

Estudo de Caso

Sistema de Caixa Automático

Curso de Especialização – DEINF - UFMA

Desenvolvimento Orientado a Objetos

Prof. Geraldo Braz Junior

Referências:

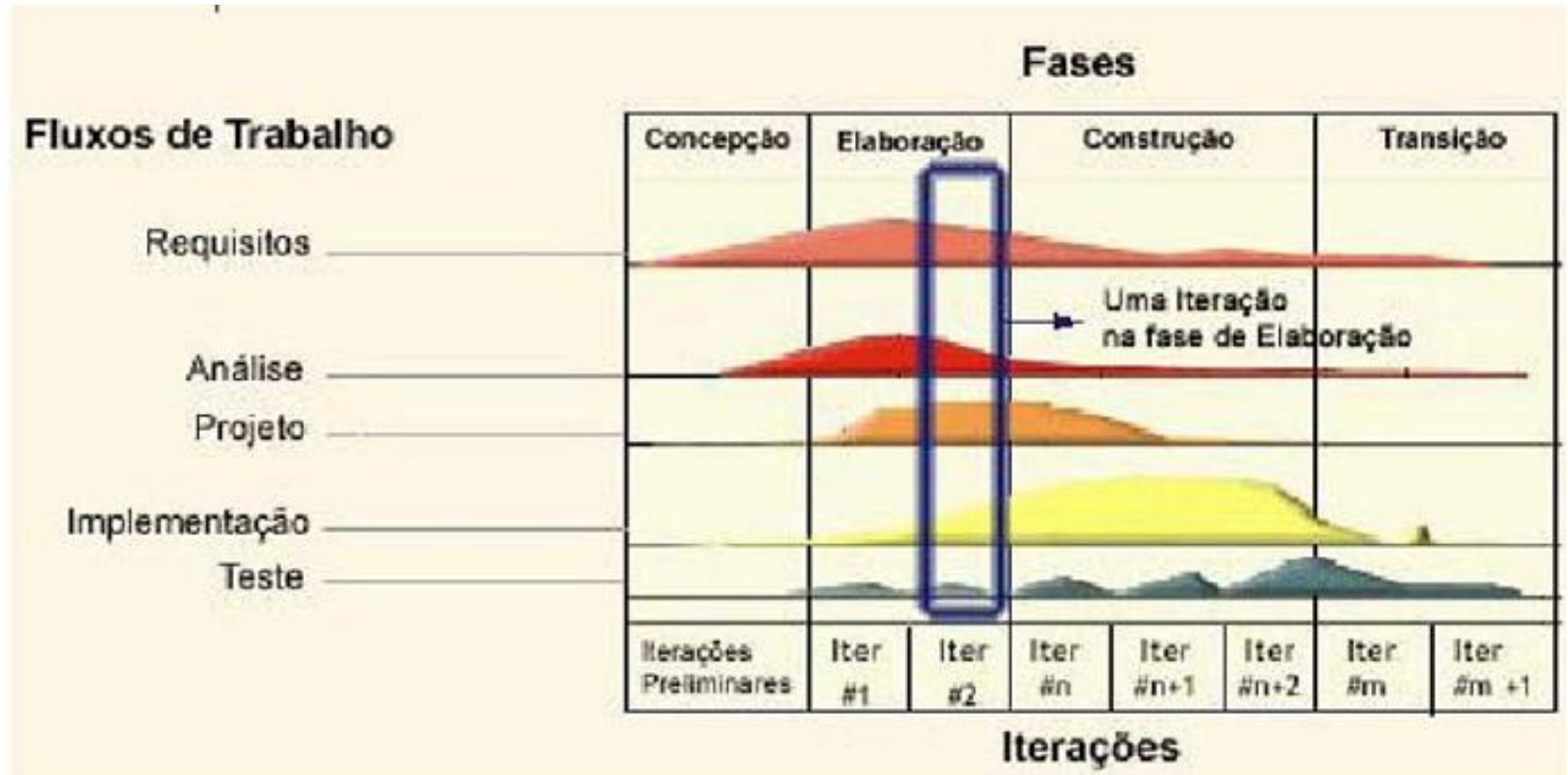
Notas de Aula Ulrich Schiel

Notas de Aula Ariadne

Receita de bolo

- Utilizar os conceitos de Processo Unificado
- Objetivo:
 - Modelar um software de teste (caixa automático)
 - Descobrir como aplicar UML como linguagem de modelagem
 - Descobrir como aplica PU junto com UML

Ciclo de Vida PU



Concepção

Concepção -> Requisitos

Obtenção de Requisitos

- Artefatos (produtos a serem gerados)
 - Requisitos
 - Modelo de Contexto
 - Modelo de Casos de Uso
 - Atores
 - Quem vai utilizar de alguma maneira o sistema
 - Casos de Usos
 - Funcionalidades Percebidas do Sistema

Obtenção de Requisitos

- Passos:
 1. Listar potenciais requisitos
 2. Entender o contexto do sistema
 3. Capturar requisitos funcionais
 4. Capturar requisitos não funcionais

Requisitos e Descrição (i)

- O sistema de caixa automático permite que clientes realizem saques e verifiquem seus saldos, de acordo com as seguintes regras de negócios:
 1. Quando uma conta é criada no banco, o seu saldo é maior que zero.
 2. Um cliente pode possuir várias contas no banco.
 3. O cliente acessa uma conta através do terminal de um caixa eletrônico do seu banco.
 4. Antes de executar qualquer operação da conta, o cliente deve fornecer o número da sua conta e a senha referente a mesma.

Requisitos e Descrição (ii)

5. Para a realização do saque, o cliente utiliza um terminal para solicitar um valor numérico de dinheiro.
6. O cliente pode sacar qualquer quantia do caixa, desde que a mesma seja igual ou inferior ao saldo disponível.
7. Vale a política do banco de que uma conta não aceita uma operação de saque quando a conta está com o saldo zerado. O dinheiro é liberado no dispensador de notas do caixa e debitado do saldo da conta.
8. Além de possuir o dinheiro disponível na conta, em uma operação de saque, a quantidade de dinheiro disponível no caixa eletrônico deve ser maior ou igual à quantia solicitada pelo cliente.

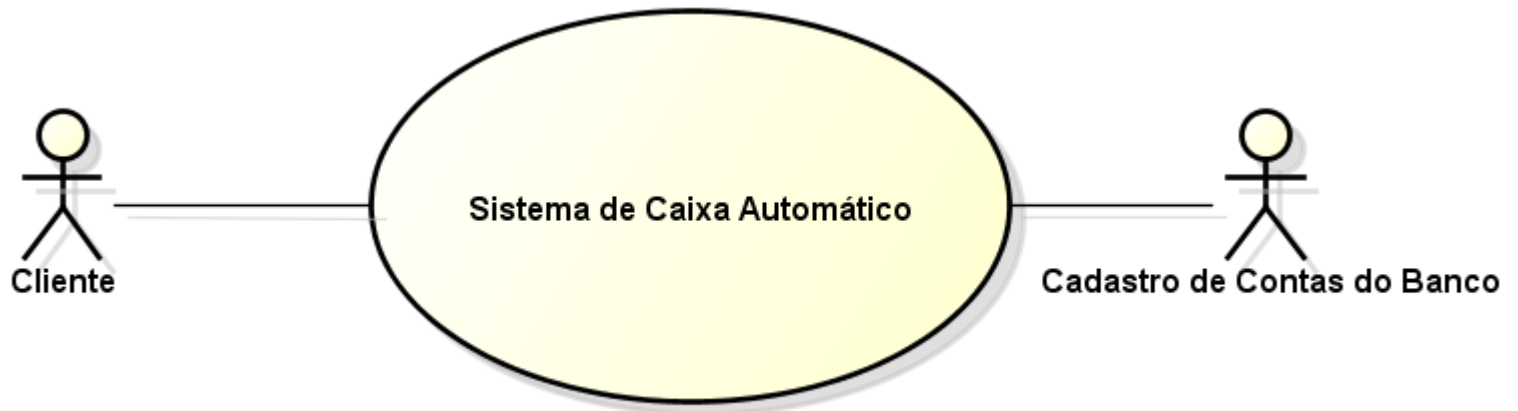
Requisitos e Descrição (iii)

9. Se o saldo de uma conta é zerado durante uma operação de saque, a conta deve se tornar inativa.
10. Os clientes que vão operar o caixa eletrônico devem estar devidamente cadastrados no banco e suas contas devem estar ativas.
11. Cada conta tem associado um número e uma senha.
12. Além disso, cada conta é associada a um cliente do banco, que possui informações como nome, RG, CPF, etc.

Requisitos e Descrição (iv)

13. As informações adicionais sobre as contas e seus clientes estão armazenadas em um Cadastro de Contas do Banco que interage com o Sistema de Caixa Automático.
14. Qualquer cliente cadastrado no banco pode efetuar depósitos em uma conta, quer a conta esteja ativa, quer ela esteja inativa.
15. Caso a conta esteja inativa e após o depósito seu saldo fique maior que zero, a conta deve ser reativada.

Modelo de Contexto (caso de uso nível 0)



powered by astah[®] 

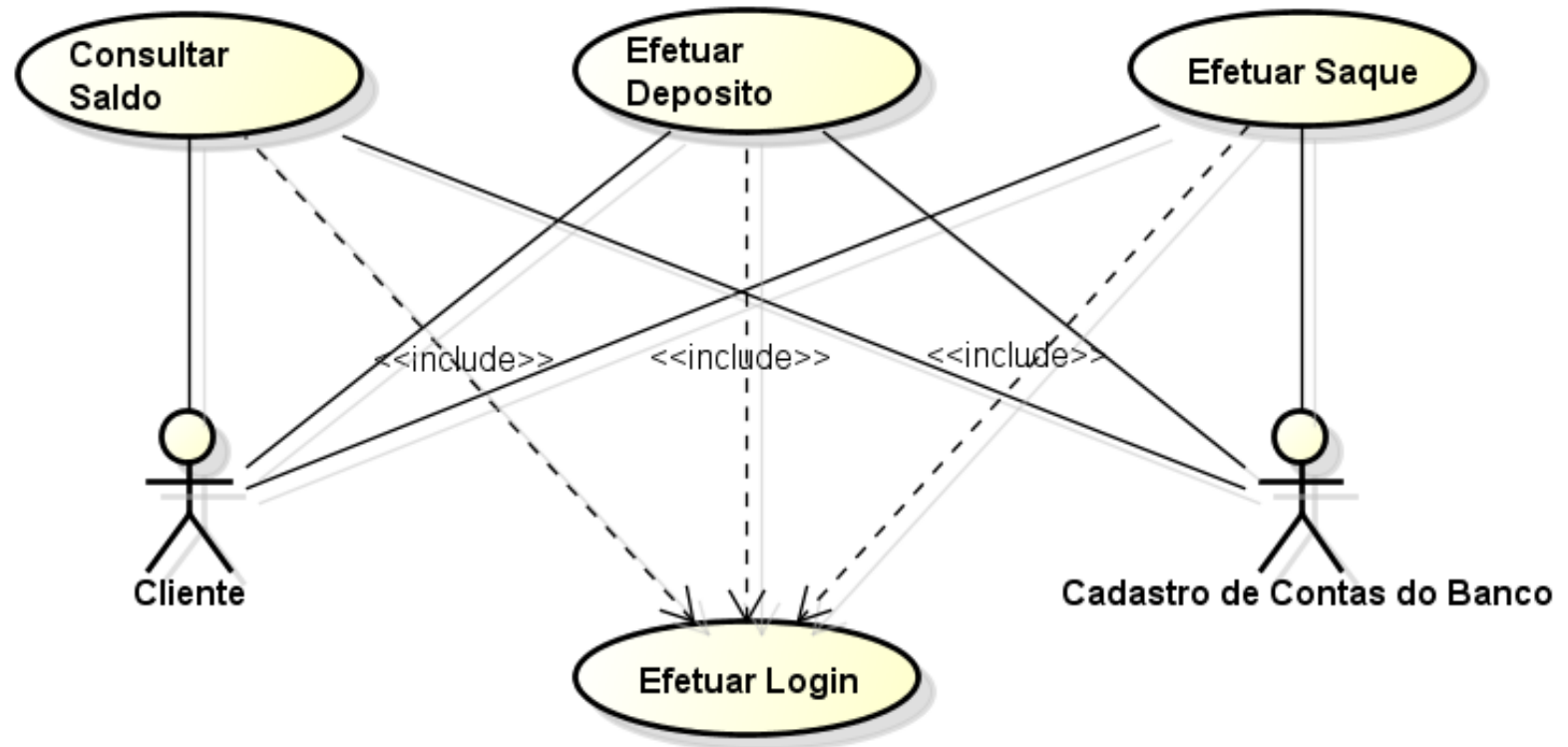
Descrição

A1) encontrar os atores e *use-cases*

- encontrar os atores
- encontrar e descrever cada *use-case*
- descrever o Modelo *Use-Case* como um todo

A2) Priorizar *Use-Cases* (visão arquitetural)

Modelo de Casos de Uso



powered by astah*

Caso de Uso: Consultar Saldo

Breve Descrição: O cliente, já autenticado, escolhe a opção “Consultar Saldo” e o sistema apresenta o seu saldo.

Atores: Cliente, Cadastro de Contas do Banco.

Pré-condição: A conta deve estar ativa e o cliente já deve ter sido autenticado junto ao sistema, através do caso de uso Efetuar Login.

Pós-condição: Estado da conta inalterado.

Requisitos Especiais: nenhum.

Fluxo

Caso de Uso: Efetuar Saque

Breve Descrição: O cliente, já autenticado, escolhe a opção “Efetuar Saque”, informa a quantia desejada e, caso o saldo da conta seja suficiente e o caixa tenha o dinheiro necessário, a quantia é liberada.

Atores: Cliente, Cadastro de Contas do Banco

Pré-condição: O cliente deve estar logado no sistema, através do caso de uso Efetuar Login. Além disso, a conta deve estar ativa e o valor a debitar deve ser maior que zero e não pode ser superior ao saldo da conta nem a quantidade de dinheiro disponível no caixa.

Pós-condição: O valor a ser sacado é subtraído do saldo da conta e do total disponível no caixa eletrônico e a quantia solicitada é fornecida ao cliente.

Requisitos Especiais: nenhum.

Fluxo

Caso de Uso: Efetuar Depósito

Breve Descrição: O cliente, já autenticado, escolhe a opção “Efetuar Depósito”, informa a quantia desejada e, a conta que deseja enviar o dinheiro

Atores: Cliente, Cadastro de Contas do Banco

Pré-condição: O cliente deve estar logado no sistema, através do caso de uso Efetuar Login.

Pós-condição: O valor a ser depositado é adicionado ao saldo da conta.

Requisitos Especiais: nenhum.

Caso de Uso: Efetuar Login

Breve Descrição: O cliente deve fornecer o número da conta e senha, essas informações devem ser autenticadas pelo Cadastro de Contas do Banco.

Atores: Cliente, Cadastro de Contas do Banco

Pré-condição: nenhuma

Pós-condição: Após uma autenticação bem realizada, o usuário está apto a operar o sistema do caixa eletrônico

Requisitos Especiais: nenhum.

Ao final dos requisitos

- Realizar uma proposta
 - Estimativa de custos
 - Definir Prioridades aos Requisitos levantados
 - Análisar os Riscos Esperados

Concepção -> Análise

Análise

- Durante a etapa de Concepção, a análise se resume a definição de uma:
 - Descrição Básica da arquitetura de objetos
- Identifica-se:
 - objetos de negócio (pedidos, contas, contratos,...)
 - objetos do mundo real (veículos, máquinas, trajetos,...)
 - eventos básicos (chegada de um pedido, partida de um transporte, ..)
- Esse trabalho deve ser realizado em paralelo a definição de casos de uso para melhor entender o domínio da aplicação

Elaboração

Elaboração - Requisitos

Requisitos

A3) Detalhar cada *Use-Case*

- estruturar a descrição do *use-case*
- formalizar a descrição do *use-case* (usar diagramas de atividade ou diagramas de interação)
- descrever o Modelo *Use-Case* como um todo

A4) Prototipar as interfaces com o usuário

- projeto lógico da interface do usuário
- projeto físico da interface do usuário e protótipo

Caso de Uso: Consultar Saldo

Caso de Uso: Consultar Saldo

Fluxo Básico :

1. O cliente escolhe no menu principal do terminal a opção “Consultar Saldo”.
2. O sistema verifica se o login foi efetuado
3. O sistema verifica se a conta está ativa, através do Cadastro de Contas do Banco.
4. O sistema obtém o saldo da conta do cliente e o imprime.

Caso de Uso: Consultar Saldo

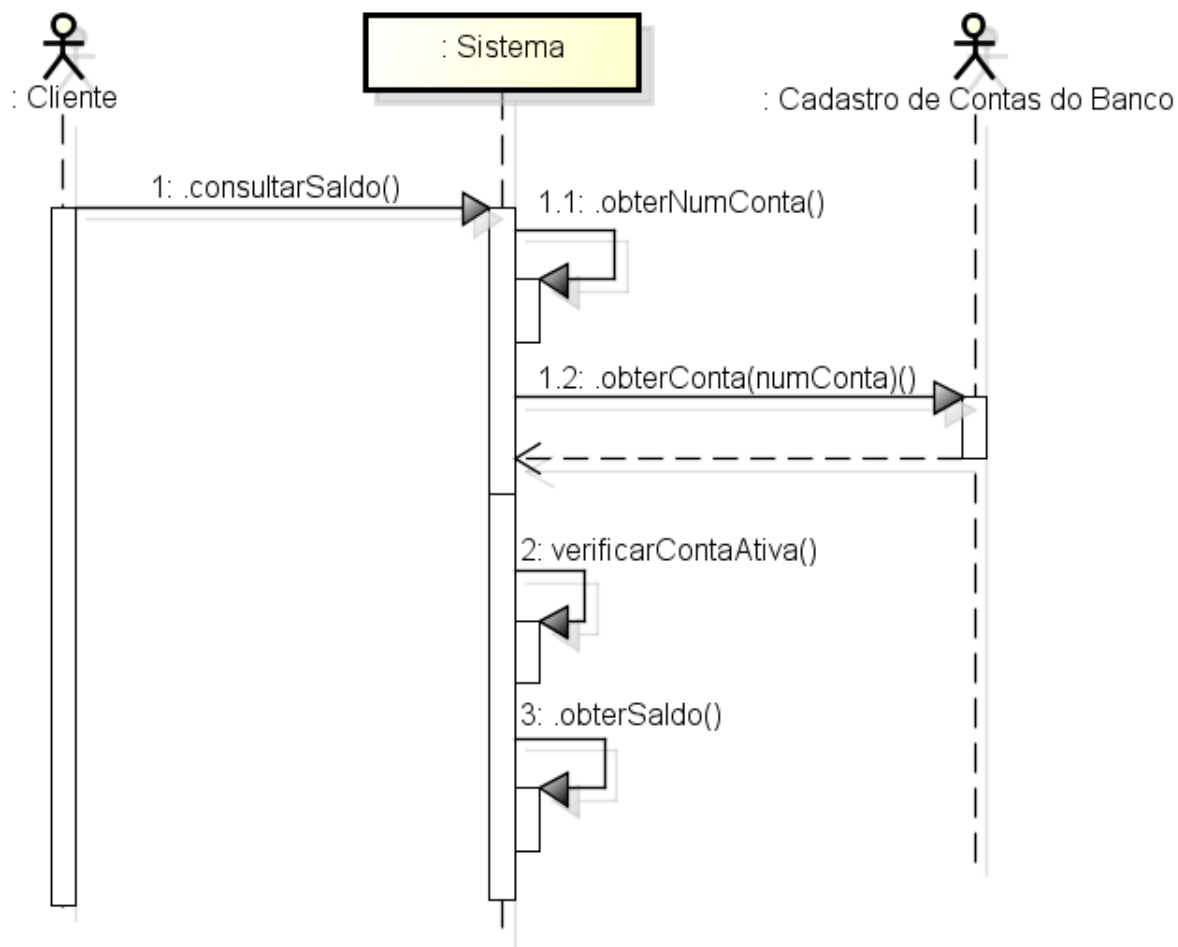
Fluxo Alternativo 1:

No passo 2 do Fluxo Básico, se o login não foi efetuado, o sistema informa isso ao cliente.

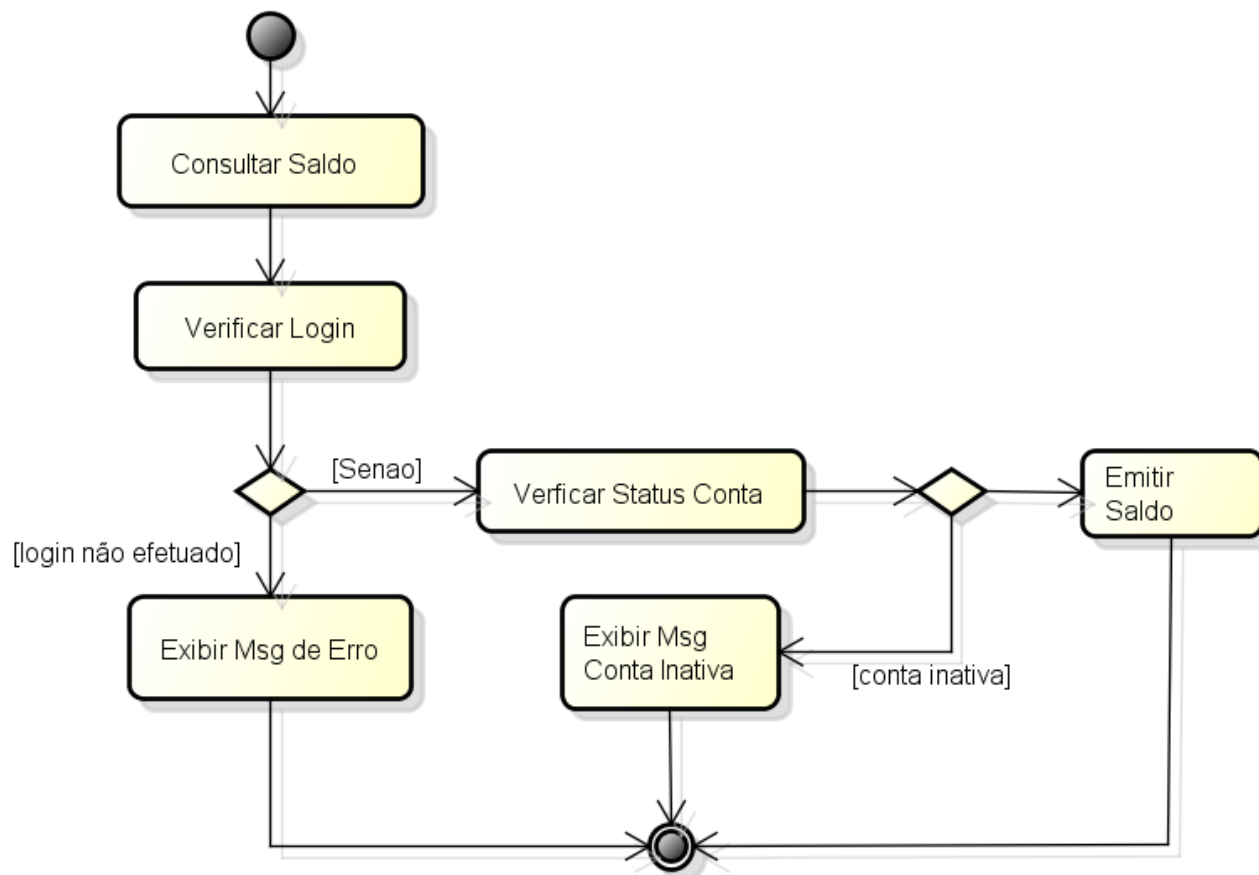
Fluxo Alternativo 2:

No passo 3 do Fluxo Básico, se a conta não estiver ativa, o sistema informa isso ao cliente e avisa que a consulta não pôde ser realizada.

Consultar Saldo - Sequencia



Consultar Saldo - Atividades



Caso de Uso: Efetuar Saque

Caso de Uso: Efetuar Saque

Fluxo Básico:

1. O cliente escolhe no menu principal do terminal a opção “Efetuar Saque”.
2. O sistema verifica se o login foi efetuado.
3. O sistema verifica se a conta está ativa, através do Cadastro de Contas do Banco.
4. O sistema solicita que o cliente informe a quantia desejada.
5. O cliente informa a quantia desejada.
6. O sistema verifica se o saldo da conta é suficiente para realizar a transação e, em caso afirmativo, se há dinheiro em quantidade suficiente no caixa.
7. O sistema subtrai o valor solicitado do saldo da conta do cliente e do valor disponível no caixa e libera a quantia solicitada, através do dispensador de notas.

Caso de Uso: Efetuar Saque

Fluxo Alternativo 1:

No passo 2 do Fluxo Básico, se o login não tiver sido efetuado, o sistema informa isso ao cliente.

Fluxo Alternativo 2:

No passo 3 do Fluxo Básico, se a conta não estiver ativa, o sistema avisa isso ao cliente e informa que o saque não pôde ser realizado.

Caso de Uso: Efetuar Saque

Fluxo Alternativo 3 :

No passo 6 do Fluxo Básico, se o valor solicitado for menor que zero ou superior ao saldo da conta ou a quantidade de dinheiro disponível no caixa, o sistema informa que não é possível realizar o saque e o porquê. Em seguida, volta ao passo 4 do Fluxo Básico.

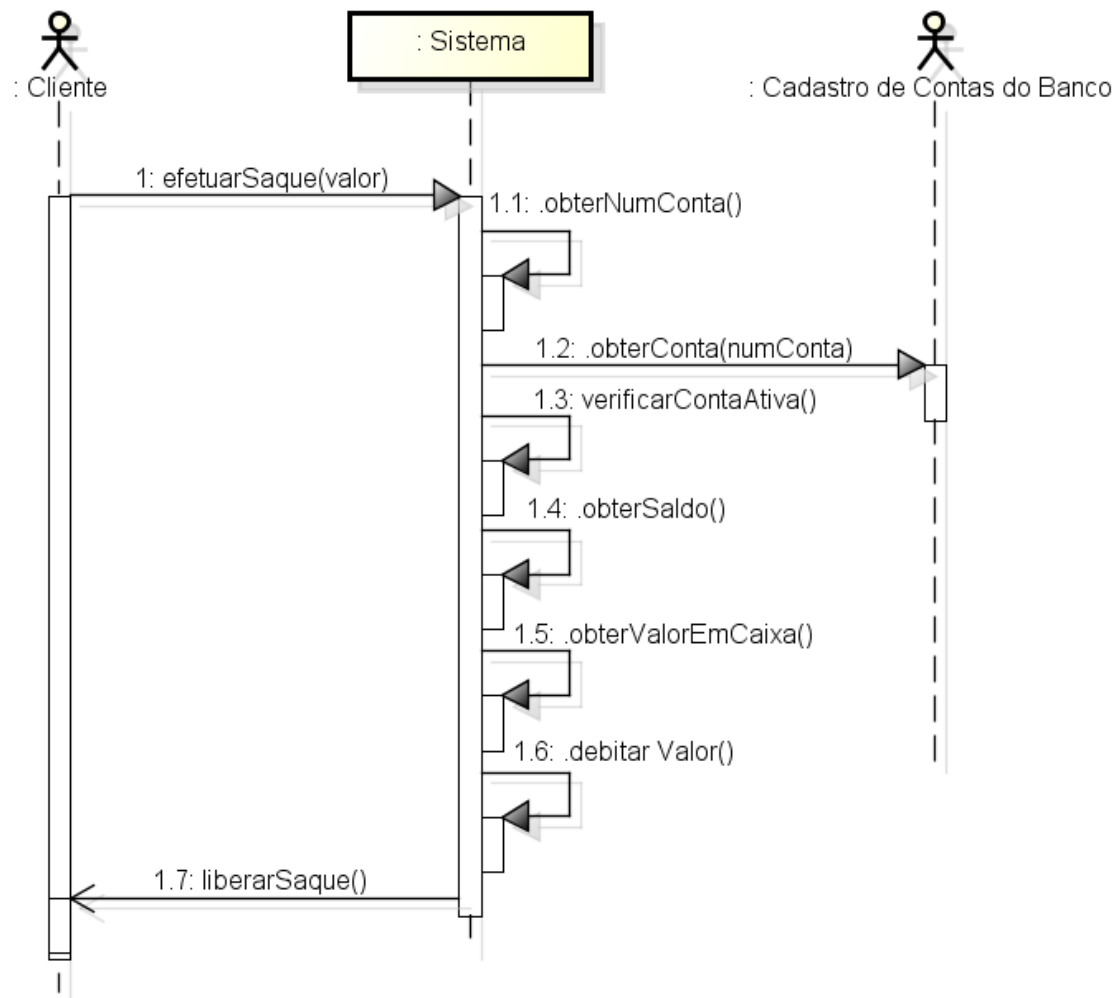
Fluxo Alternativo 4 :

Após o passo 7 do Fluxo Básico, se o saldo da conta for menor ou igual a zero, a conta deve ser desativada.

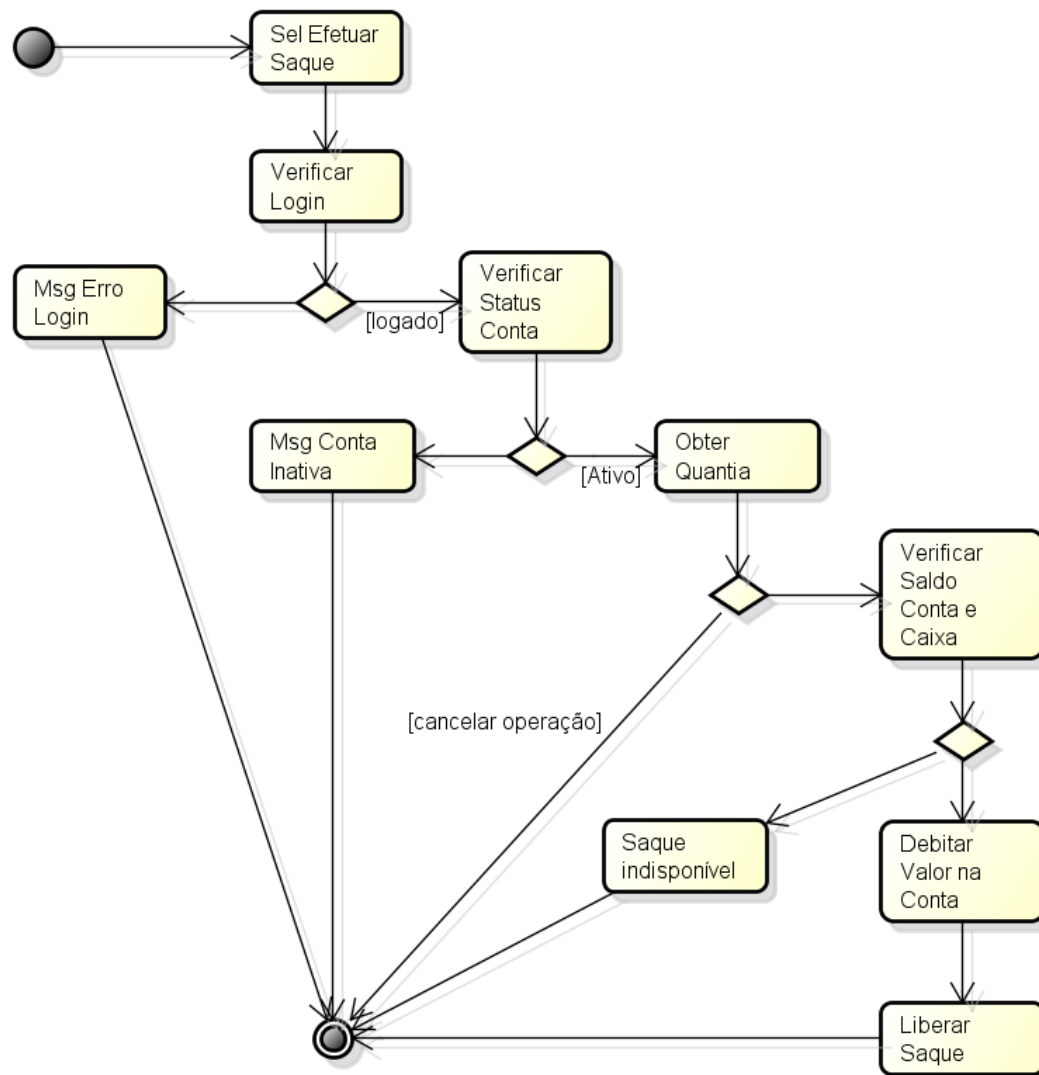
Fluxo Alternativo 5 :

No passo 5 do Fluxo Básico, o cliente pode cancelar a operação.

Efetuar Saque - Sequencia



Efetuar Saque - Atividades

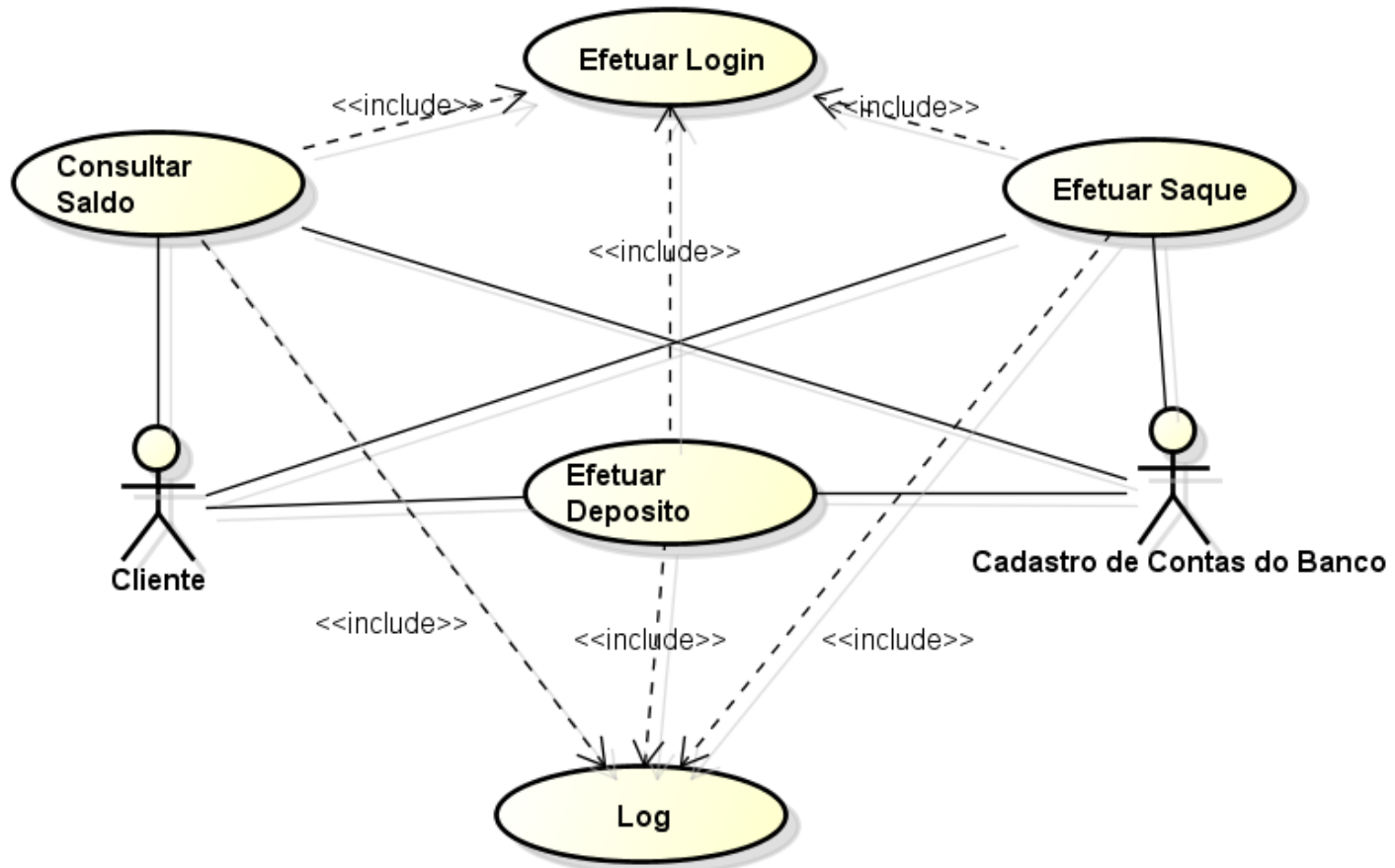


Requisitos - Elaboração

A5) Estruturar o modelo Casos de Uso

- identificar funcionalidades comuns (generalizações)
- identificar funcionalidades adicionais ou opcionais
(`<<extends>>`)
- identificar outros relacionamentos entre *use-cases*
(`<<include>>`, inverso de `<<extend>>`)

Modelo de Casos de Uso



Requisitos - Elaboração

- **Capturar requisitos não-funcionais**
- Usabilidade
 - requisitos de interfaces metáfora, frequência de uso, ..
 - documentação
- Confiabilidade
 - tolerância a falhas.
- Performance
 - tempos de resposta
 - volumes de transações
- Requisitos físicos
 - equipamentos, material, espaços, configurações de rede,
 - software

Concepção -> Análise

Análise

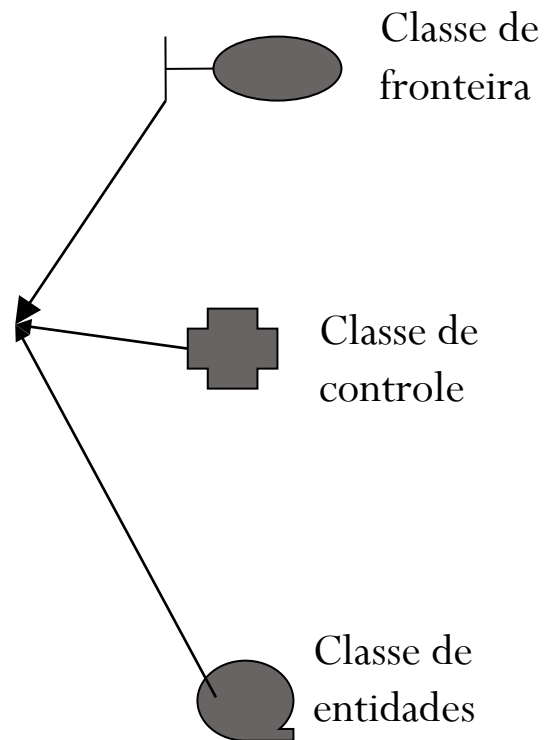
- Os **requisitos externos** são transformados em um **modelo interno** preciso e completo para desenvolver o projeto do sistema

MODELO <i>USE-CASE</i>	MODELO DA ANÁLISE
linguagem do usuário	Linguagem do desenvolvedor
Visão externa do sistema	Visão interna do sistema
Estruturado por <i>use-cases</i>	Estruturado por classes
Captura a funcionalidade do sistema	Descreve como realizar a funcionalidade
Usado para o contrato com o cliente	Usado para o desenvolvedor entender o sistema
Pode conter redundâncias, inconsistências, etc.	Deve ser preciso e inambíguo

Análise - Artefatos

1. MODELO DA ANÁLISE

2. CLASSE DE ANÁLISE



EXEMPLO

Interface de Saque

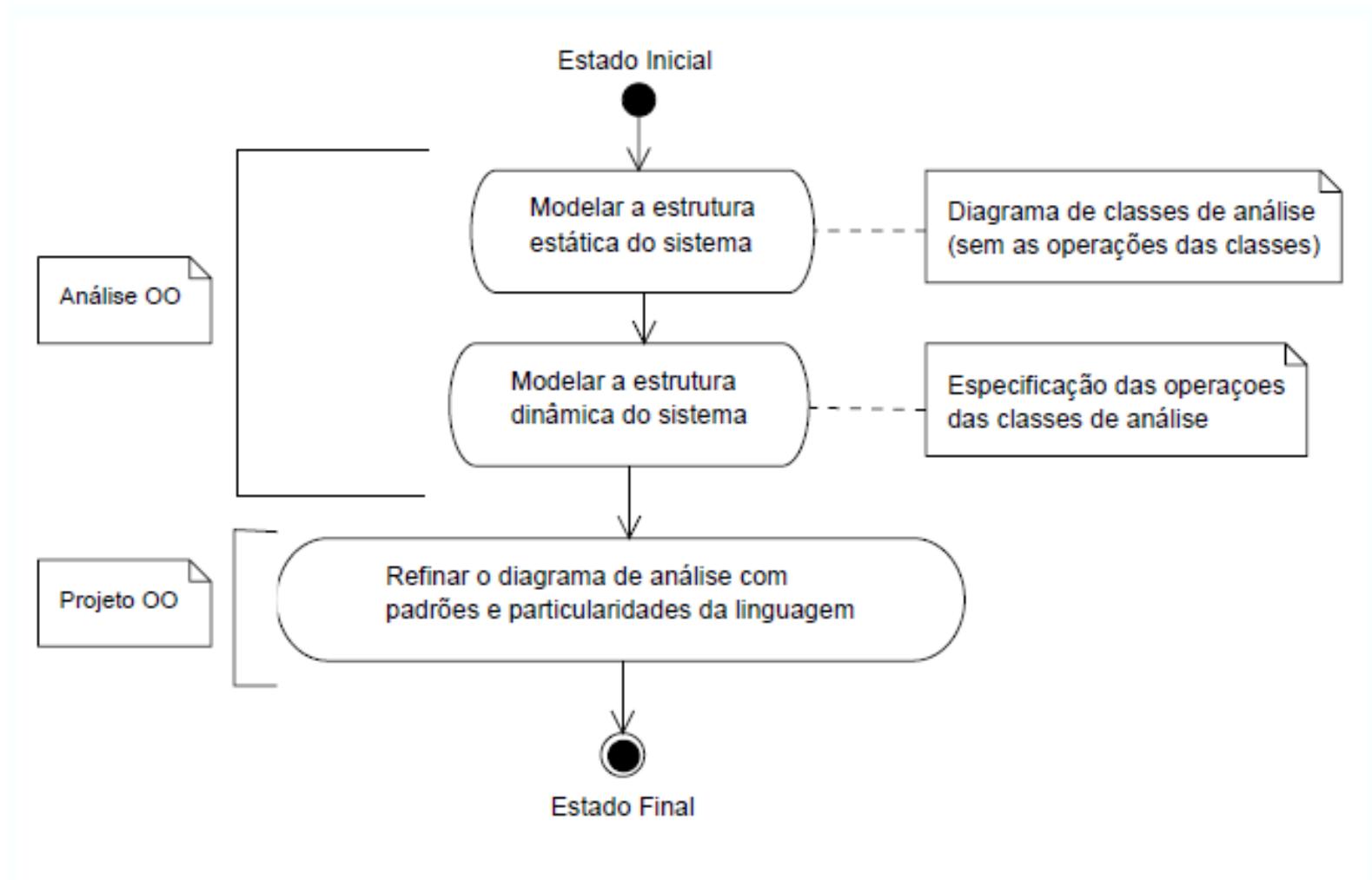
Realizar Saque

Cliente

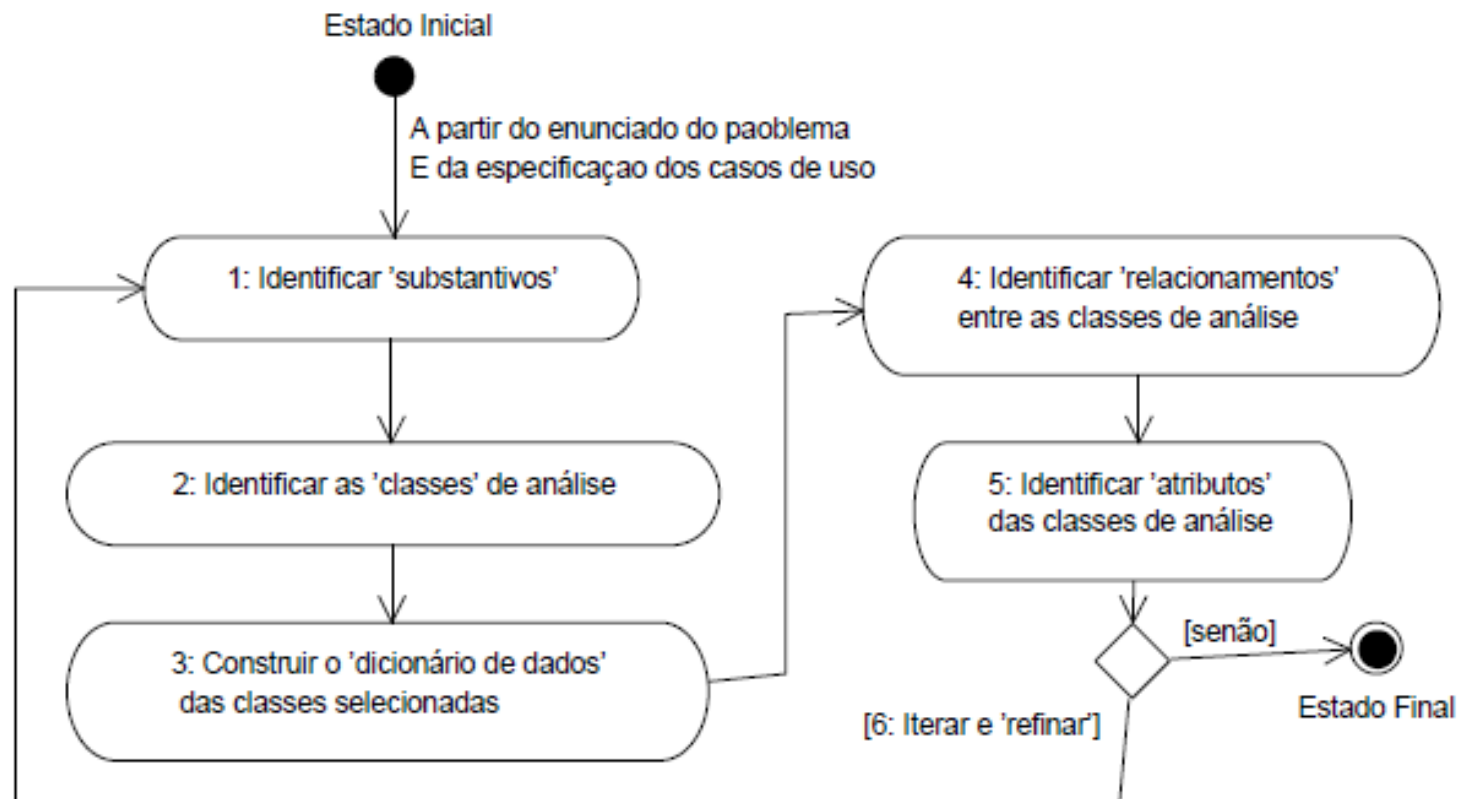
Análise - Artefatos

- **3. CONCRETIZAR A REALIZAÇÃO DE UM *USE-CASE***
 - **fluxo de eventos**
 - Descrição textual do diagrama de colaboração
 - **requisitos especiais**
 - Descrição textual de requisitos não-funcionais
- **4. PACOTES DE ANÁLISE**
 - Devem ter coesão e fraco acoplamento
 - Candidatos a subsistemas do projeto
 - PACOTE DE SERVIÇOS: é um conjunto de ações coerentes, indivisíveis para uso em vários use-cases
- **5. DESCRIÇÃO DA ARQUITETURA**

Uma abordagem para análise OO



Modelagem Estática



Identificando Classes de Análise

- As especificações dos casos de uso fornecem as informações necessárias.
- Primeiro identifica-se os conceitos, dentro do domínio do problema, que são relevantes para o sistema que se pretende construir.
 - Esses conceitos se transformam posteriormente em classes de análise.
 - Em seguida pode-se fazer uma análise textual da descrição do problema e das especificações dos casos de uso para complementar as classes relevantes para o sistema
- Importante: o diagrama de classes de análise é uma descrição de coisas no domínio do problema do mundo real, não no do projeto de software!

Caso de Uso Consultar Saldo (objetos)

Breve Descrição: O cliente, já autenticado, escolhe a opção “Consultar Saldo” e o sistema apresenta o seu saldo.

Atores: Cliente, Cadastro de Contas do Banco.

Pré-condição: A conta deve estar ativa e o cliente já deve ter sido autenticado junto ao sistema, através do caso de uso Efetuar Login.

Pós-condição: Estado da conta inalterado.

Requisitos Especiais: nenhum.

Caso de Uso Consultar Saldo (objetos)

Fluxo Básico :

1. O cliente escolhe no menu principal do terminal a opção “Consultar Saldo”.
2. O sistema verifica se o login foi efetuado
3. O sistema verifica se a conta está ativa, através do Cadastro de Contas do Banco.
4. O sistema obtém o saldo da conta do cliente e o imprime.

Caso de Uso Consultar Saldo (objetos)

Fluxo Alternativo 1:

No passo 2 do Fluxo Básico, se o login não foi efetuado, o sistema informa isso ao cliente.

Fluxo Alternativo 2:

No passo 3 do Fluxo Básico, se a conta não estiver ativa, o sistema informa isso ao cliente e avisa que a consulta não pôde ser realizada.

Caso de Uso: Efetuar Saque

Breve Descrição: O cliente, já autenticado, escolhe a opção “Efetuar Saque”, informa a quantia desejada e, caso o saldo da conta seja suficiente e o caixa tenha o dinheiro necessário, a quantia é liberada.

Atores: Cliente, Cadastro de Contas do Banco

Pré-condição: O cliente deve estar logado no sistema, através do caso de uso Efetuar Login. Além disso, a conta deve estar ativa e o valor a debitar deve ser maior que zero e não pode ser superior ao saldo da conta nem a quantidade de dinheiro disponível no caixa.

Pós-condição: O valor a ser sacado é subtraído do saldo da conta e do total disponível no caixa eletrônico e a quantia solicitada é fornecida ao cliente.

Requisitos Especiais: nenhum.

Fluxo

Caso de Uso: Efetuar Saque

Caso de Uso: Efetuar Saque

Fluxo Básico:

1. O cliente escolhe no menu principal do terminal a opção “Efetuar Saque”.
2. O sistema verifica se o login foi efetuado.
3. O sistema verifica se a conta está ativa, através do Cadastro de Contas do Banco.
4. O sistema solicita que o cliente informe a quantia desejada.
5. O cliente informa a quantia desejada.
6. O sistema verifica se o saldo da conta é suficiente para realizar a transação e, em caso afirmativo, se há dinheiro em quantidade suficiente no caixa.
7. O sistema subtrai o valor solicitado do saldo da conta do cliente e do valor disponível no caixa e libera a quantia solicitada, através do dispensador de notas.

Caso de Uso: Efetuar Saque

Fluxo Alternativo 1:

No passo 2 do Fluxo Básico, se o login não tiver sido efetuado, o sistema informa isso ao cliente.

Fluxo Alternativo 2:

No passo 3 do Fluxo Básico, se a conta não estiver ativa, o sistema avisa isso ao cliente e informa que o saque não pôde ser realizado.

Caso de Uso: Efetuar Saque

Fluxo Alternativo 3 :

No passo 6 do Fluxo Básico, se o valor solicitado for menor que zero ou superior ao saldo da conta ou a quantidade de dinheiro disponível no caixa, o sistema informa que não é possível realizar o saque e o porquê. Em seguida, volta ao passo 4 do Fluxo Básico.

Fluxo Alternativo 4 :

Após o passo 7 do Fluxo Básico, se o saldo da conta for menor ou igual a zero, a conta deve ser desativada.

Fluxo Alternativo 5 :

No passo 5 do Fluxo Básico, o cliente pode cancelar a operação.

Entidades Candidatas Identificadas

- Caso de Uso Consultar Saldo

- Saldo
- Terminal
- Saldo da conta
- Login
- Consulta
- Quantia de dinheiro disponível em caixa

Entidades Candidatas Identificadas

- Caso de Uso Efetuar Saque

- Caixa
- Dinheiro
- Quantia
- Valor a debitar
- Quantia de dinheiro disponível em caixa
- Valor a ser sacado
- Quantia solicitada
- Quantia desejada
- Transação

Entidades Candidatas Identificadas

- Caso de uso Efetuar Login

- Caixa eletrônico
- Cliente
- Número da conta
- Senha
- Acesso
- Sistema
- Cadastro de Contas do Banco

- Opção
- Menu Principal
- Conta
- Banco
- Estado da conta
- Criptografia
- Operação
- Estado do Caixa eletrônico

Entidades Candidatas Identificadas

- Caso de uso Efetuar Depósito

- Valor a depositar
- Valor a ser depositado
- Valor depositado
- Estado da conta
- Quantia informada pelo cliente
- Conta destino do depósito

Refinar a Lista de Classes

- **Classes Redundantes:** quando duas palavras significam a mesma coisa, escolha a palavra mais significativa.
- **Classes Irrelevantes:** aquelas classes que não estão diretamente relacionadas com o problema.
- **Atributos:** alguns atributos podem ser descritos por substantivos.
- **Operações:** alguns substantivos podem ser operações.
- **Papéis:** nomes de papéis são de fato nomes de processos dinâmicos ao invés de classes propriamente ditas.
- **Construções de Implementações:** qualquer coisa que faça referência a estruturas de dados, etc.

Classes Candidatas Eliminadas

- Classes redundantes:
 - Valor a ser depositado, Valor depositado e Quantia informada pelo cliente: equivalentes a Valor a depositar.
 - Caixa: idêntica a classe Caixa eletrônico.
 - Valor a ser sacado, Quantia desejada, Quantia solicitada, Quantia: equivalentes a Valor a debitar.
 - Saldo da conta, Saldo da conta do cliente: equivalentes a Saldo.
 - Operação e Opção: equivalente a Transação.
 - Conta destino do depósito: equivalente a Número da conta.

Classes Candidatas Eliminadas (ii)

- Classes Irrelevantes:
 - Número da conta: atributo da classe Conta.
 - Senha: atributo da classe Conta.
 - Estado do caixa: termo genérico para os atributos da classe Caixa eletrônico.
 - Quantidade de dinheiro disponível no caixa: atributo da classe Caixa eletrônico.
 - Estado da conta: termo genérico para os atributos da classe Conta.
 - Saldo: atributo da classe Conta.

Classes Candidatas Eliminadas (iii)

- Classes Vagas:
 - Acesso
 - Menu Principal
 - Criptografia
 - Login
 - Consulta
 - Valor a debitar
 - Valor a depositar

Lista revisada de classes

- CaixaEletrônico
- Cliente (diferente do ATOR – “dados do cliente”)
- Sistema
- Cadastro de Contas do Banco (ATOR)
- Conta
- Banco
- Terminal
- Transação
- **Atualizar dicionário de dados!**

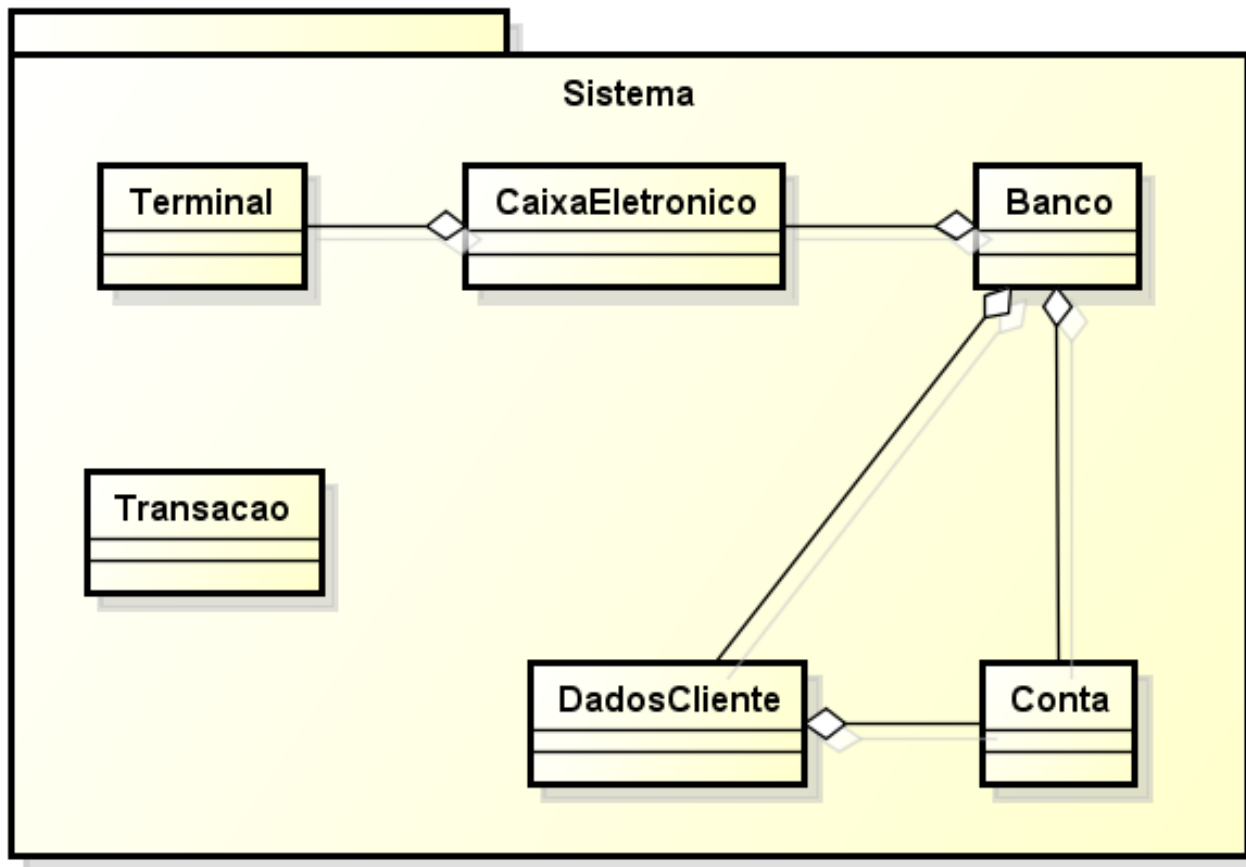
Identificar/Refinar Relacionamentos

- As classes identificadas até o momento devem ser analisadas com o intuito de identificar as associações e os relacionamentos de agregação/decomposição e de generalização/especialização entre elas
- A classe Sistema representa o sistema como um todo e, desta forma, todas as outras classes podem ser consideradas partes dela
- Para simplificar a representação do modelo, a classe Sistema pode ser substituída por um pacote que contenha todas as classes que compõem o sistema

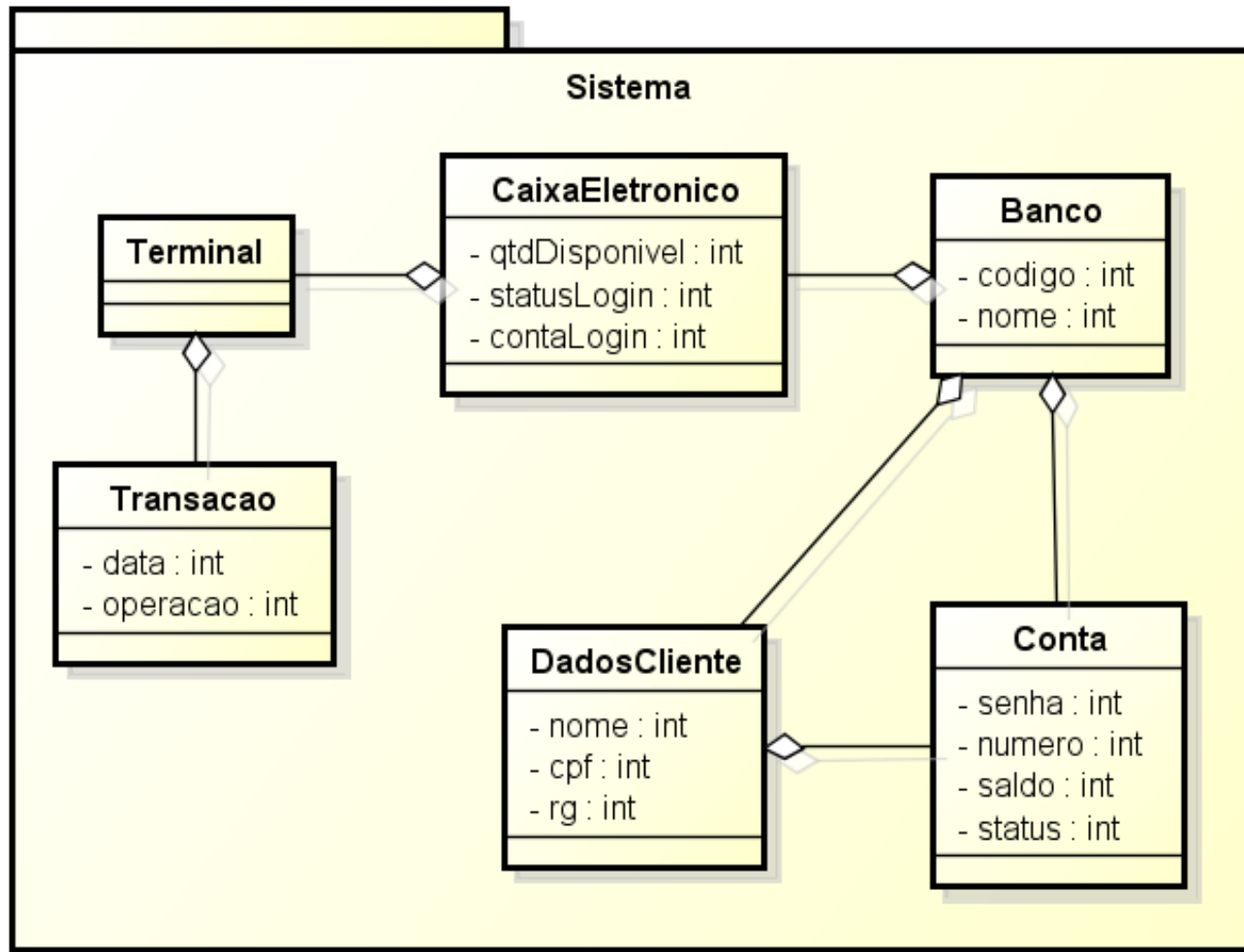
Agregações Encontradas

- Um Banco possui uma ou mais Contas
- Um Banco contém vários clientes (DadosCliente)
- Um Banco possui vários Caixas Eletrônico
- Um Caixa Eletrônico possui um Terminal
- Um cliente (DadosCliente) pode possuir várias Contas
- Em um terminal podem ser realizadas várias transações

Agregações Encontradas



Identificar Atributos



Identificar/Refinar Classes (MVC)

- Classificar as classes em (Fronteira, Controle e Entidade)
- Identificar novas classes (Fronteira e Controle)
- Novas Classes:
 - FronteiraCadastroContas (interagir com o ator Cadastro de contas)
 - ControladorCaixa (controla a lógica interna do caixa eletrônico)

Classes do Sistema

- CaixaEletronico << entity >>
- Conta << entity >>
- Banco << entity >>
- Terminal << boundary >>
- DadosCliente << entity >>
- Transacao << entity>>
- FronteiraCadastroContas << boundary >>
- ControladorCaixa << control >>
- **Atualizar Dicionário de Dados**

Identificar/Refinar relacionamentos

- Adicionar as associações com as novas classes do modelo MVC, obedecendo a relação:

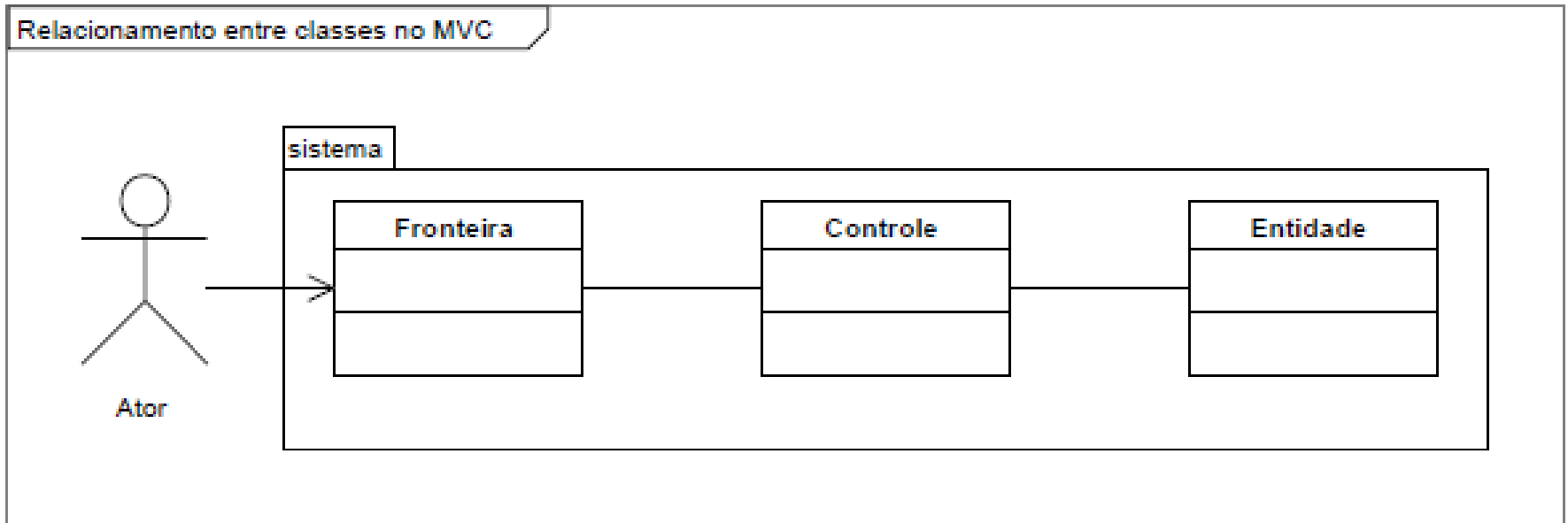
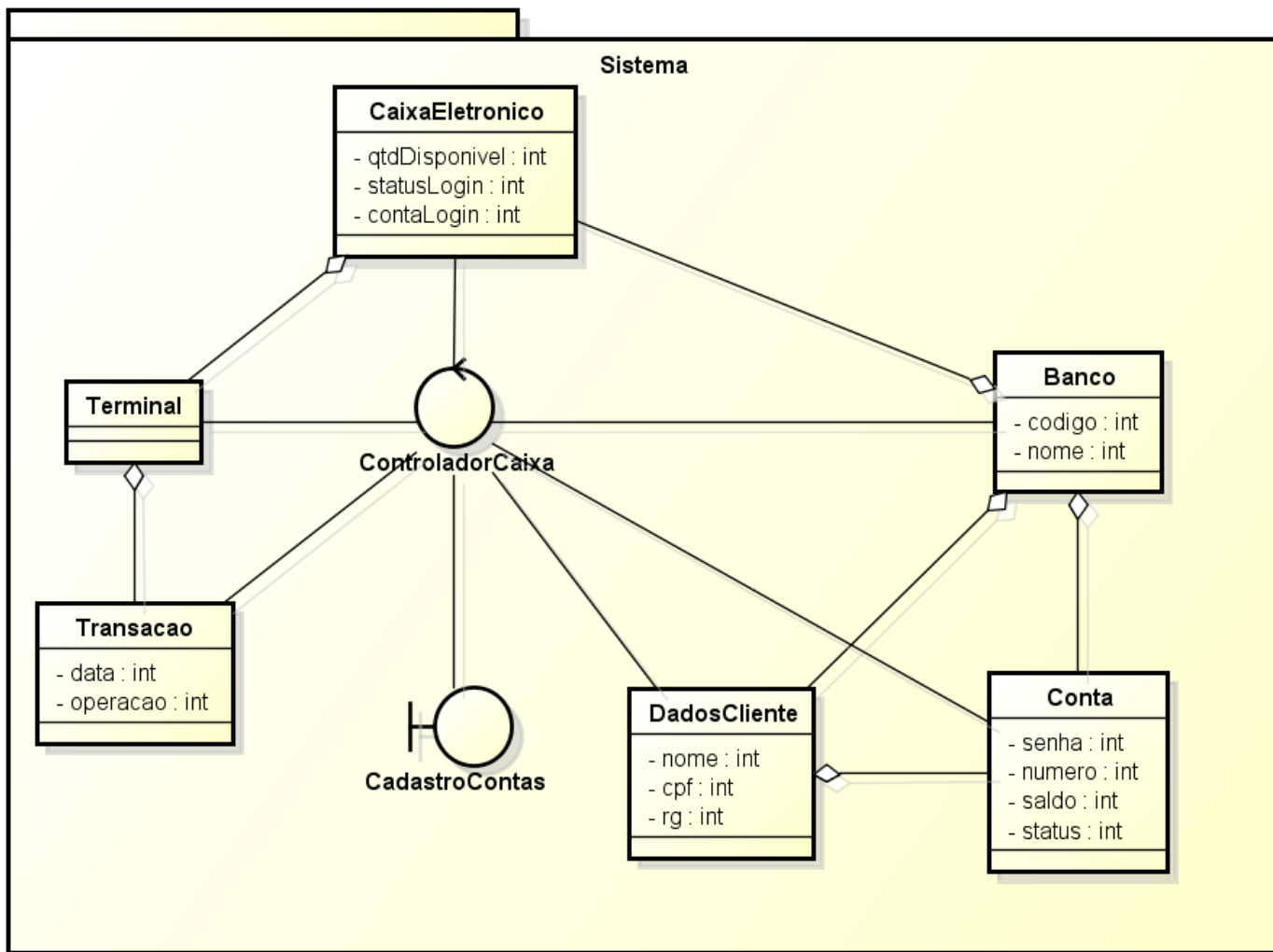
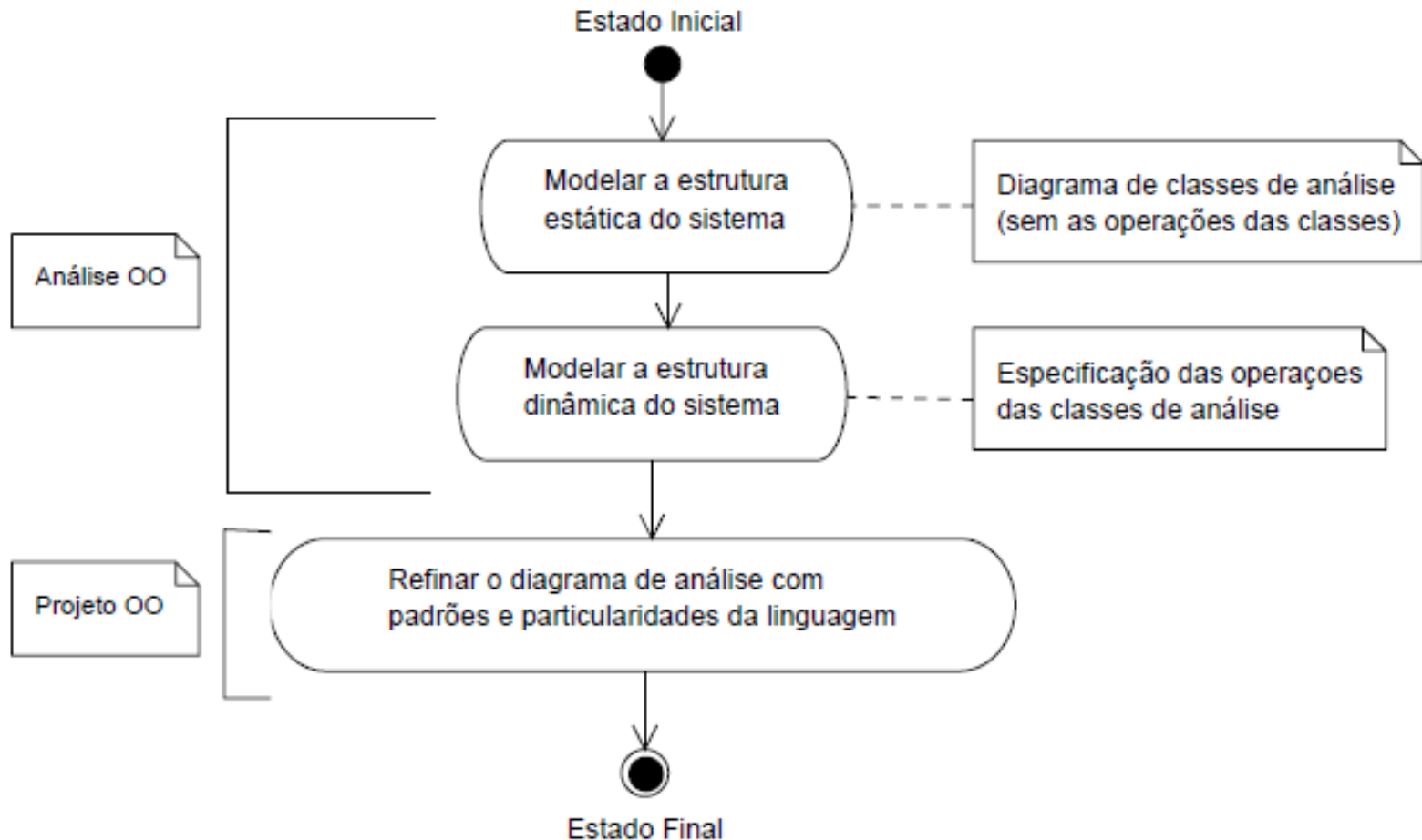


Diagrama de Classes de Análise (sem operações)



Modelagem Dinâmica



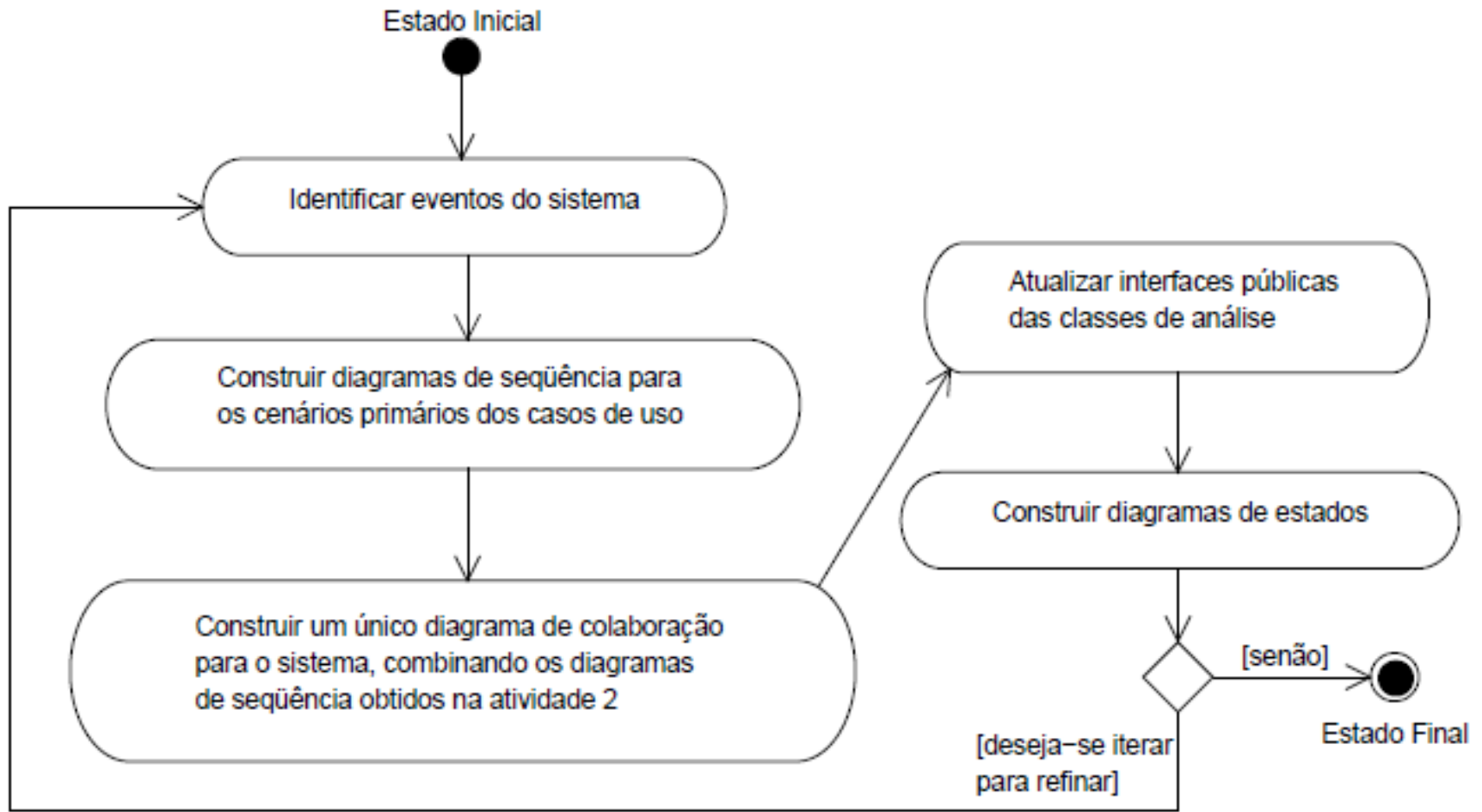
Modelagem Dinâmica

- Identifica e modela os aspectos do sistema de software que podem mudar durante a sua execução, devido a ocorrência de eventos.
- Foco no comportamento que o sistema deve apresentar.
- Usa os diagramas dinâmicos da UML (sequência, colaboração, estados).
- Especifica uma versão inicial das interfaces públicas das classes de análise.
- Sub-etapa de Análise OO - Foco no domínio do problema!

Eventos

- Ocorrências dignas de nota relativas ao sistema e envolvendo algum tipo de troca de informação.
- O evento não é a informação trocada e sim o fato de alguma informação ter sido trocada.
- O tipo de evento mais comum encontrado durante a análise é a interação entre um ator e o sistema.
 - Outros tipos também são possíveis.
- Modelamos o comportamento do sistema através de eventos e das ações executadas em resposta a eles.

Atividades Modelagem Dinâmica



Identificar Eventos do Sistema

- Deve ser realizada uma nova análise textual nas especificações dos casos de uso, prestando-se atenção aos pontos nos quais trocas de informação ocorrem.
- Normalmente, esses pontos estão associados a verbos.
- Informações relevantes: verbos e os contextos nos quais aparecem.

Caso de uso Efetuar Login (i)

Caso de Uso: Efetuar Login

Fluxo Básico:

1. O cliente solicita a opção de “Efetuar Login” no sistema.
2. O sistema pede que o cliente informe o número da conta.
3. O cliente fornece o número da conta.
4. O sistema pede que o cliente informe a sua senha.
5. O cliente fornece a senha.
6. O sistema verifica se a conta é válida e se a senha está correta, através do Cadastro de Contas do Banco. Em caso positivo, o sistema atualiza o estado do caixa eletrônico com as informações de login.
7. O sistema exibe no terminal o menu de opções que o cliente pode acessar.

Caso de uso Efetuar Login (ii)

Fluxo Alternativo 1:

No passo 6 do Fluxo Básico, se a conta fornecida não existir ou se a senha estiver errada, o sistema informa que alguma das informações fornecidas está incorreta e que não é possível autenticar o cliente. Em seguida, volta ao passo 2 do Fluxo Básico.

Fluxo Alternativo 2:

Nos passos 3 e 5 do Fluxo Básico, o cliente pode cancelar a operação.

Eventos Identificados

- De responsabilidade do Sistema
 - Verificar se a conta é válida.
 - Verificar se a senha está correta.
 - Atualizar o estado do caixa eletrônico com as informações de login
 - Verificar se o login foi efetuado.
 - Verificar se a conta está ativa.
 - Obter o saldo da conta.
 - Verificar se o cliente tem saldo suficiente para realizar a transação.

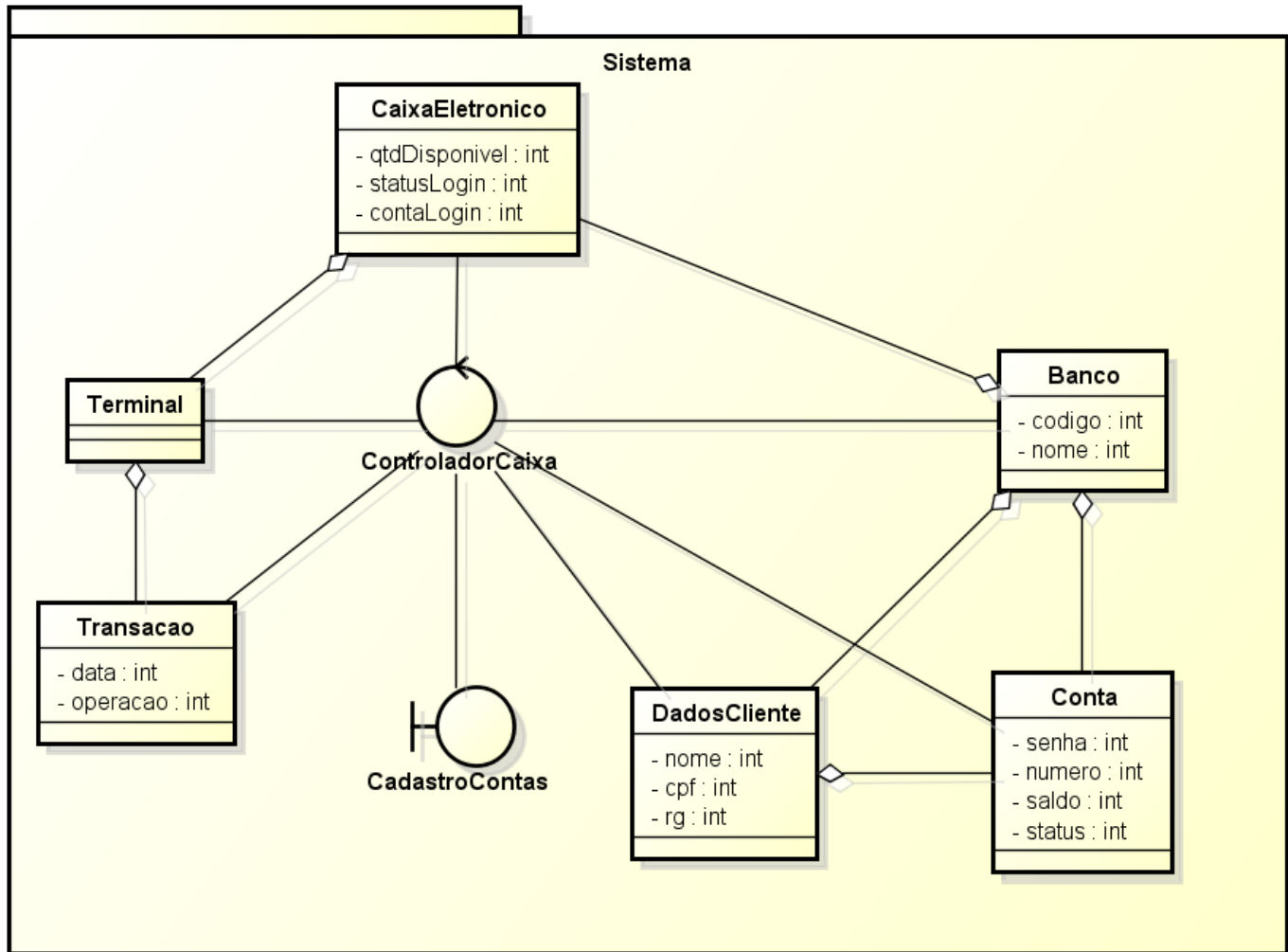
Eventos Identificados

- De responsabilidade do Sistema
 - Verificar se há dinheiro em quantidade suficiente no caixa.
 - Subtrair o valor solicitado do saldo da conta do cliente.
 - Desativar a conta.
 - Adicionar o valor depositado ao saldo da conta.
 - Verificar se a conta deve ser reativada.
 - Reativar a conta.

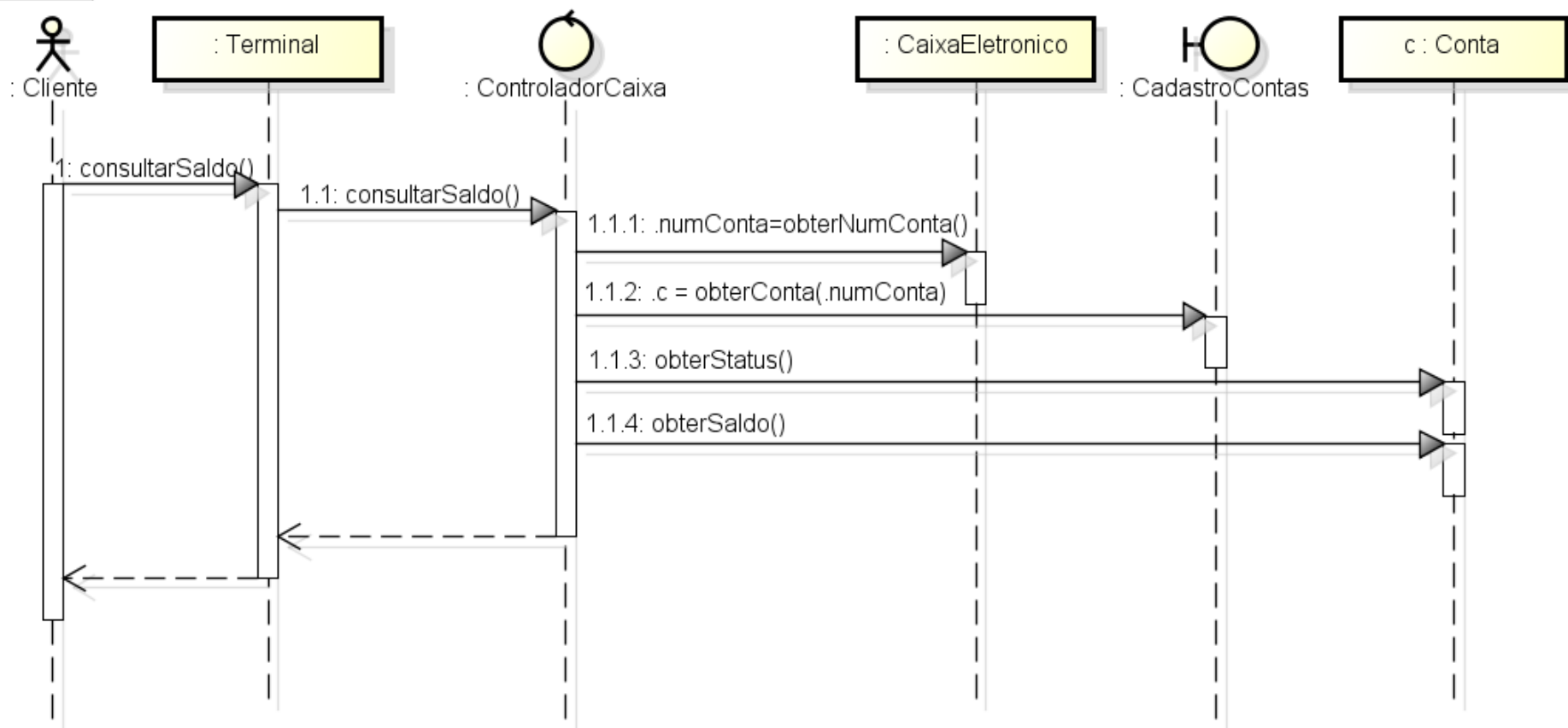
Construir Diagramas de Sequência

- Baseado nos eventos encontrados. Cada evento pode corresponder a um ou mais fluxos no diagrama de sequência.
- Deve-se ter em mente as classes descobertas na análise estática, pois é a partir da interação dos seus objetos que as funcionalidades são implementadas

Diagrama de Classes de Análise



Sequência – Consultar Saldo



Sequencia – Efetuar Saque

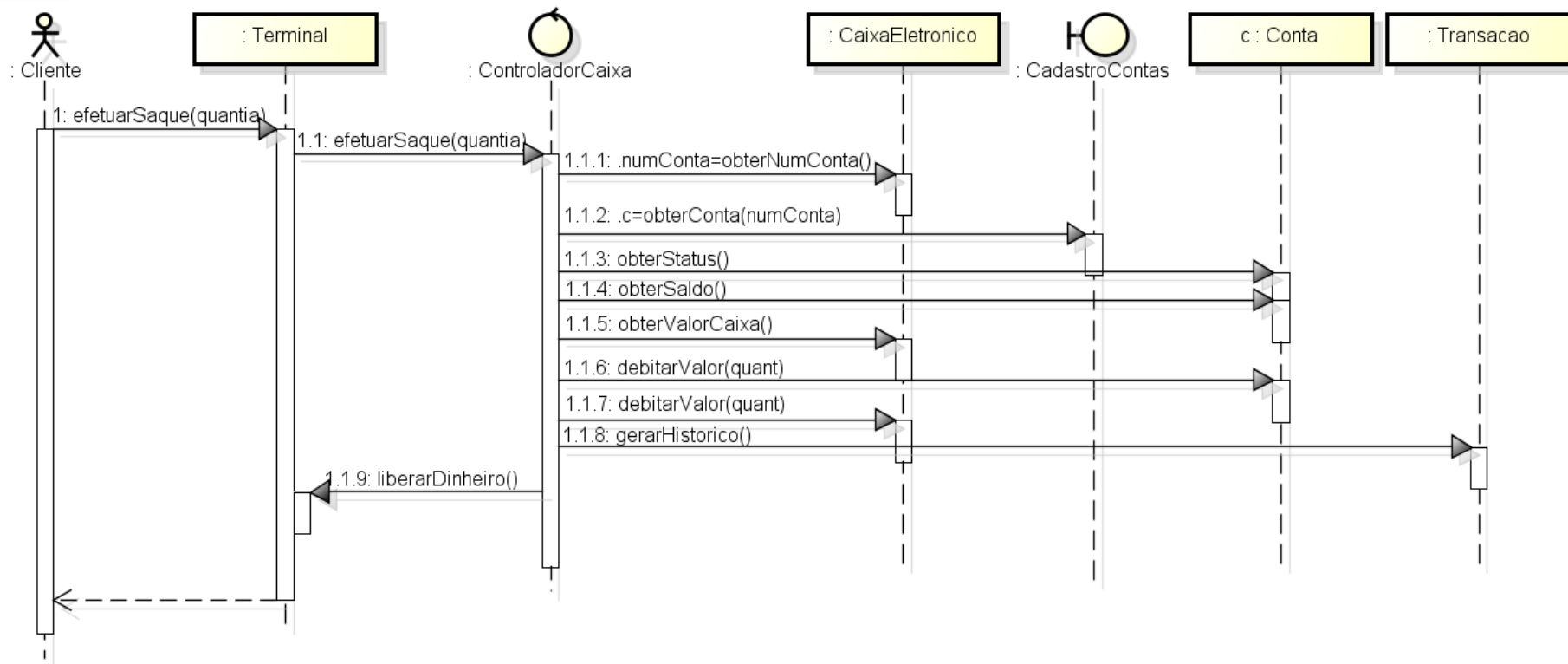
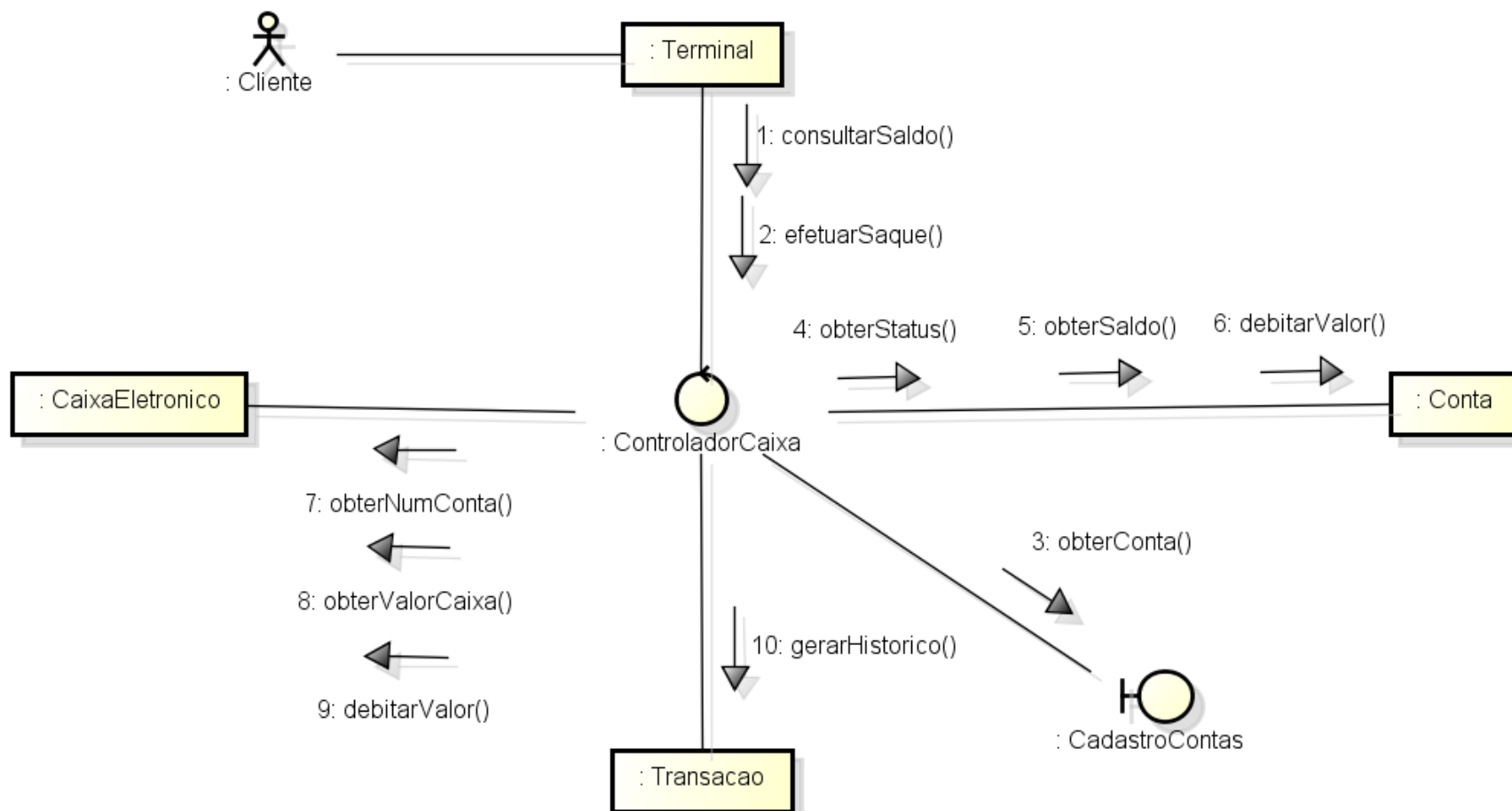


Diagrama de Comunicação

- Centraliza a representação dos eventos dos diagramas de sequência
- Explicita as associações entre as classes e facilita a identificação das operações

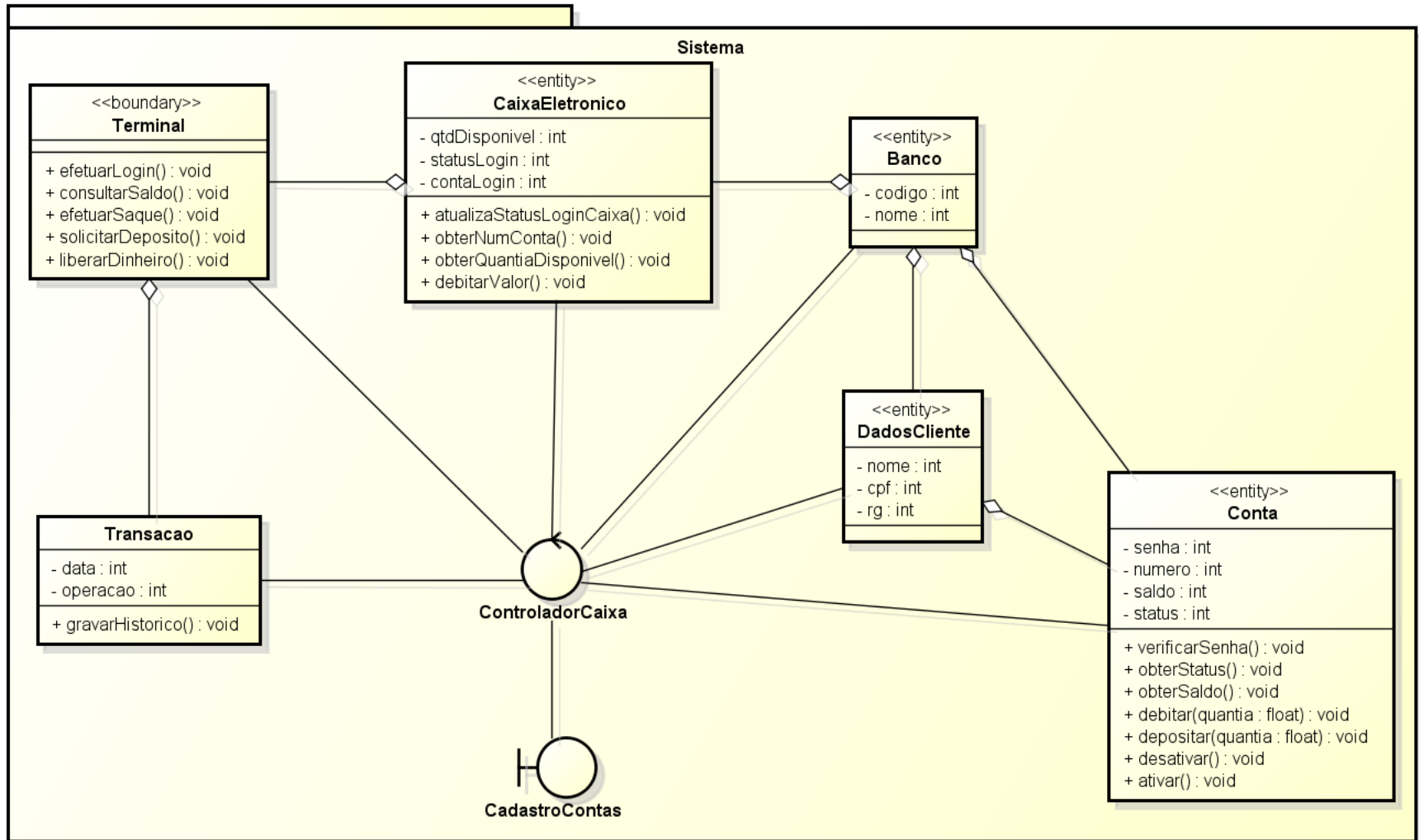
Comunicação Saque/Consulta



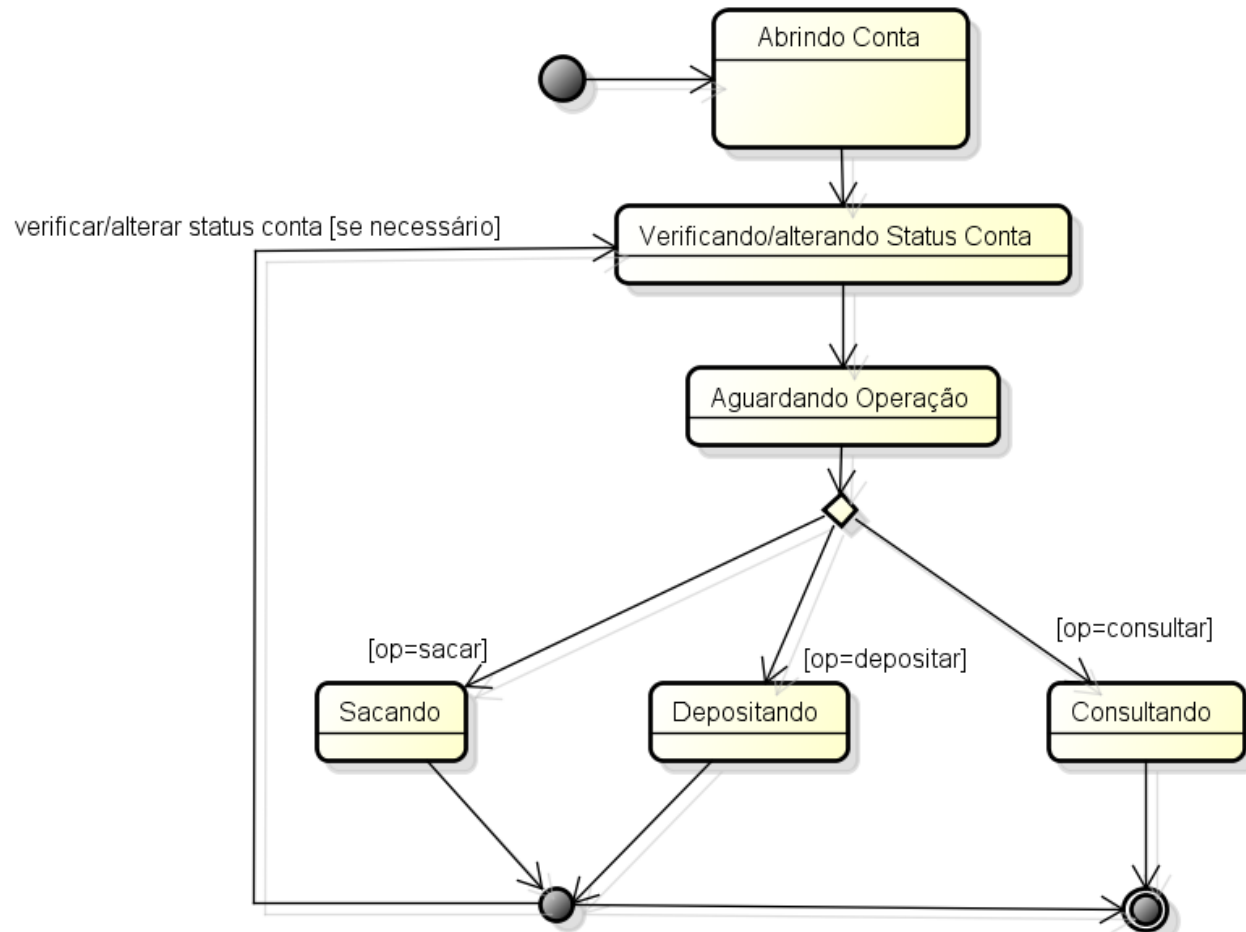
Identificação das Operações

- Cada evento recebido pode ser
 - Uma operação que a classe deve oferecer
 - O retorno de uma operação executada

Diagrama de Classes final de Análise



Estados da Classe Conta



Elaboração -> Projeto

Projeto

- Adquirir uma compreensão de aspectos de requisitos não funcionais e restrições sobre *linguagens de programação, sistemas operacionais, SGBDs, aspectos de distribuição, etc.*
- Criar informações suficientes para a implementação, descrevendo *subsistemas, interfaces e classes*.
- Estar apto a dividir a tarefa de implementação em equipes
- Determinar mais cedo as interfaces entre os subsistemas
- Criar um modelo que possibilite uma implementação que preencha as estruturas definidas sem altera-las

Projeto

MODELO DE ANÁLISE

conceitual

Genérico (c.r. projeto)

3 tipos de classes

Menos formal

Mais rápido (1/5 do projeto)

Poucos níveis

Menos dinâmica

Não se mantém no ciclo

MODELO DE PROJETO

físico

específico

Depende da implementação

Mais formal

Mais demorado (5 x análise)

Muitos níveis

Mais dinâmica, foco na
sequencia

Se mantém em todo ciclo

Projeto - Artefatos

- **1. Modelo de Projeto**
 - hierarquia de **subsistemas** contendo **classe de projeto**, **projetos de *use-cases*** e **interfaces**
- **2. Classes de Projeto**
 - na linguagem de programação da implementação
 - visibilidade dos atributos (ex. publico, protegido, privado)
 - generalizações e herança;
 - associações e agregações e atributos
 - métodos em pseudo-código

Projeto - Artefatos

- **3. Realização dos Casos de Uso**
 - Diagrama de classes
 - Diagrama de interações (diagramas de sequência)
 - Fluxo de eventos (textual)
 - Requisitos de implementação

Projeto - Artefatos

- 4. Subsistema de Projeto
 - (pacotes de análise, componentes, produtos de software, sistemas existentes) - SUBSISTEMAS DE SERVIÇO
- 5. Interface (separa funcionalidade de implementação)
- 6. Arquitetura (VISÃO DO PROJETO)
 - 1. Subsistemas, interfaces e dependências
 - 2. Classes chave, classes ativas
 - 3. Realização de *use-cases* centrais ao sistema
- 7. Modelo de Distribuição
(Diagrama de componentes)
- 8. Arquitetura (VISÃO DO MODELO DE DISTRIBUIÇÃO)
(Diagrama de Implantação)

Projeto - Arquitetura

A1) Identificar nós e suas configurações

- determinar os nós envolvidos e suas características
- determinar os tipos de conexões entre os nós
- verificar necessidades de processamentos redundantes, backups, etc.

A2) Identificar subsistemas e suas interfaces

- subsistemas da aplicação
- identificar middleware (SO, SGBD, software de comunicação, pacotes GUI, distribuição, etc.)
- definir dependências entre os subsistemas
- identificar as interfaces entre os subsistemas

Projeto - Arquitetura

A3) Identificar classes de projeto significativas

- a partir das classes de análise
- classes ativas (requisitos de concorrência, performance, inicialização, distribuição, prevenção de deadlocks)

A4) outros requisitos de projeto

- (persistência, transparência de distribuição, segurança, recuperação de erros, gerência de transações)

Projeto - Classe

A1) Definir uma classe de projeto

- a partir de classes de fronteira : depende da linguagem
- classes de entidades persistentes podem produzir tabelas relacionais
- classes de controle podem gerar várias classes de projeto (distribuição) ou serem encapsuladas em classes de entidades

A2) Definir operações

- realizar as responsabilidades da classe
- requisitos especiais (e.g. acesso ao banco de dados)
- atender às necessidades das interfaces da classe

A3) Definir atributos

- considerar os atributos da análise
- os tipos dos atributos são determinados pela linguagem de programação
valores de atributos usados por vários objetos devem ser transformados em objetos

Projeto - Classe

A4) Identificar associações e agregações

- dependendo da linguagem, transformá-los em relacionamentos
- tentar transformar cardinalidades, papéis, etc. em atributos ou em novas classes para realizar a associação
- analise a navegabilidade pelas associações

A5) Identificar generalizações

A6) Descrever métodos

- realização de operações por pseudo-código, diagramas de atividades, linguagem natural,...

A7) Descrever estados

- diagrama de estados

Projeto - Subsistema

1. Rever as dependências entre subsistemas
2. Rever as interfaces
3. Rever o conteúdo

Elaboração -> Implementação

Implementação

1. **MODELO DA IMPLEMENTAÇÃO**
2. **COMPONENTE**
3. **SUBSISTEMA DE IMPLEMENTAÇÃO**
4. **INTERFACE**
5. **ARQUITETURA (visão da implementação)**
6. **PLANO DE INTEGRAÇÃO**

Implementação

- MODELO DA IMPLEMENTAÇÃO

- É uma hierarquia de **subsistemas de implementação** contendo **componentes** e **interfaces**

- COMPONENTE

- É UM PACOTE CONTENDO ELEMENTOS DO PROJETO

- Diagrama de Componentes

- <<executable>> (programa executável)

- <<file>> (arquivo contendo código fonte ou dados)

- <<library>> (biblioteca estática ou dinâmica)

- <<table>> (tabela do banco de dados)

- <<document>> (um documento)

Implementação

- SUBSISTEMAS DE IMPLEMENTAÇÃO

 - um package em Java

 - um project em Visual Basic

 - um diretório de C++

- INTERFACES

 - Implementam as interfaces do projeto

- ARQUITETURA (visão da implementação)

 - Decomposição em subsistemas, compostos de interfaces e componentes e Componentes chave

- PLANO DE INTEGRAÇÃO

 - Primeira versão executável: testes localizados de integração para facilitar a detecção de erros: => versão final

Elaboração -> Teste

Teste

- Planejar os testes em cada iteração, tanto os **testes de integração** quanto os **testes de sistema**
- preparar casos de teste, criar procedimentos de teste e procedimentos executáveis
- Realizar os testes e analisar os resultados

Teste - Artefatos

- Modelo de Teste
- Casos de Teste

Ciclo de Vida PU

