

INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO
GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS



Orientação a Objetos com UML e
Modelagem de Dados - TP1

ALUNO: MAGNO VALDETARO DE OLIVEIRA
E-MAIL: mvaldetaro@gmail.com
TURMA: NOITE - LIVE
MATRÍCULA: 10403782775

Introdução

Este documento apresenta a modelagem inicial de um sistema bancário que oferece operações simples para seus usuários. Detalhamos seus casos de uso, cenários e apresentamos os diagramas relacionados.

Justificativa

Este trabalho nos permite desenvolver o raciocínio para desenvolvimento de um sistema e aplicar os conhecimentos adquiridos sobre as visões de sistema, orientação a objeto e UML.

Descrição

Documentação que represente a modelagem inicial de um sistema bancário, capaz de oferecer algumas operações simples para seus usuários. Como exemplo, esses usuários podem ser os clientes, através de um caixa eletrônico ou pela Internet, ou um gerente, através de uma Intranet. A documentação será composta por elementos textuais e gráficos.

Objetivo

- Compreender os conceitos de classes, atributos e métodos
- Compreender os conceitos de instanciação e interação entre objetos
- Compreender o conceito de encapsulamento
- Compreender o conceito de herança
- Compreender o conceito de polimorfismo
- Compreender as visões de um sistema
- Conhecer os principais diagramas da UML
- Conhecer outros diagramas da UML
- Descrever casos de uso
- Identificar e desenhar casos de uso
- Identificar e desenhar atores
- Identificar e desenhar relacionamentos

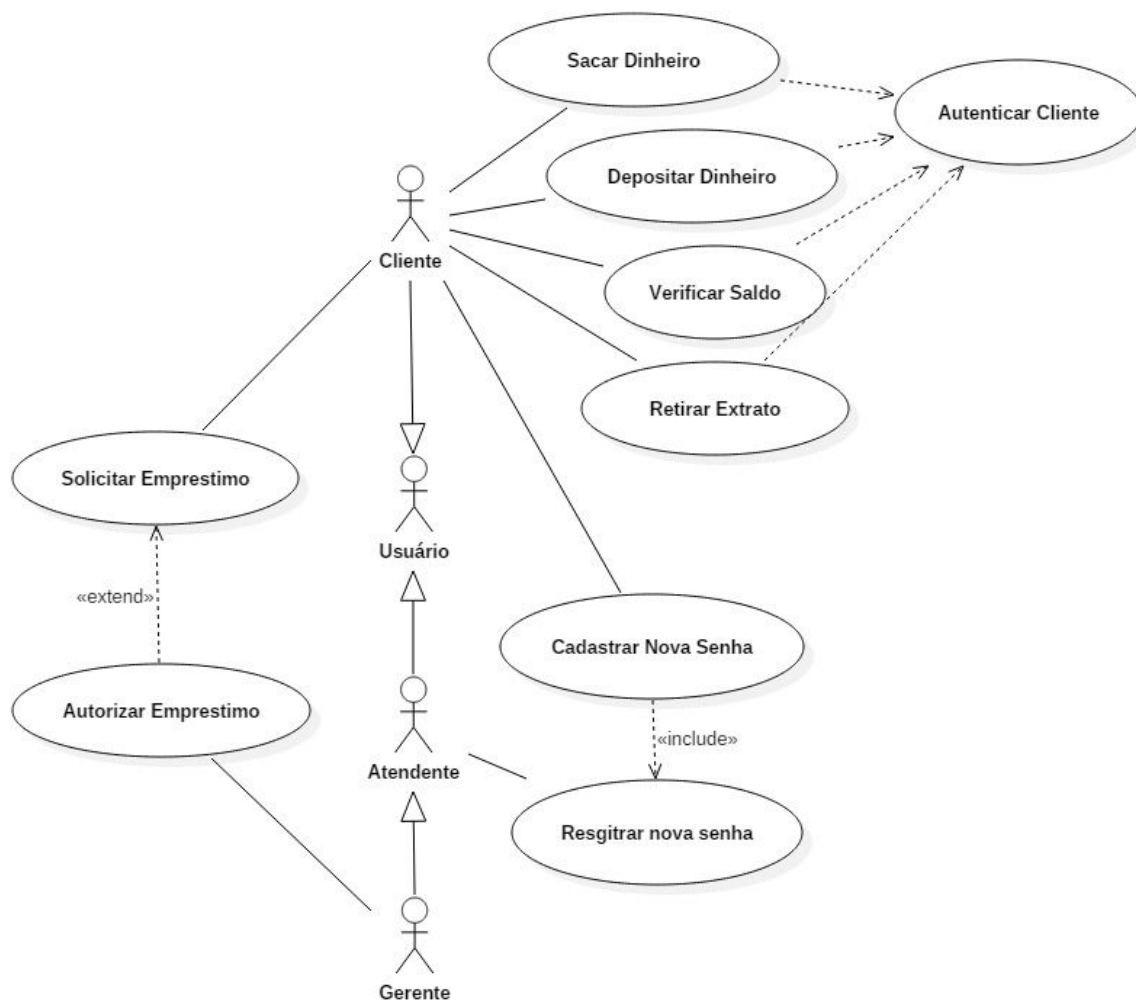
Documento de Projeto de Sistema

Projeto: Sistema bancário

Versão: 1.0

Casos de uso

Diagrama:



Caso - Sacar Dinheiro

Descrição:

Este caso de uso permite que o cliente acesse o sistema e faça retirada de uma quantia informada.

Atores:

Cliente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de crédito e senha.

Pós-condições:

Ter realizado a retirada da quantia informada.

Fluxo de Eventos**Cenário 1**

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita a realização do saque e informa a quantia;
6. O sistema requisita a quantia que será sacada;
7. O cliente retira a quantia.

Cenário 2

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita a realização do saque e informa a quantia;
6. O sistema informa que não há a quantia disponível na conta para saque;
7. O cliente retira a quantia.

Caso - Depositar Dinheiro**Descrição:**

Este caso de uso permite que o cliente acesse o sistema e faça depósito de uma quantia informada.

Atores:

Cliente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de crédito e senha.

Pós-condições:

Ter realizado o depósito da quantia informada e receber o comprovante.

Fluxo de Eventos**Cenário 1**

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita a realização de depósito;
6. O sistema solicita que informe a quantia que será depositada;
7. O cliente informa a quantia.
8. O sistema apresenta as instruções para o depósito;
9. O cliente confirma que seguiu as instruções;
10. O sistema emite o comprovante da transação.

Cenário 2

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita a realização de depósito;
6. O sistema solicita que informe a quantia que será depositada;
7. O cliente informa a quantia.
8. Se o valor for acima de R\$5,000, o sistema apresenta a instrução: “Prezado cliente, este depósito ultrapassa o valor limite por operação, procure um de nosso atendentes.”;

Caso - Verificar Saldo**Descrição:**

Este caso de uso permite que o cliente acesse o sistema e verifique o saldo de sua conta.

Atores:

Cliente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de crédito e senha.

Pós-condições:

Ter realizado visualizado o saldo da conta.

Fluxo de Eventos**Cenário 1**

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita o saldo da conta;
6. Sendo o saldo positivo, o sistema apresenta o saldo;

Cenário 2

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita o saldo da conta;
6. Sendo o saldo negativo o sistema apresenta o saldo e a opção “Solicitar empréstimo”;

Caso - Retirar Extrato**Descrição:**

Este caso de uso permite que o cliente acesse o sistema e possa verificar seu extrato.

Atores:

Cliente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de credito e senha.

Pós-condições:

Ter sido apresentado ao extrato da conta.

Fluxo de Eventos**Cenário 1**

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita o extrato da conta;
6. O sistema apresenta o saldo duas opções: "Exibir extrato na tela", "Imprimir Extrato";
7. O cliente seleciona a opção: "Exibir extrato na tela";
8. O sistema apresenta o extrato;

Cenário 2

1. O cliente insere o cartão no caixa eletrônico;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O sistema exibe menu de operações disponíveis;
5. O cliente solicita o extrato da conta;
6. O sistema apresenta o saldo duas opções: "Exibir extrato na tela", "Imprimir Extrato";
7. O cliente seleciona a opção: "Imprimir Extrato";
8. O sistema realiza a impressão do extrato;

Caso - Cadastrar Nova Senha

Descrição:

Este caso de uso permite que o cliente com auxílio de um atendente, atualize sua senha de acesso.

Atores:

Cliente, Atendente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de crédito e senha.

Pós-condições:

Ter sua senha de acesso atualizada.

Fluxo de Eventos

Cenário 1

1. O cliente insere o cartão no terminal de atendimento;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O atendente solicita ao sistema atualização da senha;

5. O sistema solicita a nova senha;
6. O cliente informa a nova senha;
7. O sistema solicita a confirmação da senha;
8. O cliente informa novamente a nova senha;
9. Se as duas senhas digitadas pelo cliente sejam iguais, o sistema solicita confirmação do atendente;
10. O sistema registra a nova senha.

Cenário 2

1. O cliente insere o cartão no terminal de atendimento;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O atendente solicita ao sistema atualização da senha;
5. O sistema solicita a nova senha;
6. O cliente informa a nova senha;
7. O sistema solicita a confirmação da senha;
8. O cliente informa novamente a nova senha;
9. Se as duas senhas digitadas pelo cliente sejam diferentes, o sistema apresenta que as senhas são diferentes e reinicia a operação.

Caso - Solicitar Empréstimo

Descrição:

Este caso de uso permite que o cliente solicite e receba ou não um empréstimo.

Atores:

Cliente, Gerente

Pré-condições:

Ter se autenticado no sistema por meio do cartão de crédito e senha.

Pós-condições:

Receber empréstimo mediante autorização do gerente.

Fluxo de Eventos

Cenário 1

1. O cliente insere o cartão no terminal de atendimento;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O atendente solicita ao sistema atualização da senha;

5. O cliente solicita o empréstimo;
6. O sistema requisita a quantia;
7. O cliente informa a quantia
8. O sistema apresentada que o pedido será avaliado por um gerente;
9. O gerente avalia o pedido de empréstimo e libera a quantia;
10. O cliente recebe a quantia na sua conta.

Cenário 2

1. O cliente insere o cartão no terminal de atendimento;
2. O sistema apresenta a solicitação de senha;
3. O cliente digita a senha;
4. O atendente solicita ao sistema atualização da senha;
5. O cliente solicita o empréstimo;
6. O sistema requisita a quantia;
7. O cliente informa a quantia
8. O sistema apresentada que o pedido será avaliado por um gerente;
9. O gerente avalia o pedido de empréstimo e rejeita o pedido;
10. O cliente recebe uma email ou SMS avisando que seu pedido foi rejeitado.

Classes

As classes que identificamos neste sistema são:

Classe - Usuário

Características	id, nome, sobrenome, cpf, genero, dataNascimento, endereco, telefone
Comportamentos	acessarSistema()
Descrição	<p>Esta classe encapsula todas as principais características dos usuários do sistema e recebe uma instância da classe Endereco por agregação. Todas as características são protegidas e comportamentos são públicos.</p> <p>O comportamento “-acessarSistema” apenas retorna o valor booleano “false”.</p>

Classe - Endereco

Características	idUsuario, rua, numero, bairro, cidade, estado, cep
Comportamentos	cadastrarEndereco(), retornarEndereco(), excluirEndereco()
Descrição	<p>Esta classe encapsula todas as características que compõem os endereços dos usuarios do sistema. Todos os seus atributos são privados.</p> <p>O comportamento “cadastrarEndereco()” recebe os valores informados para os atributos e os carrega no sistema, este comportamento não tem retorno.</p> <p>O comportamento “retornarEndereco()” requisita os dados do sistema e retorna um tipo String com o endereço completo do usuário.</p> <p>O comportamento “excluirEndereco()” remove do sistema todos os dados de endereço do usuário identificado pelo atributo “idUsuario”, este comportamento não tem retorno.</p>
Exemplos de Objetos	<p>objeto 1: idUsuario: Usuario01.id rua: Rua Madalena Casagrande numero: 1652 bairro: Nova Criciúma cidade: Criciúma estado: SC cep: 88801340</p> <p>objeto 2: idUsuario: Usuario02.id rua: Rua Doutor Sylvio Carvalhaes numero: 855 bairro: Jardins cidade: Campinas estado: SP cep: 13060040</p> <p>objeto 3: idUsuario: Usuario03.id rua: Rua Rio Tocantins numero: 1782 bairro: Urucum cidade: Pindamonhangaba estado: SP cep: 12443550</p>

Classe - Cliente

Características	tipo, agencia, conta
Comportamentos	acessarSistema(), criar(), alterar(), excluir()
Descrição	<p>Esta classe herda as características e métodos da classe Usuário. Suas características e métodos são privados</p> <p>O comportamento “-acessarSistema” verifica se o valor passado para os atributos “tipo” e “agencia” são válidos no sistema e retorna o valor booleano “true” se forem válidos, permitindo o acesso do usuário ao sistema e “false” caso não sejam válidos impedindo o acesso ao sistema.</p> <p>O comportamento criar() permite incluir um novo cliente no sistema.</p> <p>O comportamento alterar() permite alterar os dados de um cliente no sistema.</p> <p>O comportamento excluir() permite excluir todos os dados de registro um cliente no sistema.</p>
Exemplos de Objetos	<p>objeto 1: id: 00001 nome: “Maria João” sobrenome: “da Silva” cpf: 38846007379 genero: “feminino” dataNascimento: “01/04/1976” Endereco: “Rua Madalena Casagrande, 1652, Criciúma, SC, 88801340” telefone: 4858637107 tipo: “fisica” agencia: 8888 conta: 777777</p> <p>objeto 2: id: 00002 nome: “Raissa” sobrenome: “Souza Pinto” cpf: 79154400201 genero: “feminino” dataNascimento: “07/04/1987” Endereco: “Rua Doutor Sylvio Carvalhaes, 855, Campinas, SP, 13060040” telefone: 1939053701 tipo: “fisica” agencia: 7788 conta: 888888</p>

	objeto 3: id: 00003 nome: "Luan" sobrenome: "Souza Costa" cpf: 16797776961 genero: "feminino" dataNascimento: "07/04/1987" Endereco: "Rua Rio Tocantins, 1782, Pindamonhangaba, SP, 12443550" telefone: 1939053701 tipo: "fisica" agencia: 1781 conta: 818283
--	---

Classe - Atendente

Características	idAtendente, matricula, senha, agencia
Comportamentos	acessarSistema(), criar(), alterar(), excluir()
Descrição	<p>Esta classe herda as características e métodos da classe Usuário. Suas características e métodos são privados</p> <p>O comportamento "acessarSistema()" verifica se o valor passado para os atributos "matricula" e "senha" são válidos no sistema e retorna o valor booleano "true" se forem válidos, permitindo o acesso do usuário ao sistema e "false" caso não sejam válidos impedindo o acesso ao sistema.</p> <p>O comportamento criar() permite incluir um novo atendente no sistema.</p> <p>O comportamento alterar() permite alterar os dados de um atendente no sistema.</p> <p>O comportamento excluir() permite excluir todos os dados de registro um atendente no sistema.</p>
Exemplos de Objetos	objeto 1: id: 90001 nome: "Ana Maria" sobrenome: "Marinho" cpf: 38846007379 genero: "feminino" dataNascimento: "01/04/1976" Endereco: "Rua Madalena Casagrande, 1652, Criciúma, SC, 88801340"

	<p>telefone: 4858637107 matricula: 0123456789 senha: a1s2d3f4 agencia: 1234</p> <p>objeto 2: id: 90002 nome: "Marcia" sobrenome: "Souza Texeira" cpf: 79154400201 genero: "feminino" dataNascimento: "07/04/1987" Endereco: "Rua Doutor Sylvio Carvalhaes, 855, Campinas, SP, 13060040" telefone: 1939053701 matricula: 08978675 senha: a1s2d3f4 agencia: 1234</p> <p>objeto 3: id: 90003 nome: "Reginaldo" sobrenome: "Costa Souza" cpf: 16797776961 genero: "feminino" dataNascimento: "07/04/1987" Endereco: "Rua Rio Tocantins, 1782, Pindamonhangaba, SP, 12443550" telefone: 1939053701 matricula: 13467980 senha: q1w2e3r4 agencia: 1234</p>
--	---

Classe - Conta

Características	agencia, conta, tipo, saldo
Comportamentos	verificarSaldo(), sacar(), depositar(), transferir(), retirarExtrato()
Descrição	<p>Esta classe esta associada a classe Cliente, todos os objetos do tipo Cliente pode acessar seus métodos e atribuir e requisitar valores de seus atributos.</p> <p>O comportamento "verificarSaldo()" é executado assim que o objeto cliente é criado no sistema. Requisita ao sistema o valor do saldo e atribuí este valor a característica "saldo" para que o resultado seja utilizado por outros métodos. Toda execução de outros</p>

	<p>comportamentos desta classe, executa este comportamento ao final.</p> <p>O comportamento “sacar()” recebe um valor informado pelo usuário e requisita ao sistema que este valor seja reduzido do saldo, retorna “true” e o valor que deve ser reduzido do saldo, caso o valor seja superior ao saldo, este método retorna “false”.</p> <p>O comportamento “depositar()” recebe um valor informado pelo usuário e requisita ao sistema que este valor seja adicionado ao saldo e retorna true. Caso o valor seja acima de R\$5000, 00 retorna false.</p> <p>O comportamento “transferir()” recebe um valor informado pelo usuário e requisita ao sistema que este valor seja reduzido do saldo, retorna “true”. Caso o valor seja acima de R\$5000, 00 retorna false.</p> <p>O comportamento “retirarExtrato()” lista todo os histórico de transações efetuadas pelo cliente. Esta lista é composta por um array de objetos do tipo “Historico”.</p>
Exemplos de Objetos	<p>objeto 1: agencia: 8888 conta: 777777 tipo: CC saldo: 5000</p> <p>objeto 2: agencia: 7788 conta: 888888 tipo: CP saldo: 12000</p> <p>objeto 3: agencia: 1781 conta: 818283 tipo: CS saldo: 1800</p>

Classe - Historico

Características	id, data, operacao, valor
Comportamentos	criar(), excluir(), listar()
Descrição	Esta classe esta associada a classe “Conta” e trata todos os procedimentos realizados pelos clientes no sistema. Todos os seus métodos são privados e os atributos “data”, “operacao” e “valor” são públicos, o atributo id é privado.

	<p>O comportamento “criar()” adiciona um novo Histórico ao sistema.</p> <p>O comportamento “excluir()” remove um Histórico específico por meio do atributo “id” do sistema.</p> <p>O comportamento listar() constrói e retorna um array com todas os procedimentos realizados pelo usuário registrados no sistema.</p>
Exemplos de Objetos	<p>objeto 1: id: 0000001 data: “01-01-2016” operacao: Operacao() valor: -100</p> <p>objeto 2: id: 0000002 data: “01-01-2016” operacao: Operacao() valor: -300</p> <p>objeto 3: id: 0000003 data: “05-01-2016” operacao: Operacao() valor: 800</p>

Classe - Operacao

Características	id, codigo, descricao, tipo
Comportamentos	criar(), excluir()
Descrição	<p>Esta classe esta associada a classe “Historico” e registra todas a operações do sistema.</p> <p>O comportamento “criar()” adiciona um novo registro de operação no sistema.</p> <p>O comportamento “excluir()” remove um registro específico por meio do atributo “id” do sistema.</p>
Exemplos de Objetos	<p>objeto 1: id: 000001 codigo: 135 descricao: “Débito em conta corrente no caixa eletrônico” tipo: “debito”</p> <p>objeto 2: id: 000002</p>

	<p>codigo: 138 descricao: "Transferência para conta de mesma titularidade" tipo: "transferencia"</p> <p>objeto 3: id: 000003 codigo: 456 descricao: "Depósito por meio de auto atendimento" tipo: "credito"</p>
--	--

Conclusão

Desenvolvemos neste trabalho três visões do sistema, estas são:

- Visão de casos de uso;
- Visão lógica;
- Visão de processos.

Neste trabalho foi utilizado o diagrama de Casos de Uso da UML. Podemos aplicar também os diagrama de Classe e diagrama de Sequência.

Referências

Roteiro de Aprendizagem 1, 2 e 3 e os recursos nestes disponíveis
Disponível em: <http://lms.infnet.edu.br/moodle/course/view.php?id=594>
Acesso em: 10 de setembro 2016.

Java orientação a objetos (associacao, composicao, agregacao)
Disponível em:
<http://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao>
Acesso em: 10 de setembro 2016.