



**Universidade Federal dos Vales Jequitinhonha e
Mucuri - UFVJM**
Sistemas de Informação



Prof. Eduardo Pelli

Programação Orientada a Objetos

Trabalho Prático 1

Álvaro Patrocínio
João Vitor Pinheiro

Diamantina, 31 de agosto de 2023

Contents

1	Introdução	3
1.1	O Problema	3
1.2	Proposta	4
2	Casos de uso	5
2.1	Descrição	5
2.1.1	Atores	5
2.1.2	Casos de Uso	5
2.1.3	Relacionamentos	5
2.2	Diagrama de Caso de Uso	7
2.3	Explicação	9
3	Fluxo de Eventos	10
3.1	Caso de Uso 1: Venda Simples	10
3.2	Caso de Uso 2: Verificar produto no estoque	10
3.3	Caso de Uso 3: Verificar validade de um produto	11
3.4	Caso de Uso 4: Verificar rendimento diário de um caixa	11
3.5	Caso de Uso 5: Gerar balanço mensal	11
3.6	Caso de Uso 6: Cancelar Venda	12
3.7	Caso de Uso 7: Incluir um administrador	12
3.8	Caso de Uso 8: Editar um administrador	12
3.9	Caso de Uso 9: Remover um administrador	13
3.10	Caso de Uso 10: Alterar a senha de um administrador	13
3.11	Caso de Uso 11: Incluir cliente e definir o id do cliente.	13
3.12	Caso de Uso 12: Editar cliente.	13
3.13	Caso de Uso 13: Remover cliente	14
4	Sequência	15
4.1	Descrição	15
4.1.1	Atores	15
4.1.2	Objetos	15
4.1.3	Linhas de Vida	15
4.1.4	Mensagens	16
4.1.5	Ocorrência de Execução	16
4.2	Diagramas de Sequência	17
4.2.1	Diagrama 1: Venda Simples	17
4.2.2	Diagrama 2: Verificar Produto em Estoque	19
4.2.3	Diagrama 3: Verificar Validade dos Produtos	20
4.3	Diagrama 4: Verificar Rendimento de cada Caixa	21
4.4	Diagrama 5: Gerar Balanço Mensal	22

5	Classes	23
5.1	Descrição	23
5.1.1	Classes	23
5.1.2	Associações	23
5.1.3	Herança	24
5.1.4	Composição	24
5.2	Diagrama de Classe	25
6	Estado	27
6.1	Descrição	27
6.1.1	Estado	27
6.1.2	Transições	27
6.1.3	Eventos	27
6.2	Diagrama de Estado	28
7	Conclusão	30
8	Referências Bibliográficas	31
8.1	Referências adicionais	31

1 Introdução

O conceito da programação orientada a objetos, de maneira simplificada, representa uma abordagem fundamental para a resolução de problemas e tem desempenhado um papel fundamental no desenvolvimento de software contemporâneo. Uma das pedras angulares desse paradigma é a "analogia biológica" proposta por Alan Kay. Nessa analogia, ele estabeleceu uma conexão entre um sistema de software e o funcionamento de um organismo vivo, onde cada componente atua como uma entidade autônoma que se comunica com outras através da troca de mensagens. Alan estabeleceu os seguintes princípios para a orientação a objetos:

- Qualquer coisa é um objeto;
- Objetos realizam tarefas por meio de requisição de serviços a outros objetos;
- Cada objeto pertence a uma determinada classe, e uma classe agrupa objetos similares;
- A classe é um repositório para comportamento associado ao objeto;
- Classes são organizadas em hierarquias.

Uma das partes fundamentais no paradigma da programação orientada a objetos é a modelagem. A modelagem é empregada quando a compreensão completa de um sistema se torna difícil devido à sua complexidade, o que é uma situação comum em sistemas orientados a objetos. A criação de um modelo permite simplificar a realidade e melhorar a compreensão do sistema em desenvolvimento.

Assim, ao utilizar a programação orientada a objetos como técnica de modelagem, a diferença semântica entre a realidade que está sendo modelada e os modelos criados é consideravelmente reduzida.

Nesse contexto, neste trabalho em questão, será realizada uma modelagem para um sistema de gerenciamento de um mercado, o SuperEXPRESS. A linguagem de modelagem unificada, UML, será utilizada para criar os diagramas necessários.

1.1 O Problema

Será desenvolvida uma análise orientada a objetos com o intuito de gerenciar um mercado, o SUPEREXPRESS, com estrutura de 5 caixas.

O sistema será utilizado por seu administrador e funcionários.

Os funcionários poderão utilizar o software para gerenciamento de estoque, balanços, vendas, fechamento de caixa e gerenciamento de despesas e relatórios.

O administrador terá acesso total ao sistema, já os funcionários terão acesso a realização de vendas, gerenciamento de vendas e gerenciamento de estoque.

1.2 Proposta

Para realizar a análise orientada a objetos do sistema especificado, serão feitos: um diagrama de caso de uso do problema, o fluxo de eventos e os diagramas de sequência de alguns casos de uso presentes no diagrama de caso de uso, um diagrama de classe para o sistema e um diagrama de estado para um determinado estado.

2 Casos de uso

2.1 Descrição

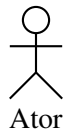
O diagrama de casos de uso é uma das representações definidas pela UML (Unified Modeling Language) e tem como objetivo facilitar a comunicação entre os analistas e o cliente durante o processo de desenvolvimento de sistemas. Este diagrama permite ao cliente visualizar as principais funcionalidades do sistema, bem como as interações entre seus componentes.

O diagrama de casos de uso descreve cenários que demonstram as funcionalidades do sistema sob a perspectiva do usuário, proporcionando uma visão de alto nível do sistema a partir de uma perspectiva externa. Isso o torna uma ferramenta crucial para a visualização, especificação e documentação do comportamento do sistema.

A criação desse modelo envolve a definição de vários elementos, incluindo casos de uso, atores e relacionamentos.

2.1.1 Atores

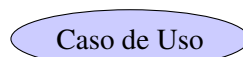
Os atores representam os diferentes papéis ou entidades externas que interagem com o sistema. Os atores podem ser pessoas reais, outros sistemas, dispositivos ou até mesmo processos externos. Eles desempenham um papel importante na definição dos cenários de uso do sistema. Na UML, um ator é representado por um boneco e um rótulo com o nome do ator, conforme apresentado a seguir:



2.1.2 Casos de Uso

Os casos de uso são representações das funcionalidades ou serviços oferecidos pelo sistema. Cada caso de uso descreve uma interação específica entre o sistema e um ator (que pode ser um usuário real, outro sistema ou entidade externa). Eles ajudam a identificar como os usuários interagem com o sistema e quais tarefas podem realizar.

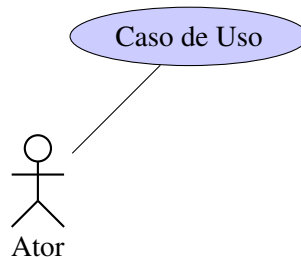
Na UML, um caso de uso é representado da seguinte forma:



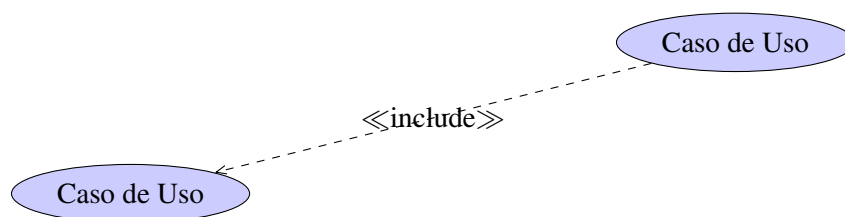
2.1.3 Relacionamentos

Relacionamentos são as conexões entre casos de uso e atores. Existem diferentes tipos de relacionamentos, como associação, inclusão e extensão, que descrevem como os casos de uso se relacionam entre si e com os atores. Esses relacionamentos ajudam a definir o fluxo de interações no sistema.

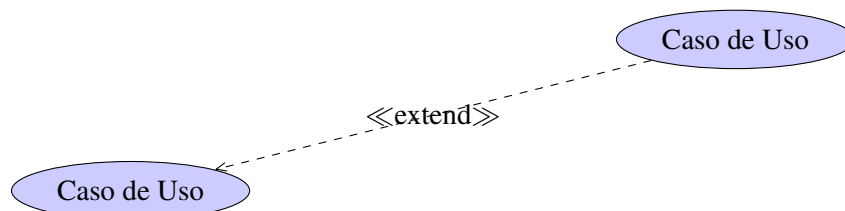
1. Comunicação: Dentro dos diagramas de casos de uso da UML, o relacionamento de comunicação é o mais comum e serve para indicar a ligação entre um ator e um caso de uso específico. Esse vínculo representa que o ator está conectado ao sistema por meio desse caso de uso, o que significa que o ator realiza ações e troca informações com o sistema por intermédio desse caso de uso em particular. É importante notar que um ator pode ter relações com vários casos de uso, o que reflete a variedade de maneiras pelas quais os atores podem interagir com o sistema.



2. O relacionamento de inclusão é aplicado exclusivamente entre casos de uso. Quando dois ou mais casos de uso compartilham uma sequência de interações comuns, essa sequência pode ser descrita em um caso de uso separado. Dessa forma, vários casos de uso podem incorporar o comportamento desse caso de uso comum. A representação visual desse relacionamento na UML é demonstrada a seguir:



3. Extensão: O relacionamento de extensão é empregado para representar situações em que diferentes sequências de interações podem ser adicionadas a um mesmo caso de uso. Isso significa que cada uma dessas sequências representa um comportamento opcional que só ocorre sob condições específicas. A representação visual desse relacionamento na UML é ilustrada a seguir:

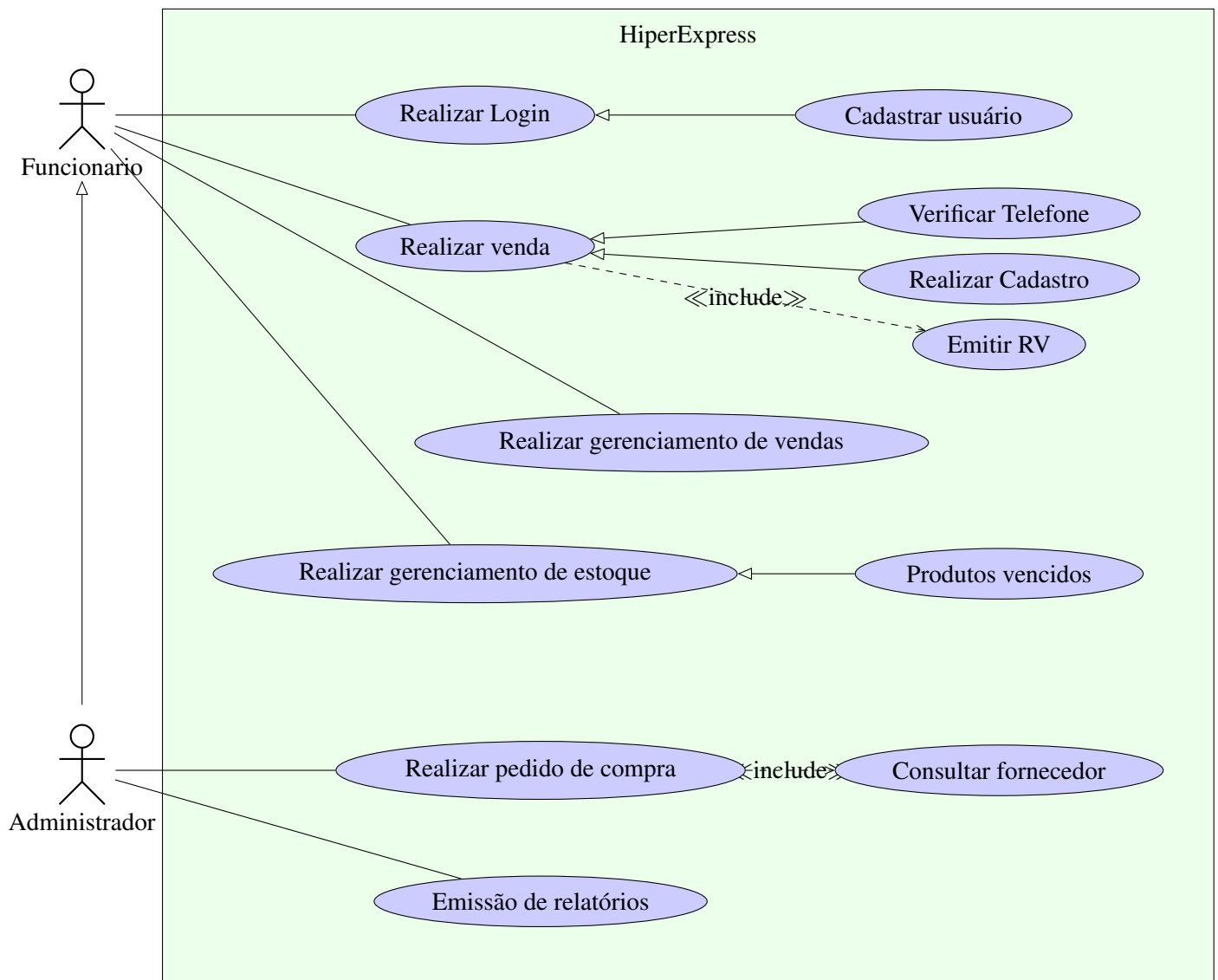


4. O relacionamento de generalização pode ser estabelecido entre dois casos de uso ou entre dois atores. Nesse tipo de relação, um caso de uso ou um ator herda as características de um caso de uso ou ator mais genérico, representando uma relação de especialização. A representação visual desse relacionamento na UML entre dois atores é demonstrada abaixo:



2.2 Diagrama de Caso de Uso

Utilizando os conceitos apresentados na sub seção anterior, foi construído o diagrama de caso de uso para o sistema do SuperEXPRESS. Tal diagrama é apresentado a seguir:



2.3 Explicação

O diagrama acima mostra as funcionalidades utilizadas por cada ator envolvido no sistema de gerenciamento do mercado.

Como pode ser visto, em tal diagrama temos dois atores: o funcionário e o administrador do mercado. Cada funcionário interage com o sistema utilizando as seguintes funcionalidades: Realizar Login, Realizar venda, gerenciar vendas, e gerenciamento de estoque.

O administrador interage com o sistema realizando todas as ações que um funcionário realiza, e também: Pedido de compra e emissão de relatórios.

3 Fluxo de Eventos

Cada cenário de utilização, como aqueles que fazem parte do diagrama de casos de uso anterior, pode ser delineado em texto através de sequência de ações.

De uma perspectiva diferente, é a sequência de ações que descreve um cenário de utilização de forma textual, seguindo um padrão de operação desde o ponto de partida, detalhando as informações que são compartilhadas entre os cenários de utilização.

A seguir, apresentaremos as sequências de ações para 5 cenários de utilização do sistema em questão: Venda simples, verificar produto no estoque, verificar validade de um produto, verificar rendimento diário de um caixa e gerar balanço mensal.

3.1 Caso de Uso 1: Venda Simples

O funcionário ou administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Escolher a opção “Entrada de operador” que aparecerá na tela inicial;
3. Selecionar opção de identificação de cliente via telefone;
4. Ao abrir a aba de cadastro de cliente, preencher os campos (Nome, sobrenome, apelido, endereço, número de telefone, e-mail e CPF), logo após selecionar a opção finalizar cadastro;
5. Informar o código de barras (EAN) ou código interno do produto a ser registrado na venda;
6. Selecionar fechar venda para realizar a finalização;
7. Navegar até a forma de pagamento desejada e selecionar para realizar a finalização da venda;
8. Em caso de troca de operador ou fechamento do caixa, selecionar opção “Saída de Operador”;

3.2 Caso de Uso 2: Verificar produto no estoque

O funcionário ou administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Inserir as informações para filtro, sendo “Nome”, “Código de barras” ou “Código interno”;
3. Aparecerá assim, o item em questão. Para pesquisa de outro filtro, entrar com as informações do item;
4. De posse da informação buscada, fechar a aba de estoque e se desconectar do sistema caso não queira realizar outra operação;

3.3 Caso de Uso 3: Verificar validade de um produto

O funcionário ou administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Escolher a opção “Estoque” que aparecerá na tela inicial;
3. Inserir as informações para filtro, sendo “Nome” ou “Código de barras”;
4. Aparecerá assim, o item em questão, nas informações do produto, terá a coluna de validade. Para pesquisa de outro filtro, entrar com as informações do item;
5. De posse da informação buscada, fechar a aba de estoque e se desconectar do sistema caso não queira realizar outra operação;

3.4 Caso de Uso 4: Verificar rendimento diário de um caixa

O funcionário ou administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Selecionar a opção gerenciamento de vendas;
3. Inserir as informações para filtro, sendo “PDV” que no caso é “Ponto de Venda”, “Operador” e “Data”;
4. Selecionar o PDV listado;
5. De posse da informação buscada, fechar a aba de gerenciamento de vendas e se desconectar do sistema caso não queira realizar outra operação;

3.5 Caso de Uso 5: Gerar balanço mensal

O administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Selecionar a opção emissão de relatórios;
3. Filtrar selecionando a data inicial e data final em que quer realizar o balanço;
4. Será gerado o balanço;
5. De posse da informação buscada, fechar a aba de emissão de relatórios e se desconectar do sistema caso não queira realizar outra operação;

3.6 Caso de Uso 6: Cancelar Venda

O funcionário ou administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Escolher a opção “Entrada de operador” que aparecerá na tela inicial;
3. Selecionar opção de identificação de cliente via telefone;
4. Ao abrir a aba de cadastro de cliente, preencher os campos (Nome, sobrenome, apelido, endereço, número de telefone, e-mail e CPF), logo após selecionar a opção finalizar cadastro;
5. Informar o código de barras (EAN) ou código interno do produto a ser registrado na venda;
6. Selecionar fechar venda para realizar a finalização;
7. Selecionar cancelamento de venda;

3.7 Caso de Uso 7: Incluir um administrador

O administrador do sistema realizará os seguintes passos:

1. O administrador do sistema realiza o login.
2. Na tela inicial, seleciona a opção "Gerenciar Usuários".
3. Escolhe a opção "Incluir Administrador".
4. Preenche os campos de informações do novo usuário.
5. Seleciona a opção "Finalizar Cadastro".

3.8 Caso de Uso 8: Editar um administrador

O administrador do sistema realizará os seguintes passos:

1. O administrador do sistema realiza o login.
2. Na tela inicial, seleciona a opção "Gerenciar Usuários".
3. Escolhe o usuário que deseja editar.
4. Realiza as alterações necessárias nos campos de informações do usuário.
5. Seleciona a opção "Salvar Edições".

3.9 Caso de Uso 9: Remover um administrador

O administrador do sistema realizará os seguintes passos:

1. O administrador do sistema realiza o login.
2. Na tela inicial, seleciona a opção "Gerenciar Usuários".
3. Escolhe o usuário que deseja remover.
4. Seleciona a opção "Remover Usuário".
5. Confirma a exclusão.

3.10 Caso de Uso 10: Alterar a senha de um administrador

O administrador do sistema realizará os seguintes passos:

1. Realizar login no sistema utilizando suas credenciais de acesso;
2. Na tela inicial, seleciona a opção "Configurações de Conta".
3. Escolhe a opção "Alterar Senha".
4. Preenche a senha atual e a nova senha.
5. Confirma a alteração da senha.

3.11 Caso de Uso 11: Incluir cliente e definir o id do cliente.

O administrador ou funcionário do sistema realizará os seguintes passos:

1. Na tela inicial, seleciona a opção "Clientes".
2. Escolhe a opção "Incluir Cliente".
3. Preenche os campos (Nome, Sobrenome, Apelido, Endereço, Número de Telefone, E-mail e CPF) para cadastrar o cliente.
4. Seleciona a opção "Finalizar Cadastro".
5. Confirma a alteração da senha.

3.12 Caso de Uso 12: Editar cliente.

O administrador ou funcionário do sistema realizará os seguintes passos:

1. Na tela inicial, seleciona a opção "Clientes".
2. Escolhe o cliente que deseja editar.
3. Realiza as alterações necessárias nos campos de informações do cliente.
4. Seleciona a opção "Salvar Edições".

3.13 Caso de Uso 13: Remover cliente

O administrador ou funcionário do sistema realizará os seguintes passos:

1. Na tela inicial, seleciona a opção "Clientes".
2. Escolhe o cliente que deseja remover.
3. Seleciona a opção "Remover Cliente".
4. Confirma a exclusão.

4 Sequência

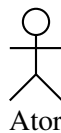
4.1 Descrição

Um diagrama de sequência é uma representação gráfica que descreve os eventos de entrada e saída relacionados a um sistema específico. Ele apresenta, para um cenário particular de um caso de uso, os eventos gerados pelos atores externos, a ordem em que ocorrem e as interações com outros sistemas. Em termos simples, o diagrama de sequência visualiza as interações entre atores e as ações que eles iniciam.

O diagrama de sequência também possui um conjunto de elementos gráficos. Alguns deles como atores, objetos, linhas de vida, e mensagens.

4.1.1 Atores

Os atores são representados como ícones simples, geralmente caixas com o nome do ator dentro. Eles representam os participantes externos que interagem com o sistema ou com os objetos do sistema.



4.1.2 Objetos

Os objetos são representados por retângulos com o nome do objeto dentro. Eles são usados para representar instâncias de classes ou componentes do sistema que participam das interações no diagrama de sequência.

objeto:NomeDaClasse

4.1.3 Linhas de Vida

Uma linha de vida é uma linha vertical que se estende ao longo do diagrama de sequência e representa a existência temporal de um objeto durante a interação. Ela é associada a um objeto e mostra quando o objeto está ativo e quando não está.

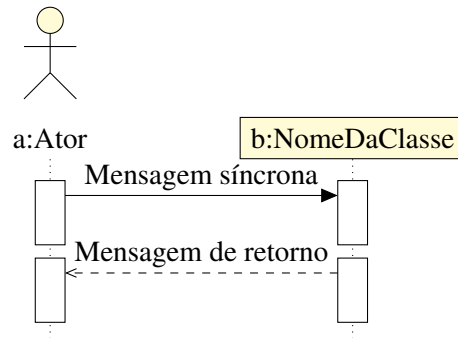


a:Ator

b:NomeDaClasse

4.1.4 Mensagens

As mensagens são setas direcionadas que conectam objetos e representam a comunicação entre eles. Existem diferentes tipos de mensagens, como mensagens síncronas (chamadas de método que bloqueiam o remetente até que o receptor responda) e mensagens assíncronas (chamadas de método que não bloqueiam o remetente).



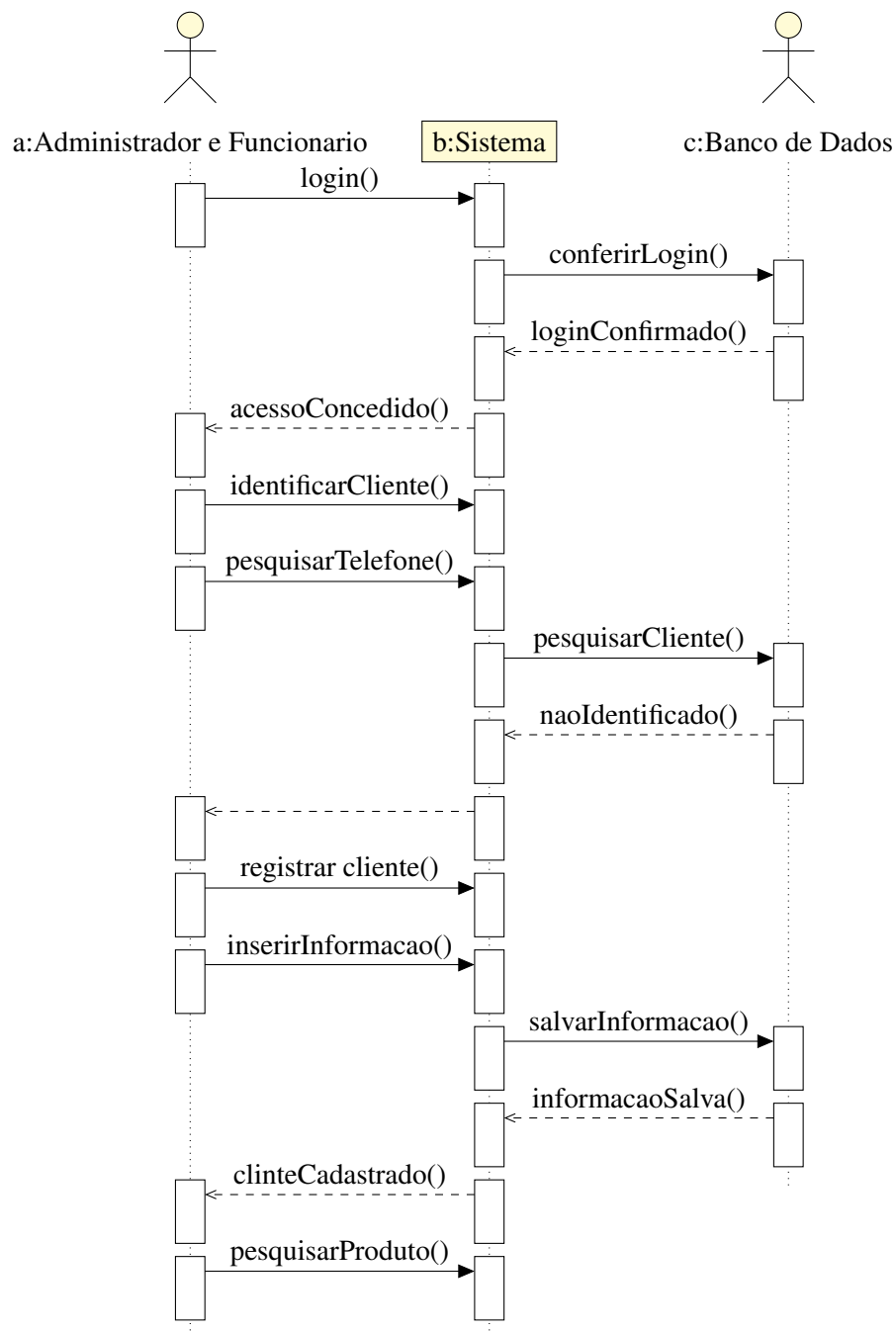
4.1.5 Ocorrência de Execução

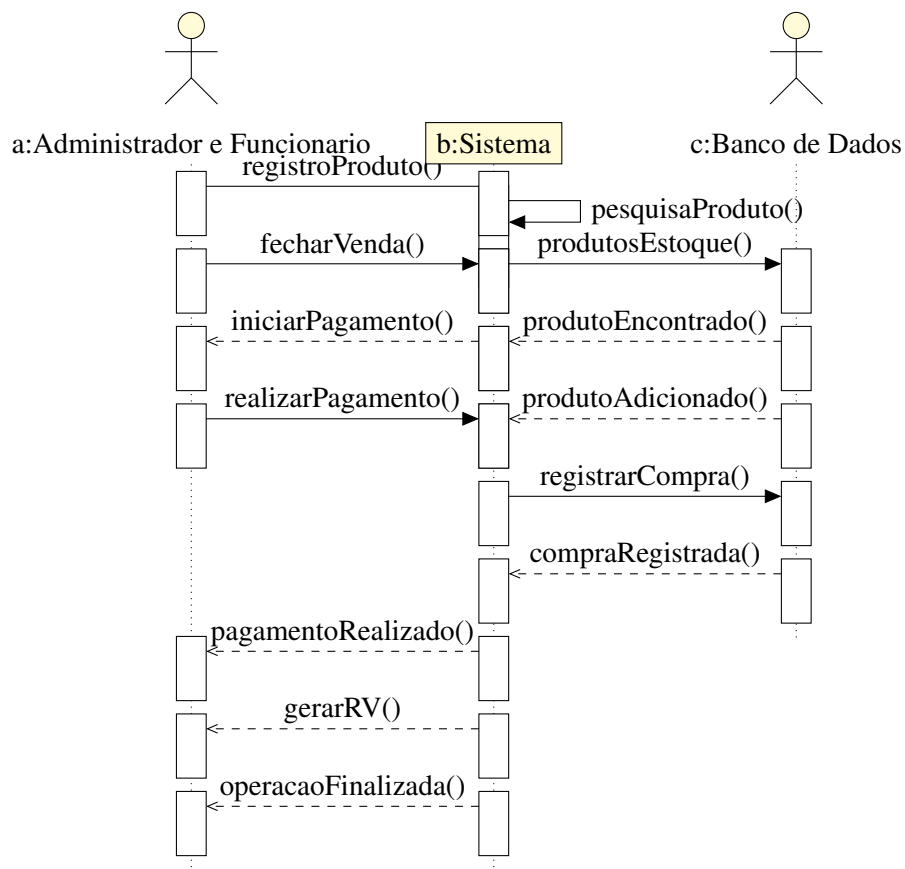
As ocorrências de execução são retângulos posicionados sobre a linha de vida de um objeto para mostrar o período durante o qual o objeto está executando uma operação específica. Isso ajuda a representar a duração das atividades do objeto.

4.2 Diagramas de Sequência

Utilizando os conceitos apresentados na sub seção anterior, foram construídos os diagramas de sequência para os 5 casos de uso utilizados para realização do fluxo de eventos, sendo eles: Verificar produto no estoque, realizar uma venda simples, verificar validade de um produto, verificar rendimento diário de um caixa e gerar balanço mensal.

4.2.1 Diagrama 1: Venda Simples

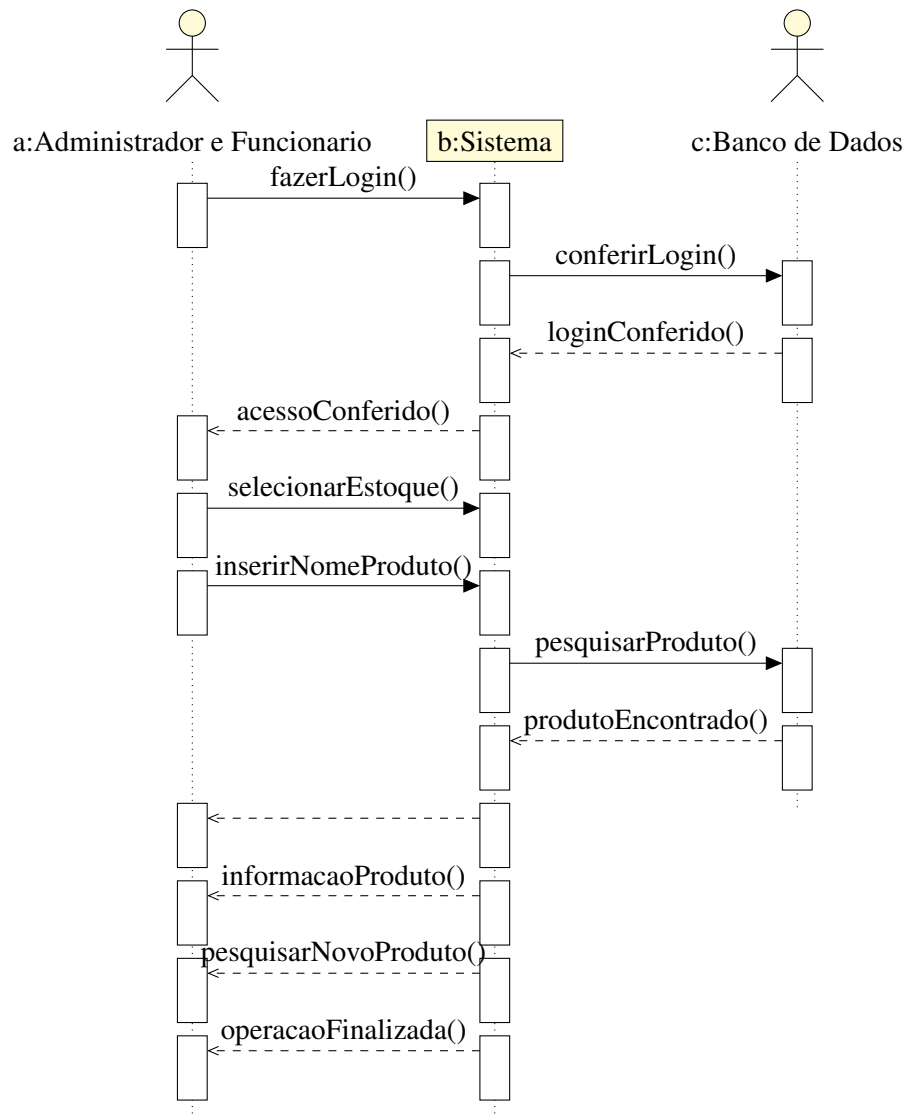




O diagrama de sequência apresentado acima, mostra os passos seguidos para fazer uma venda simples. No mesmo, possuímos o ator (o funcionário), e uma classe (Sistema) e um ator (banco de Dados). Inicialmente, o funcionário envia uma mensagem ao sistema (fazerLogin()) na qual são recebidos o login e a senha. O sistema, por sua vez, utiliza-se do método (conferirLogin()), a fim de verificar a existência do usuário.

Feito o login, o está liberado o acesso para o funcionario executar o processo de compra. Inicialmente, pesquisamos o cliente pelo numero de telefone, se não haver registro do cliente o sistema retorna uma ação de cadastro desse cliente. Após isso, iniciamos o processo de pesquisa de produto. O funcionario pesquisa o produto no sistema e o sistema procura esse produto no banco de dados. Ao encontrar esse produto inicia-se o processo de pagamento, seguido do registro de compra e ao final, o sistema gera o registro de venda (RV) e finaliza a operação.

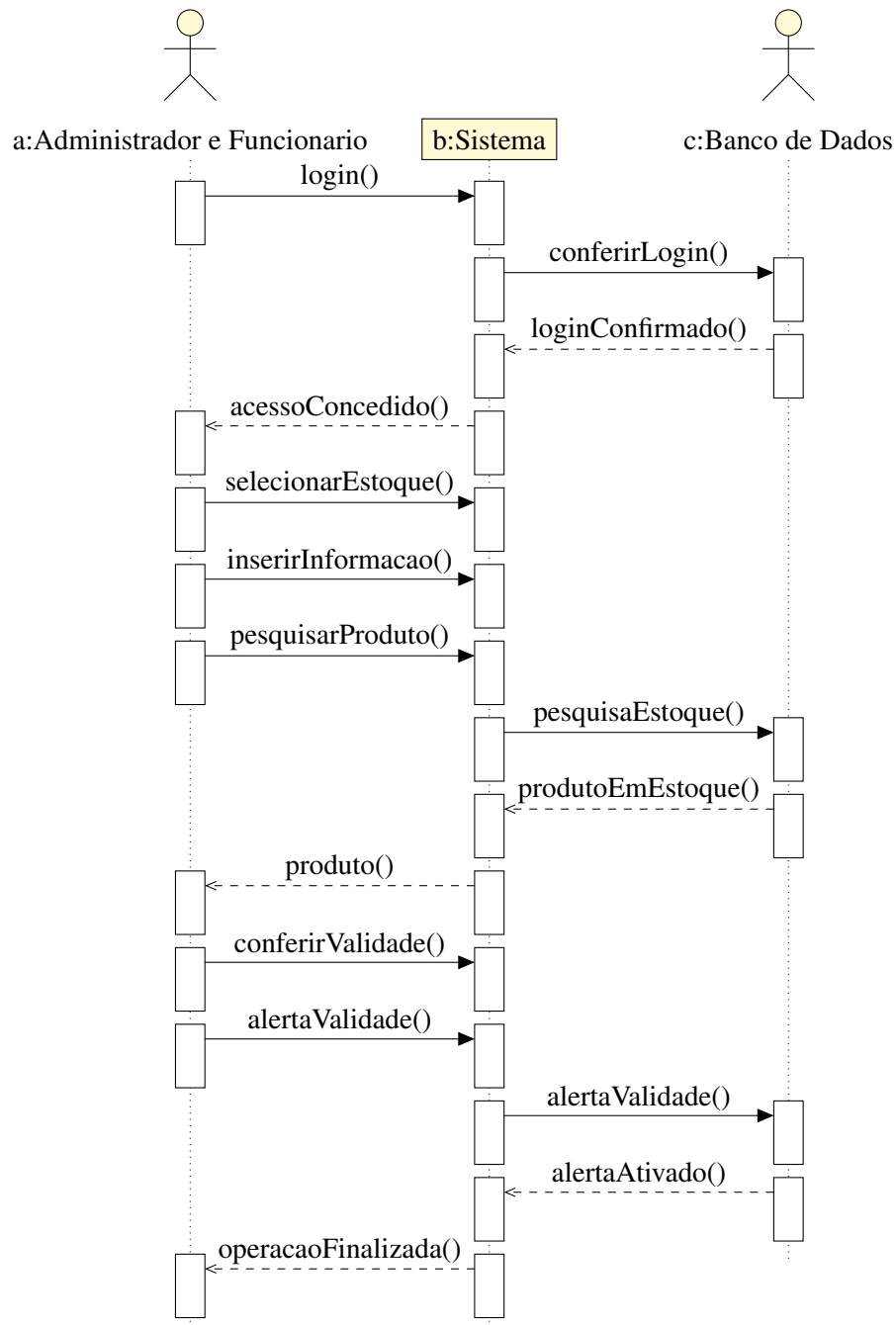
4.2.2 Diagrama 2: Verificar Produto em Estoque



O diagrama de sequência apresentado acima, mostra os passos seguidos para verificar produto em estoque. No mesmo, possuímos o ator (o funcionário), uma classe (Sistema) e um ator (Banco de Dados). Inicialmente, o funcionário envia uma mensagem ao sistema na qual são recebidos o login e a senha. O sistema, por sua vez, utiliza-se do método `conferirLogin`, a fim de verificar a existência do usuário.

Apos a etapa de login o funcionario seleciona a opção estoque no sistema e pesquisa um produto pelo nome. O sistema irá pesquisar o produto no banco de dados e retornar caso o produto exista no estoque. Pro fim, o sistema pergunta ao usuário se deseja fazer outra pesquisa e finaliza operação em seguida.

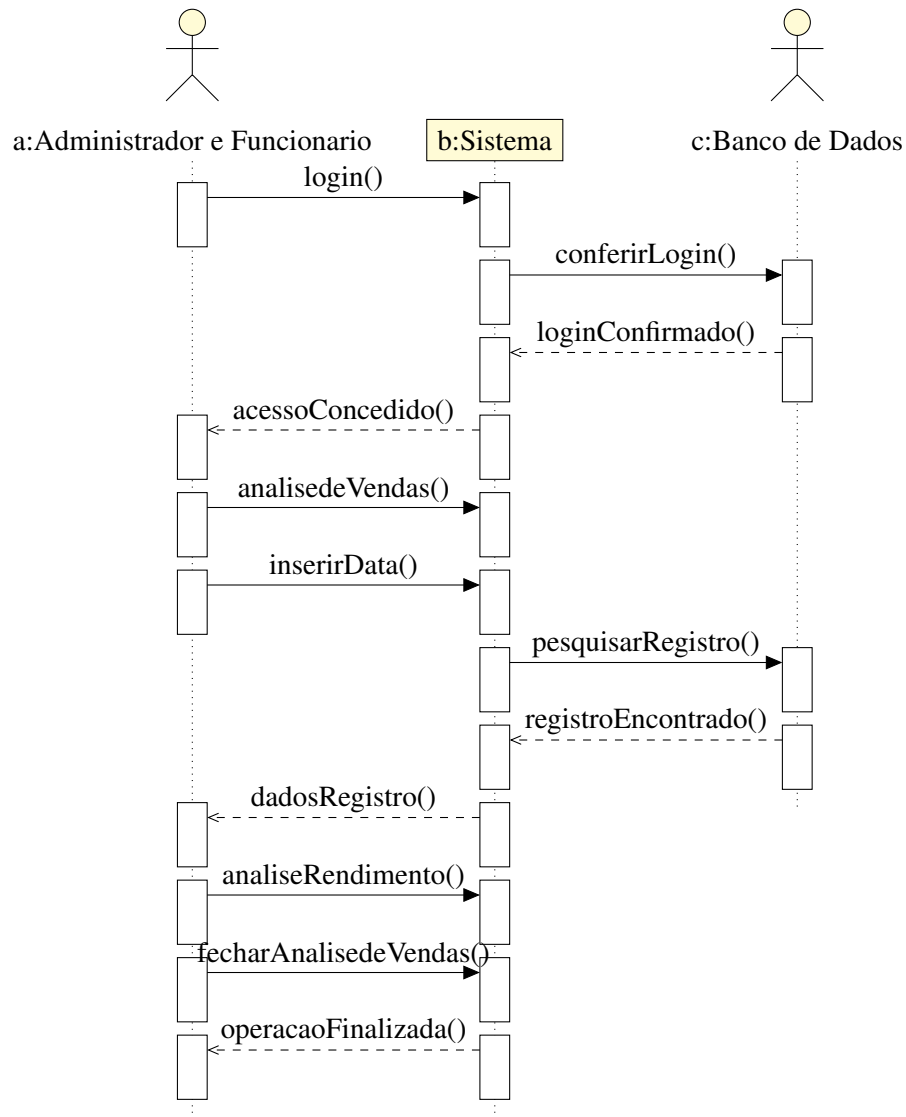
4.2.3 Diagrama 3: Verificar Validade dos Produtos



O diagrama de sequência apresentado acima, mostra os passos seguidos para verificar validade dos produtos em estoque. No mesmo, possuímos o ator (o funcionário), uma classe (Sistema) e um ator (Banco de Dados). Inicialmente, o funcionário envia uma mensagem ao sistema na qual são recebidos o login e a senha. O sistema, por sua vez, utiliza-se do método `conferirLogin()`, a fim de verificar a existência do usuário.

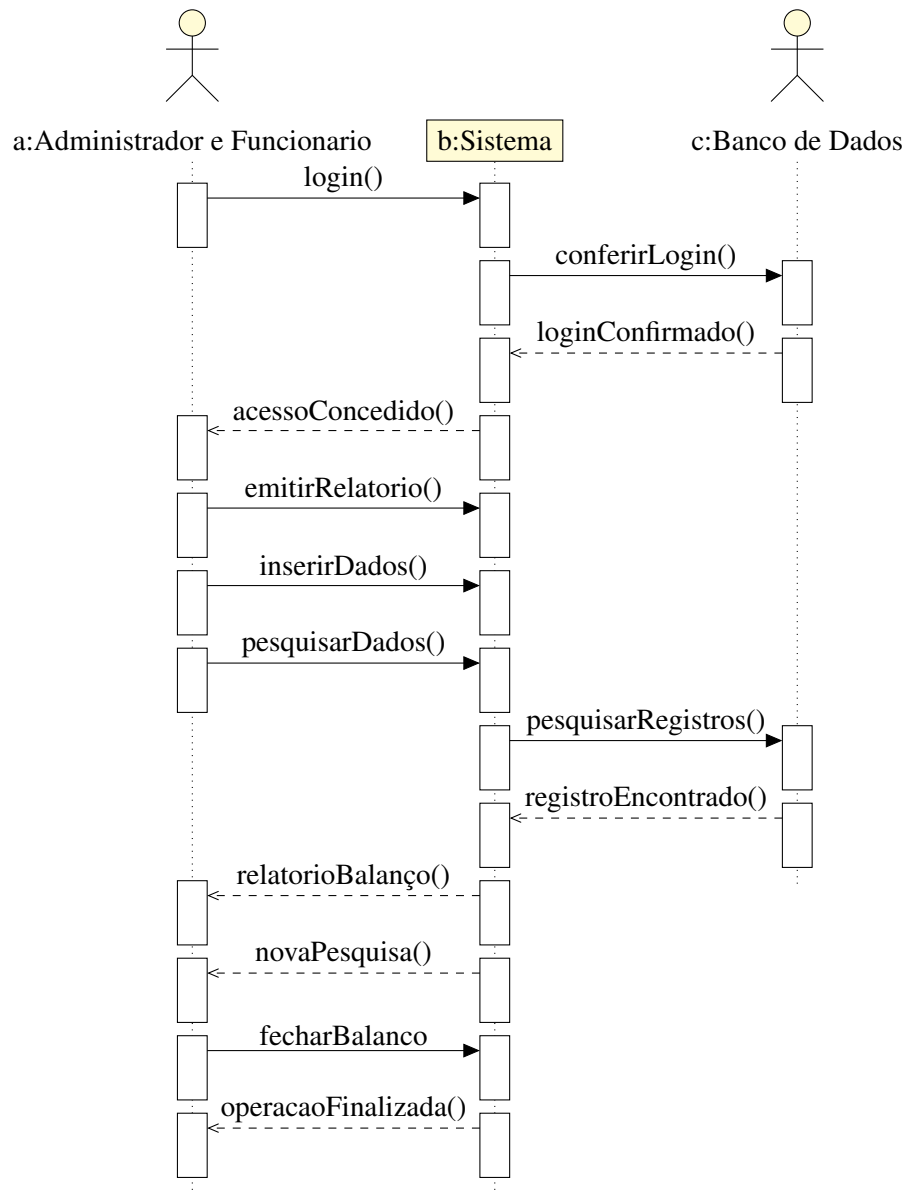
Apos, o funcionario selecionará a opção estoque onde estão registrados os produtos do mercadinho. Com o estoque selecionado, o usuário filtra os produtos com base nas informações e realiza a pesquisa, caso o produto seja encontrado, o sistema retorna as informações do produto para o funcionario fazer a analise de validade. O sistema tambem retorna a opção de ativar o alerta de validade dos produtos em estoque.

4.3 Diagrama 4: Verificar Rendimento de cada Caixa



O diagrama de sequência apresentado acima, mostra os passos seguidos para a análise de rendimento diário de uma caixa. No mesmo, possuímos o ator (o funcionário), uma classe (Sistema) e um ator (Banco de Dados). Inicialmente, o funcionário envia uma mensagem ao sistema na qual são recebidos o login e a senha. O sistema, por sua vez, utiliza-se do método "conferirLogin", a fim de verificar a existência do usuário.

4.4 Diagrama 5: Gerar Balanço Mensal



O diagrama de sequência apresentado acima, mostra os passos seguidos para gerar o balanço mensal. No mesmo, possuímos o ator (o administrador), uma classe (Sistema) e um ator (Banco de dados). Inicialmente, o administrador envia uma mensagem ao sistema na qual são recebidos o login e a senha. O sistema, por sua vez, utiliza-se do método "conferirLogin()" a fim de verificar a existência do usuário.

O administrador envia uma mensagem ao sistema solicitando a emissão do relatório mensal através da qual passa o mês e o ano do relatório. O sistema então envia ao banco de dados para verificar a existência de informações referentes ao período selecionado para gerar o relatório. Caso exista, o sistema responde com a mensagem que imprime o relatório (relatorioBalanço()).

5 Classes

5.1 Descrição

Um diagrama de classes é uma representação visual que visa criar uma visão estática de um sistema em desenvolvimento. Ele exibe um conjunto de classes, interfaces e suas interações, destacando como os elementos do sistema estão organizados e se relacionam entre si.

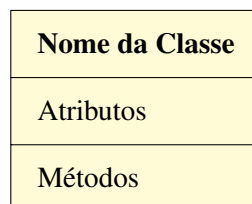
Esses diagramas são usados em todas as etapas do processo de desenvolvimento, desde a análise inicial até a especificação detalhada. Eles desempenham um papel crucial na visualização, definição e documentação da estrutura do sistema.

Para compreender um diagrama de classes, é importante dominar alguns conceitos-chave, que são os seguintes:

5.1.1 Classes

Em programação orientada a objetos, uma classe atua como um modelo ou um "molde" para a criação de objetos. Dentro de uma classe, você define os atributos que os objetos criados a partir dela terão, bem como o comportamento ou funcionalidades que esses objetos possuirão. Essas funcionalidades são geralmente definidas por meio de métodos, que são funções ou procedimentos que podem ser chamados nos objetos para executar ações específicas.

Em resumo, uma classe serve como um plano ou uma descrição de como os objetos de um determinado tipo devem ser criados e quais ações eles podem realizar, com base nos atributos e métodos definidos dentro da classe. Quando você cria instâncias (objetos) dessa classe, elas herdam as características e comportamentos definidos na classe, tornando possível criar e gerenciar objetos com facilidade e estrutura.



5.1.2 Associações

Os objetos de um sistema, que são instâncias das classes, têm a capacidade de colaborar entre si trocando mensagens para desempenhar as diversas funcionalidades desse sistema. Esse tipo de interação e conexão entre objetos é conhecido como "associação".

Em UML, as associações são representadas por linhas (normalmente segmentos de reta) que ligam as classes às quais pertencem os objetos relacionados, conforme mostra a imagem a seguir:



5.1.3 Herança

Ao contrário das relações de associação, que envolvem a interação entre objetos, as relações de herança ocorrem entre classes dentro de um sistema. Nesse tipo de relação, as classes mais específicas, chamadas de subclasses ou filhas, herdam características da classe mais geral, chamada de superclasse ou classe pai. Essa herança abrange atributos, métodos e até mesmo associações. A expressão "é um tipo de" ilustra bem esse tipo de relação.

Na UML, as relações de herança são representadas visualmente por uma seta que parte da sub-classe em direção à superclasse, como mostrado na imagem a seguir:



5.1.4 Composição

A relação chamada composição, assim como a herança, ocorre entre classes dentro de um mesmo sistema e envolve a incorporação de um objeto de uma classe em outra classe, onde a segunda classe pode ser composta por um número variável e de tipos diferentes de objetos conforme necessário. A expressão "todo/parte" ilustra adequadamente essa relação.

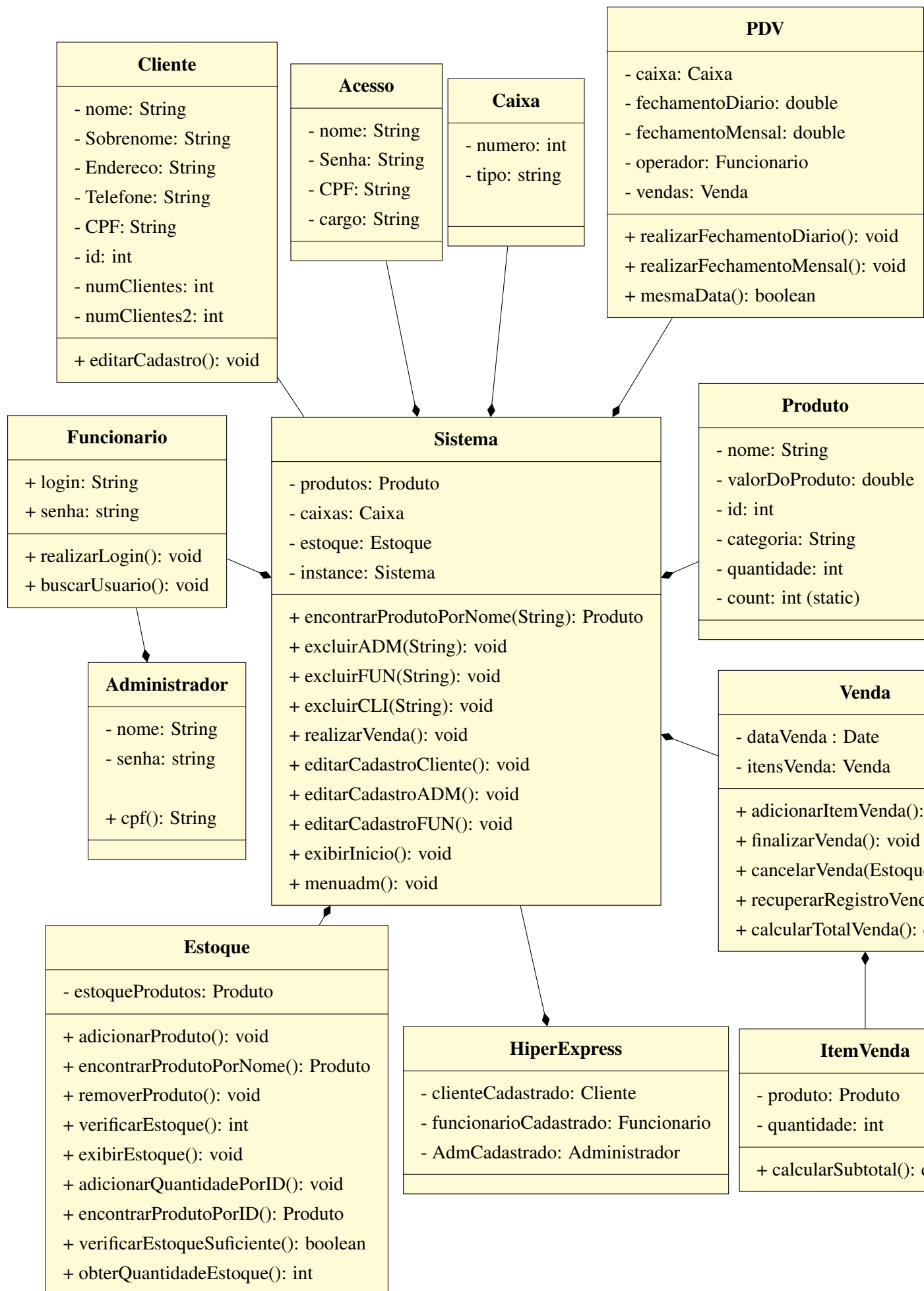
Na UML, as relações de composição são representadas visualmente por uma linha com um losango na extremidade que se conecta à classe que representa a totalidade ou o todo, como mostrado na imagem a seguir:



5.2 Diagrama de Classe

Usando os princípios mencionados anteriormente, foi elaborado um diagrama de classes para o sistema SuperEXPRESS. Neste diagrama, a classe "Sistema" foi definida, incluindo suas características (atributos) e funcionalidades (métodos). Além disso, foi estabelecida uma relação de composição entre a classe "Sistema" e outras classes no diagrama. Isso significa que a classe "Sistema" é composta por instâncias das outras classes, indicando uma conexão "todo/parte" entre elas.

A classe Administrador possui relação de herança com a classe: Funcionario, as outras classes do diagrama são: PedidoDeCompra, Cliente, Produto, GerenciamentoDeCaixa, Venda, AlertaEstoque, Balanco e nosso Sistema.



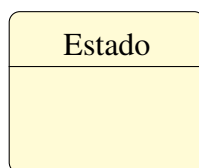
6 Estado

6.1 Descrição

Os diagramas de estado proporcionam uma representação dinâmica de um sistema específico. Eles mostram os eventos significativos e os estados relevantes de um objeto, bem como o comportamento desse objeto em resposta a um evento.

6.1.1 Estado

Um estado é uma condição ou situação específica em que um objeto ou sistema pode existir em um determinado momento. Os diagramas de estado representam os estados relevantes que um objeto pode assumir durante sua vida.



O estado inicial de um objeto, ou seja, o momento em que ele é criado, é representado por um círculo preenchido.

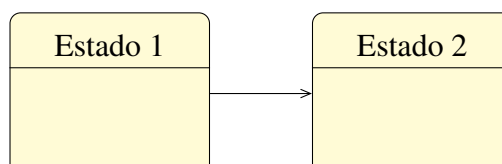


O estado final, ou seja, o fim do ciclo de vida de um objeto é representado por um círculo.



6.1.2 Transições

Uma transição é uma mudança de estado que ocorre em resposta a um evento específico. Ela define como o objeto se move de um estado para outro e pode estar associada a ações ou atividades a serem executadas durante essa transição.

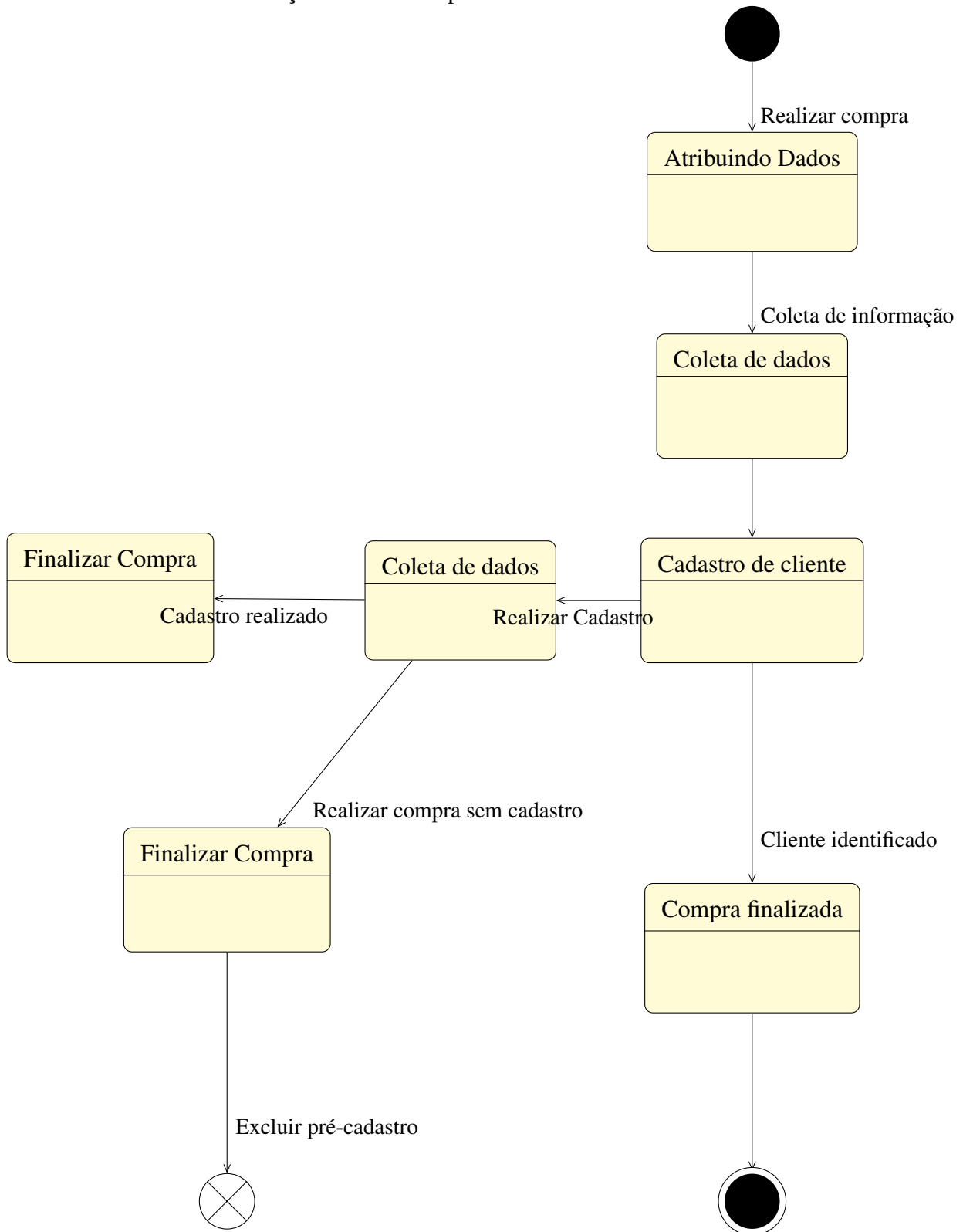


6.1.3 Eventos

Um evento é uma ocorrência significativa que desencadeia uma mudança de estado em um objeto ou sistema. Esses eventos podem ser desencadeados por ações externas, como entradas do usuário, ou por condições internas.

6.2 Diagrama de Estado

Utilizando os conceitos apresentados na sub seção anterior, foi construído o diagrama de estado modelando os estados da realização de uma compra.



Conforme visto acima após a criação de um objeto da classe Venda, inicialmente são atribuídos os dados do cliente para o objeto. Após os atributos do objeto receberem valores, a venda é realizada (verificação do telefone, passagem de produtos etc), e, então, a venda é finalizada.

Em seguida, a venda pode seguir para três caminhos: o cliente é identificado e logo após encaminhado para finalizar a venda, realizar o cadastro para finalizar a venda ou optar por não se identificar para realizar a compra.

7 Conclusão

É evidente que o desenvolvimento deste trabalho destacou a importância crucial de modelar um sistema antes de iniciar a programação efetiva. Além disso, ficou claro o quão desafiador pode ser abstrair um sistema para representá-lo de forma precisa e eficiente.

A contribuição deste trabalho para a vida acadêmica e profissional dos autores é evidente. Foi uma oportunidade valiosa para aprender a utilizar ferramentas como LaTeX e TikZ, que certamente serão úteis ao longo da carreira acadêmica. Além disso, a prática de abstrair e modelar problemas complexos foi uma lição valiosa, complementando os conceitos da disciplina de Programação Orientada a Objetos.

Esse aprendizado não apenas enriqueceu o conhecimento teórico, mas também proporcionou habilidades práticas e ferramentas úteis para futuros projetos e desafios tanto na academia quanto na indústria.

8 Referências Bibliográficas

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. Rio de Janeiro: Elsevier, 2007.

BROOCH, G., RUMBAUGH, J., JACOBSON, I. UML: Guia do Usuário. Rio de Janeiro: Elsevier, 2005.

LARMAN, Craig. Utilizando UML e Padrões. Porto Alegre: Bookman, 2007.

- . Guia Prático de Citações e Referências Segundo as Normas da ABNT. Disponível em: <<http://www.citethis.com.br/abnt>>.

SOUZA, Givaldo. Diagrama de Sequência. Disponível em: <<https://docente.ifrn.edu.br/givanaldorochoa/disciplinas/de-software-licenciatura-em-informatica/diagrama-de-sequencia>>.

FIGUEIREDO, Eduardo. Diagrama de Sequência. Disponível em: <<https://homepages.dcc.ufmg.br/figueiredo/diagrama-sequencia/01.pdf>>.

RICARTE, Ivan. O que é uma classe. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/classes/01-que-e-uma-classe.pdf>>.

ABDALA, Daniel. Diagrama de Estados. Disponível em: <<http://www.facom.ufu.br/~abdala/DAS5312/Diagrama-de-Estados.pdf>>.

8.1 Referências adicionais

Template disponibilizado pela aluna: Ana Letícia César.

Slides do professor Eduardo Pelli nas aulas da disciplina, e disponibilizados no Classroom;

Documentação do TikZ

Documentação do TikZ-UML