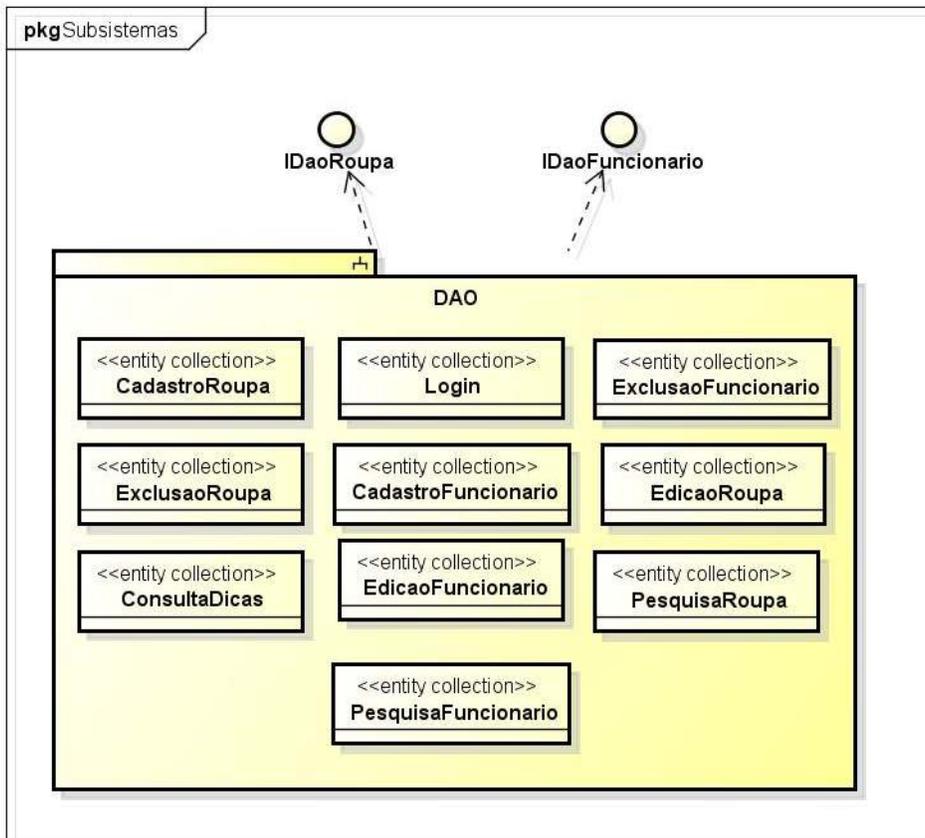
Subsistemas

O subsistema permite que seja planejado, desenvolvido e testado de maneira independente do restante do sistema. Com isso ele se torna um componente do projeto, facilitado sua implementação. No projeto PVS foi identificado dois subsistemas: GUI, que constitui a interface gráfica, onde esta se comunica diretamente com o usuário e o DAO, que agrega a parte de persistência dos dados onde neste projeto os dados serão armazenados em um Sistema de Banco de Dados. Os dois subsistemas possuem interfaces relacionadas a eles. Essas interfaces têm como função realizar a comunicação entre o subsistema e as outras classes do sistema.



powered by Astah

Figura 25 - Subsistema GUI



powered by Astah

Figura 26 - Subsistema DAO

Camadas

No diagrama de camadas foi separado em três camadas. A camada GUI corresponde as classes de telas do sistema, ou seja, o subsistema GUI. Este subsistema se comunica com a interface ITelas cuja função é fazer a comunicação entre a GUI e a próxima camada chamada de Negócio. Nesta camada estão as classes de controle do sistema. Estas classes se comunicam com as interfaces da Camada de Dados (IDaoFuncionario e IDaoRoupa). Na camada de Dados estão presentes todas as classes de persistência.

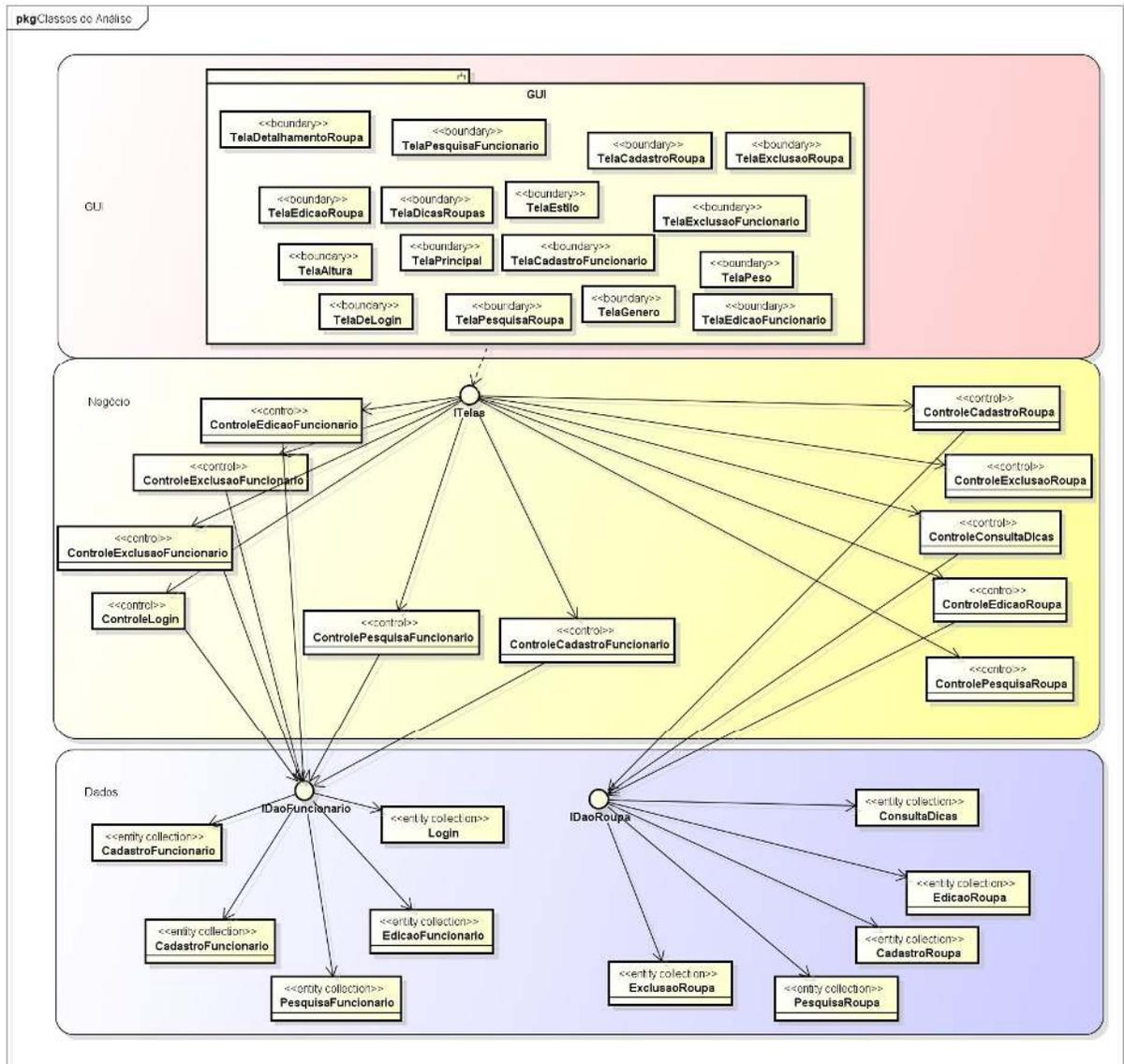
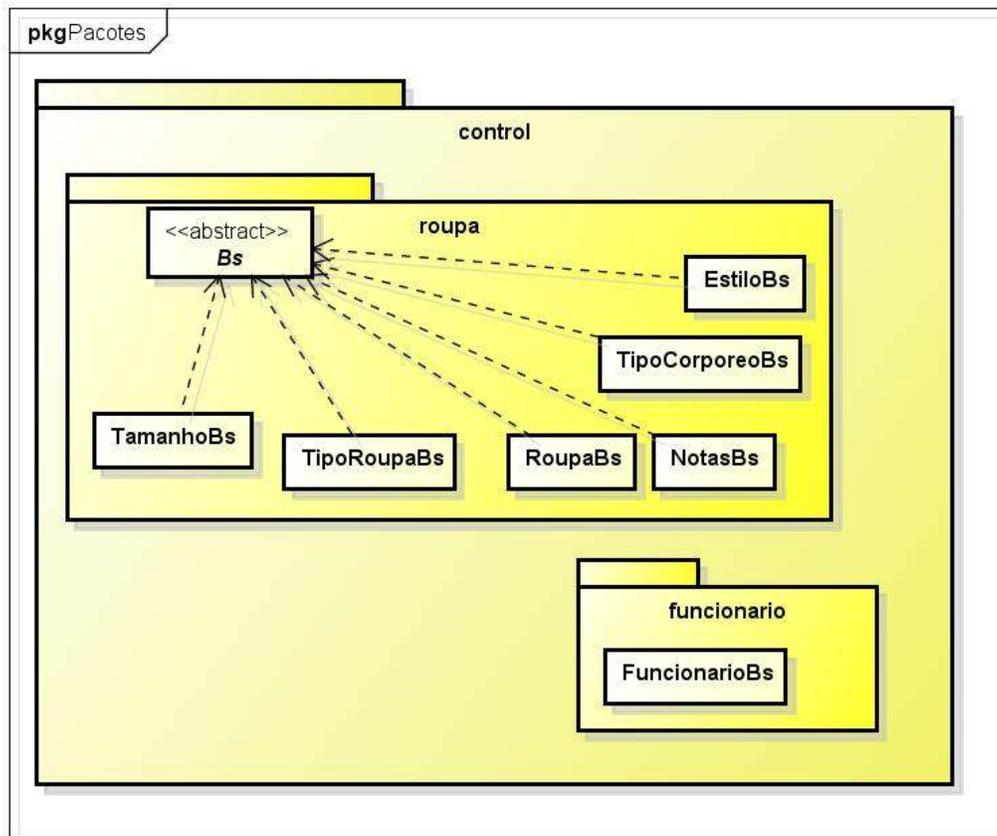


Figura 27 - Diagrama em Camadas

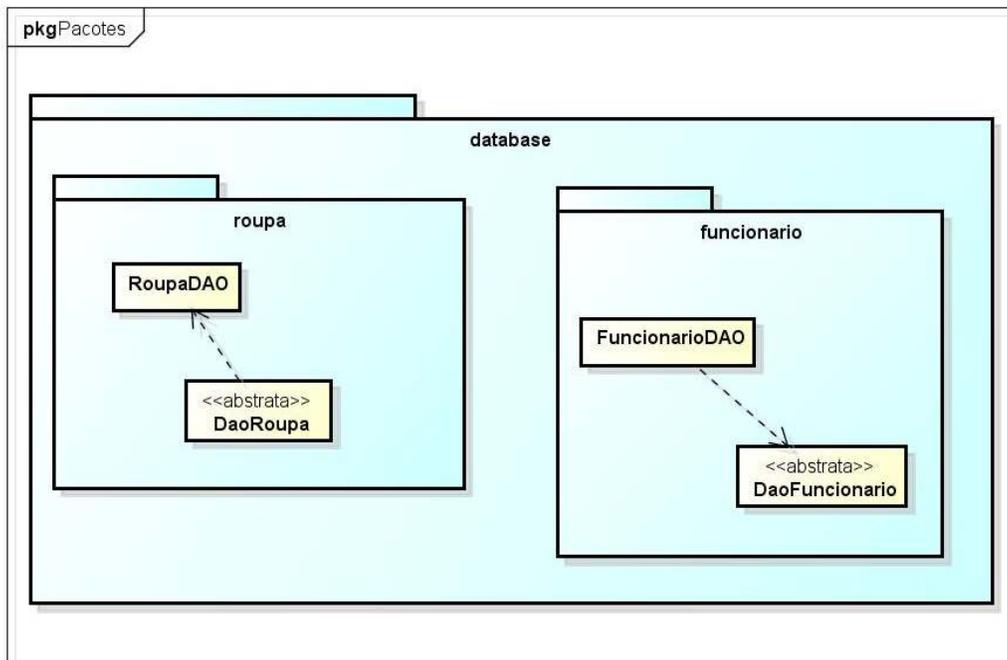
Diagrama de Pacotes

Os pacotes têm como objetivo organizar o código e as classes do projeto. O PVS foi dividido em quatro pacotes principais: *control*, *database*, *view* e *model*. O pacote *control* engloba as classes de controle e as classes responsáveis pelas regras de negócio do projeto. O pacote *database* engloba as classes que irão realizar a persistência dos dados. O pacote *view* por sua vez abrange todas as classes que se comunicam diretamente com o usuário. E por fim, o pacote *model* é constituído pelas classes de entidade/domínio do sistema. A divisão entre pacotes *roupa* e *funcionario* foi realizada para facilitar no desenvolvimento do projeto.



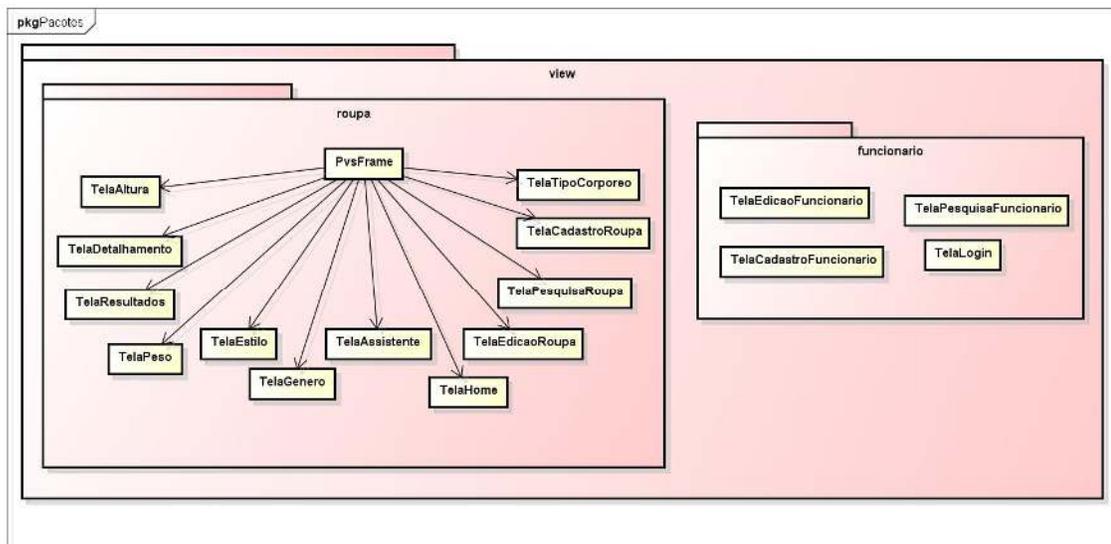
powered by Astah

Figura 28 - Diagrama de Pacotes control



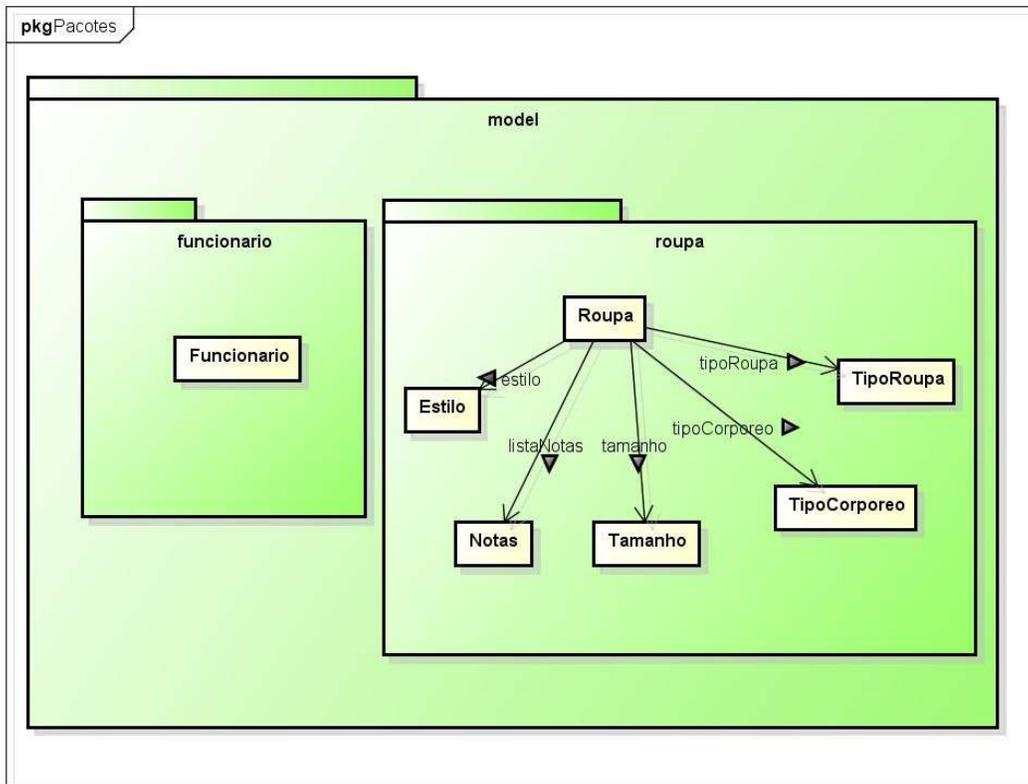
powered by Astah

Figura 29 - Diagrama de Pacotes database



powered by Astah

Figura 30 - Diagrama de Pacotes view



powered by Astah

Figura 31 - Diagrama de Pacotes model

MVC

“O MVC inicialmente foi desenvolvido no intuito de mapear o método tradicional de entrada, processamento, e saída que os diversos programas baseados em GUI utilizavam.”

Fonte: [Introdução ao Padrão MVC](http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308#ixzz38XrXPJEM) <<http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308#ixzz38XrXPJEM>>

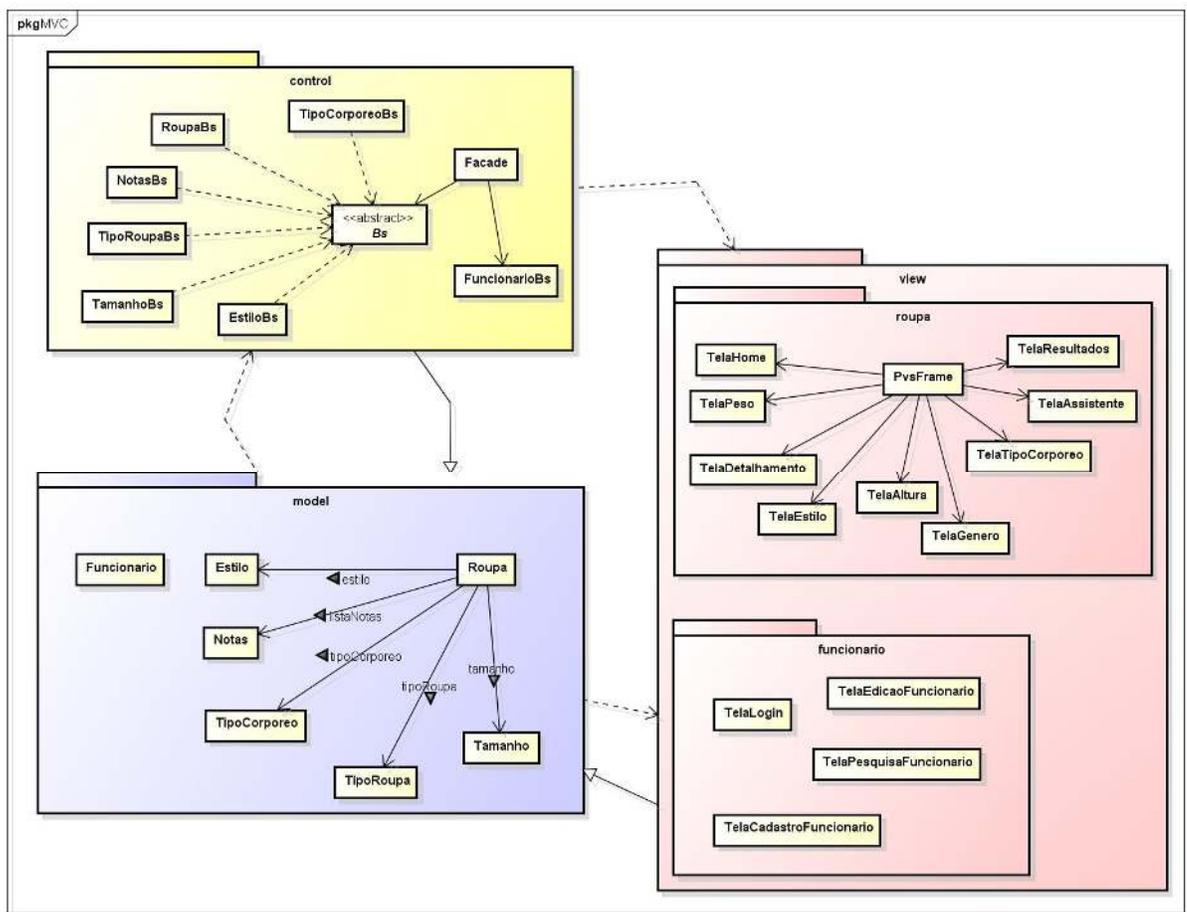


Figura 32 - Diagrama MVC

Padrões de Projeto

Os padrões de projeto são descrições de objetos que se comunicam e classes que são customizadas para resolver um problema genérico de design em um contexto específico

-Gamma, Erich, et al. "Padrões de projeto." *Soluções Reutilizáveis de Software orientado a objetos*(2000).

Os Padrões de Projeto auxiliam na resolução de problemas genéricos no desenvolvimento de software. No Projeto PVS foram utilizados os seguintes padrões:

Singleton

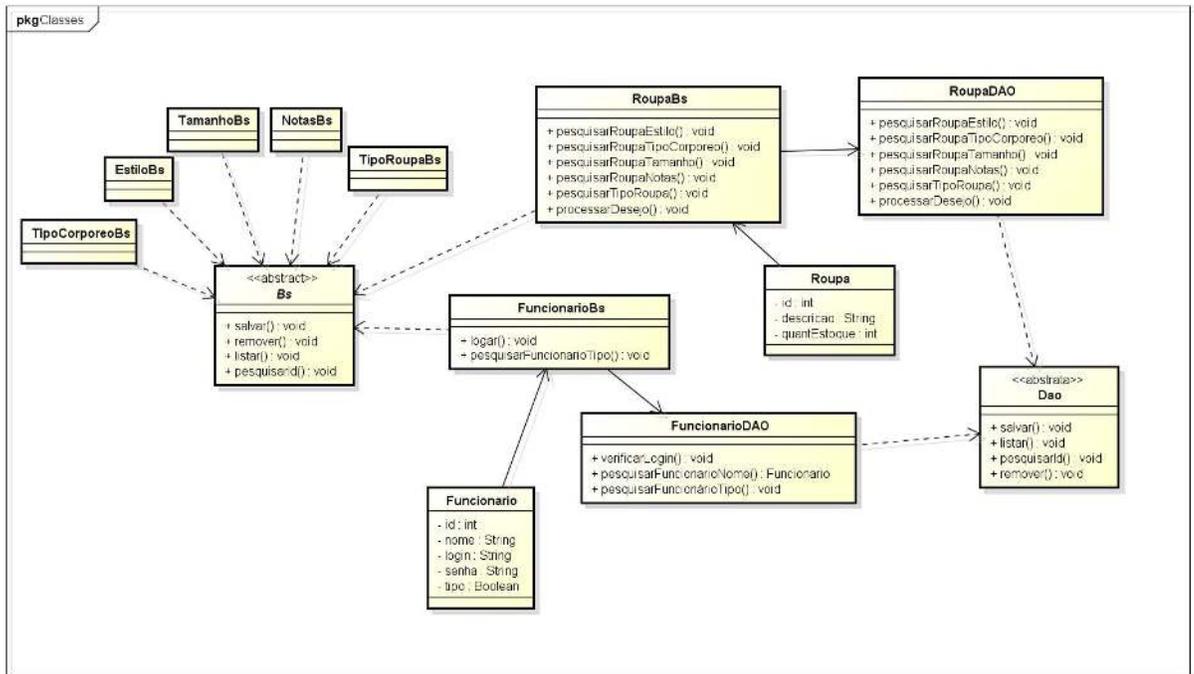
O padrão Singleton é do tipo criacional. Ele garante que a classe tenha apenas uma única instância, fornecendo um único meio de acesso para ela. No projeto esse padrão está modelado nas classes de negócio e dao. Com esse padrão pode-se manter o controle ao acesso dessas classes.

Façade

O padrão Façade é do tipo estrutural. Ele fornece uma única interface para um conjunto de classes em um subsistema. Ela é importante pois faz com que ocorra um desacoplamento entre as classes do sistema. No projeto ela é utilizada para as classes de telas (que interagem diretamente com o usuário, o subsistema GUI) e as classes de negócio do sistema.

Oserver

O padrão Observer é um padrão comportamental. Ele notifica vários outros objetos sobre eventos e/ou mudanças ocorridas em um outro objeto. Ou seja, quando um objeto que é observado muda de estado os dependentes automaticamente têm seu estado atualizado. No Projeto esse padrão é utilizado junto com o padrão Façade para observar as mudanças das classes de interface com o usuário.



powered by Astal

Figura 33 - Diagrama de Classes sem a implementação dos Padrões de Projeto

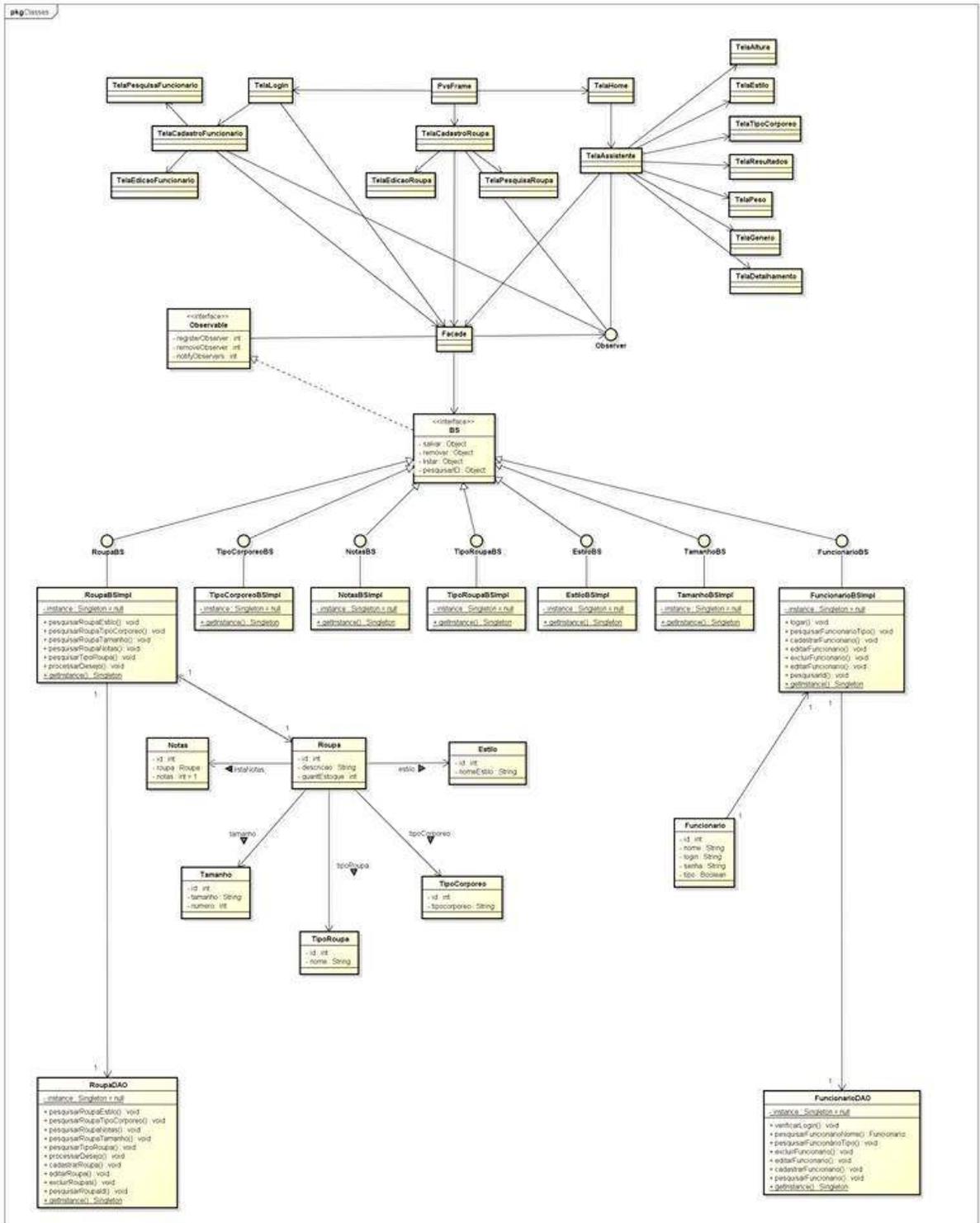


Figura 34 - Diagrama de Classes com a implementação dos Padrões de Projeto

Prototipação da Interface

Fluxo de Telas

A seguir é mostrado o fluxo das telas do Sistema. Na Tela Home o usuário pode selecionar a opção *Login* e acessar o sistema para realização de funções pertencentes ao funcionário. Em casa uma das opções específicas que o usuário pode realizar, é possível voltar a Página Inicial. Se o usuário for um cliente da loja, o sistema solicita um conjunto de informações e retorna ao usuário, na tela de resultados, as dicas de moda. Em qualquer momento o usuário pode voltar para a Página Inicial do Sistema.

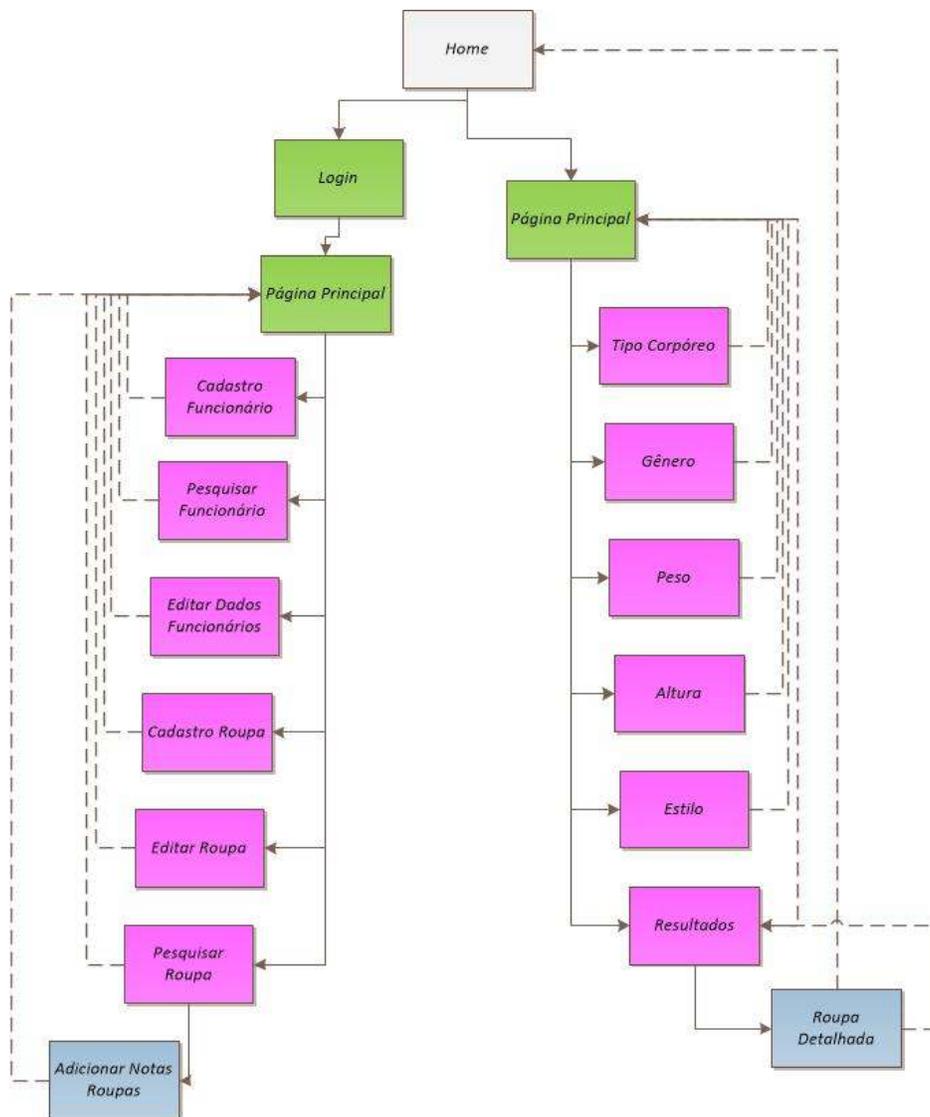


Figura 35 - Fluxo das Telas

Protótipos das Telas

Os protótipos de tela apresentados a seguir mostram as telas referentes ao valor de negócio do sistema que é as dicas de moda aos usuários.



Figura 36 – Protótipo de Telas (Tela 1)

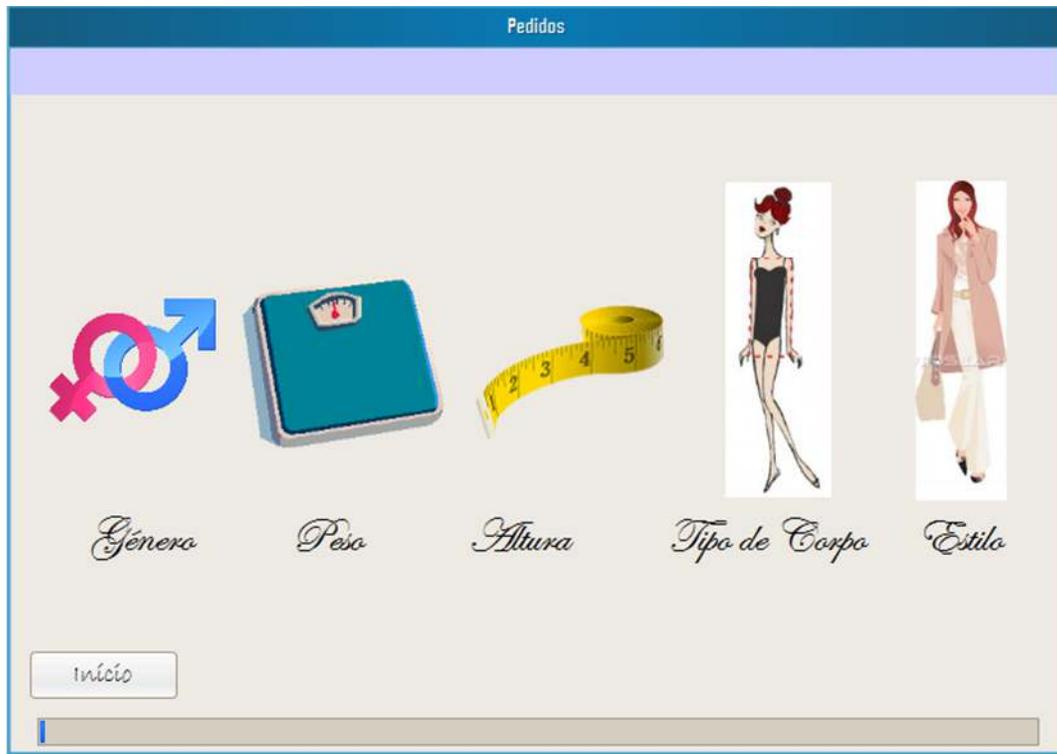


Figura 37 - Protótipo de Telas (Tela 2)



Figura 38 - Protótipo de Telas (Tela 3)

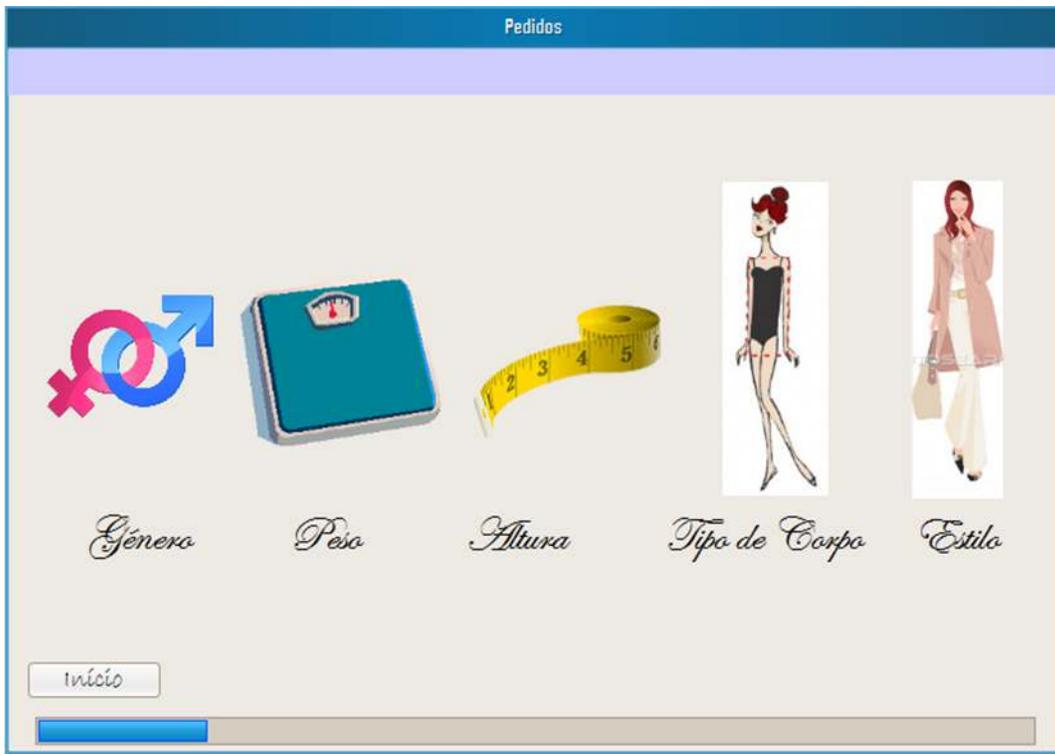


Figura 39 - Protótipo de Telas (Tela 4)



Figura 40 - Protótipo de Telas (Tela 5)

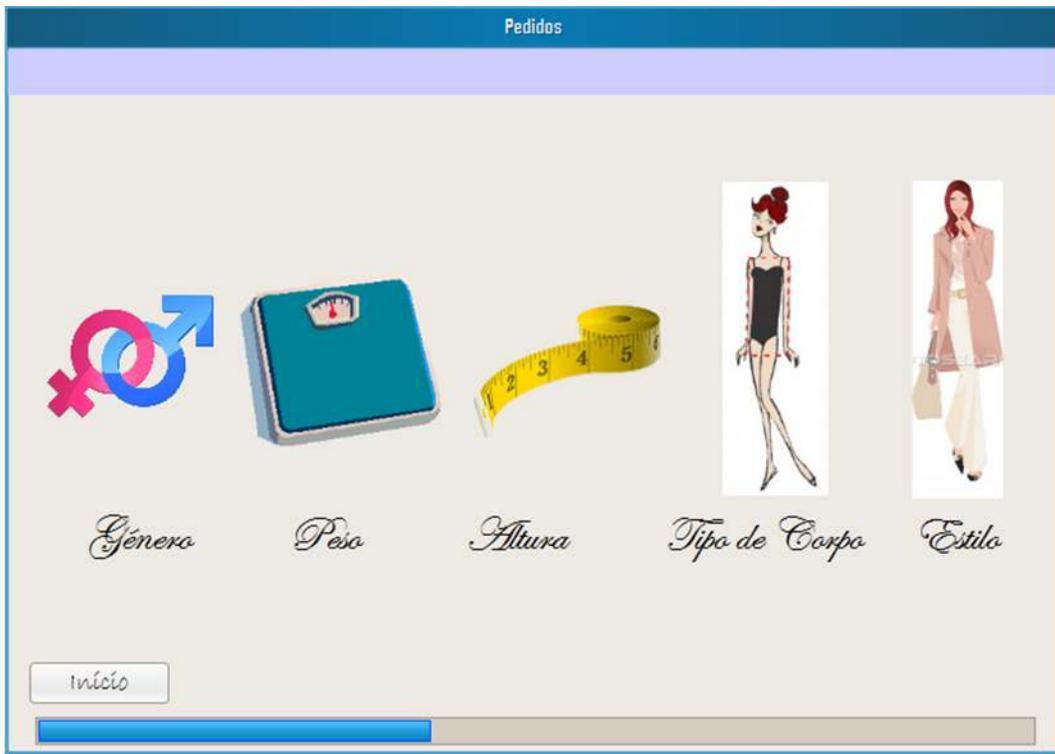


Figura 41 - Protótipo de Telas (Tela 6)



Figura 42 - Protótipo de Telas (Tela 7)

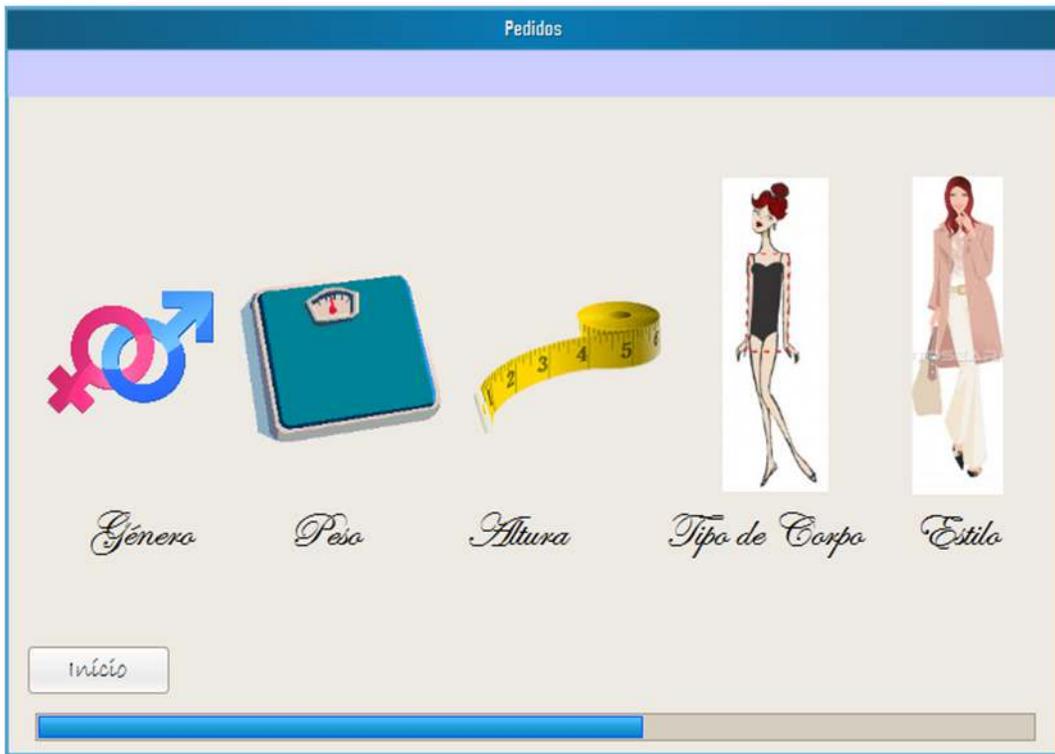


Figura 43 - Protótipo de Telas (Tela 8)

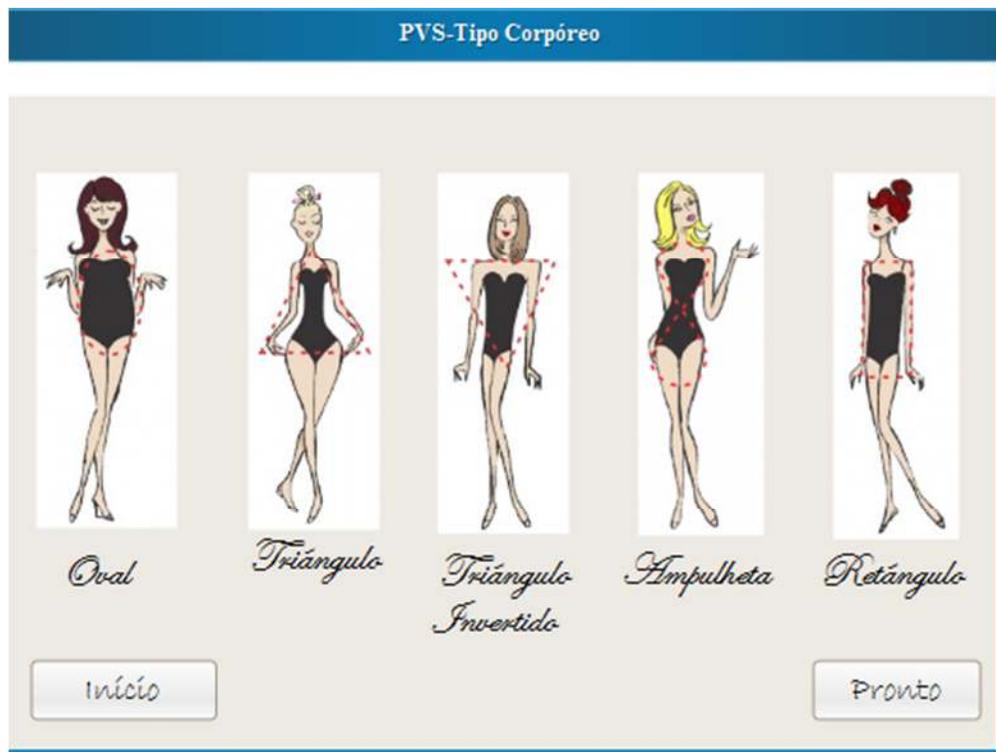


Figura 44 - Protótipo de Telas (Tela 9)



Figura 45 - Protótipo de Telas (Tela 10)



Figura 46 - Protótipo de Telas (Tela 11)

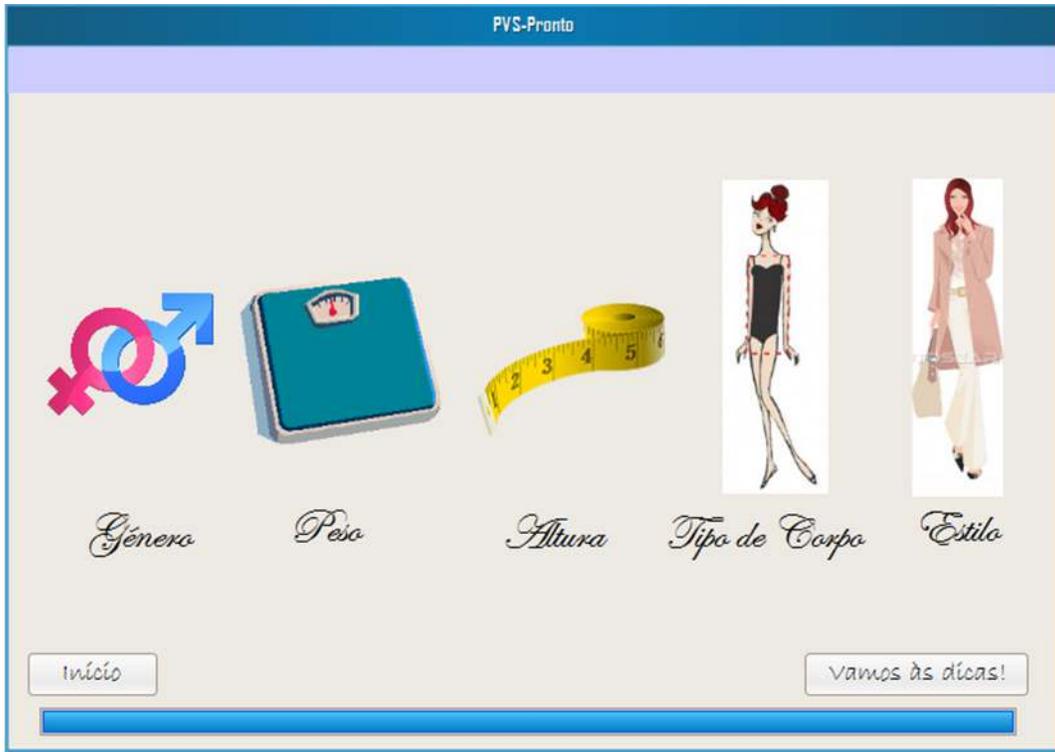


Figura 47 - Protótipo de Telas (Tela 12)

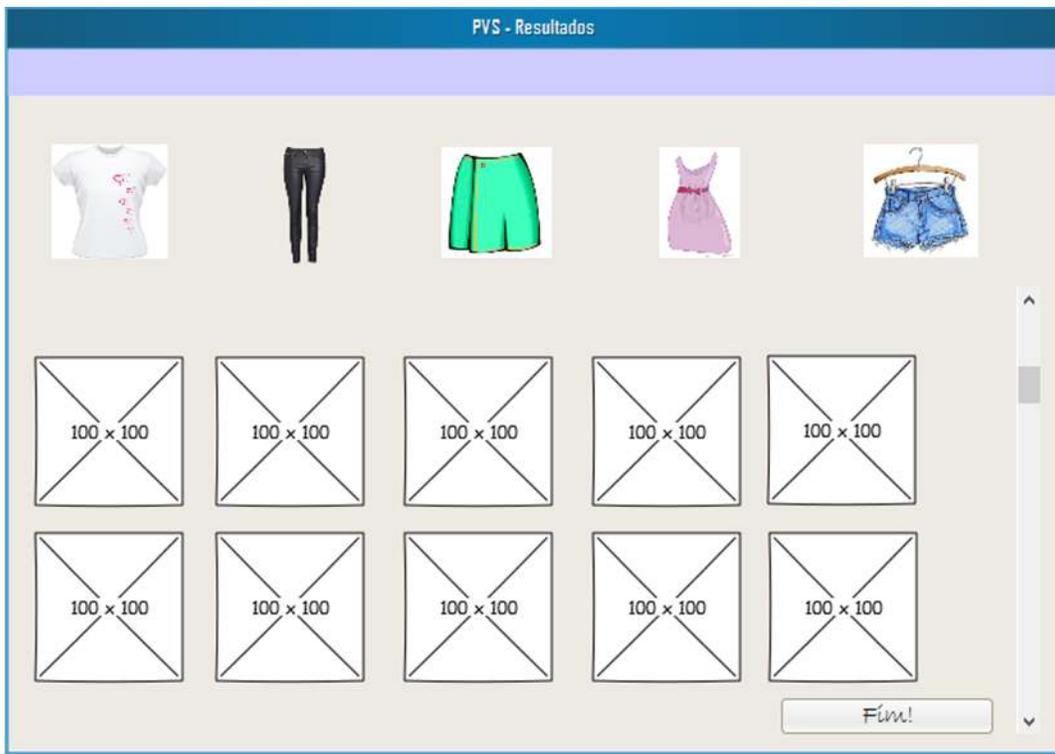


Figura 48 - Protótipo de Telas (Tela 13)



Figura 49 - Protótipo de Telas (Tela 14)

SOA

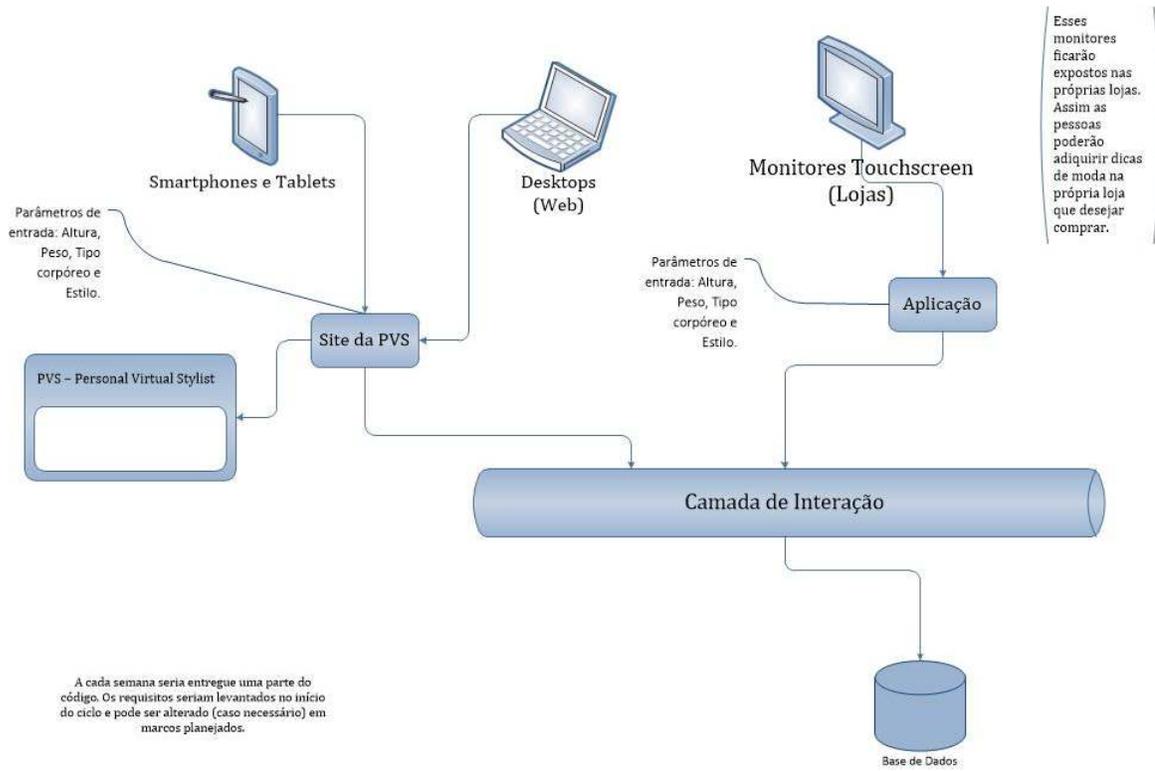


Figura 50 - Diagrama SOA Sistema PVS completo (ideia)

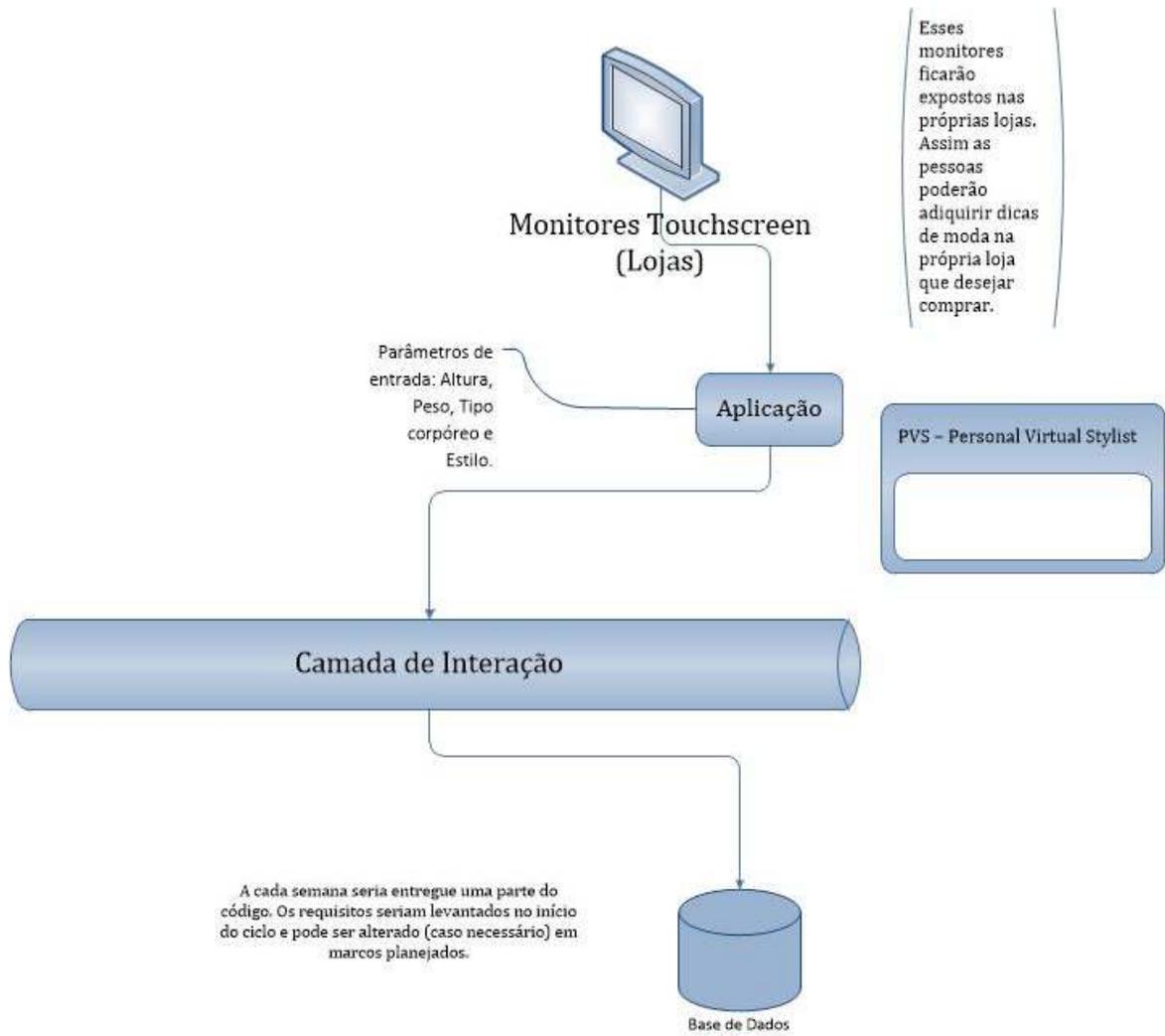


Figura 51 - Diagrama SOA Sistema PVS a ser implementado

Banco de Dados

