

Faculdade de Tecnologia Senac Goiás

Curso: Gestão de Tecnologia da Informação

Disciplina: Banco de dados

Alunos: Gabriel Napolis Mascarenhas, Lucas Magalhães Sousa, Lucas Keven Costa.

Análise e projeto de sistemas.

Documento de Requisitos do Sistema

inBank

Versão 1.0

Conteúdo

- 1. INTRODUÇÃO 3
 - 1.1 VISÃO GERAL DO DOCUMENTO 3
 - 1.2 CONVENÇÕES, TERMOS E ABREVIACÕES 3
 - 1.2.1 Identificação dos requisitos 3
 - 1.2.2 Prioridades dos requisitos 3
- 2. DESCRIÇÃO GERAL DO SISTEMA 4
 - 2.1 ABRANGÊNCIA E SISTEMAS RELACIONADOS 4
- 3. REQUISITOS FUNCIONAIS (CASOS DE USO) 4
 - [RF001] Inserir dados 4
 - [RF002] Retornar calculo 4
 - [RF003] Imprimir simulação 4
- 4. REQUISITOS NÃO-FUNCIONAIS 5
 - [NF001] Usabilidade 5
 - [NF002] Desempenho 5
 - [NF003] Segurança 5
 - [NF004] Hardware e Software 5
- 5. REFERÊNCIAS 6

1. Introdução

Este documento especifica os requisitos do sistema inBank, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

1. Seção 2 – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
2. Seção 3 – Requisitos funcionais (casos de uso): especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
3. Seção 4 – Requisitos não-funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
4. Seção 5 – Referências: apresenta referências para outros documentos utilizados para a confecção deste documento.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do nome do identificador do requisito,

de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

5. Essencial é o requisito sem o qual o sistema não entra em funcionamento. Requisitos

essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

6. Importante é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

7. Desejável é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

2.1 Abrangência e sistemas relacionados

Documento de Requisitos

Projeto: inBank

O sistema inBank é uma ferramenta destinada à bancos e financeiras. Fornece uma plataforma intuitiva e de fácil acesso para simulações de empréstimos e financiamentos baseadas em tabelas previamente definidas.

O sistema fornece todos os dados de forma rápida, expondo ao usuário uma ferramenta de maneira cômoda e eficiente.

3. Requisitos funcionais (casos de uso)

[RF001] Manter dados do cliente

Descrição do caso de uso: Este requisito propõe ao sistema realizar a manutenção (cadastro, exclusão, edição e listagem) dos dados do cliente no banco de dados, dando ao usuário possibilidade de realizar as operações básicas.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: não tem.

Saídas e pós-condição: não tem.

[RF002] Realizar simulações

Descrição do caso de uso: O sistema deverá possibilitar ao usuário realizar simulações de

empréstimos e financiamentos através dos modelos PRICE e SAC.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: o cliente deve estar previamente cadastrado (ver [Inserir dados – RF001]).

Saídas e pós-condição: não tem.

4. Requisitos não-funcionais

[NF001] Usabilidade

O sistema deve ter boa usabilidade pois é importante que seja simples. Para isso a interface com o usuário deve ser simples e intuitiva.

Prioridade: Essencial Importante Desejável

[NF002] Desempenho

Documento de Requisitos

Projeto: inBank

É importante que o sistema tenha um baixo tempo de resposta.

Prioridade: Essencial Importante Desejável

[NF003] Segurança

O sistema não pode apresentar falhas de segurança visando possível integração futura com um sistema bancário.

Prioridade: Essencial Importante Desejável

[NF004] Hardware e Software

Visando criar um produto com maior extensibilidade, reusabilidade e flexibilidade, deve ser adotado como linguagem principal de desenvolvimento Java seguindo cuidadosamente as técnicas de orientação a objetos. Entretanto, outras linguagens também poderão ser usadas quando indicações técnicas recomendem. O uso da linguagem Java permite não especificar qual será o sistema operacional e a máquina em que o programa irá executar. No entanto, essa máquina deverá se comunicar com um sistema de banco de dados. O sistema de banco de dados adotado será o PostgreSQL, por funcionar bem e por ser gratuito.

Prioridade: Essencial Importante Desejável

5. Referências

1. Booch, Grady – UML : guia do usuário / Grady Booch, James Rumbaugh, Ivar Jacobson ;

tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. – Rio de Janeiro : Elsevier, 2005 – 9ª reimpressão.

2. Bezerra, Eduardo – Princípios de análise e projeto de sistemas com UML / Eduardo Bezerra. – Rio de Janeiro : Elsevier, 2007 – 2ª reimpressão.

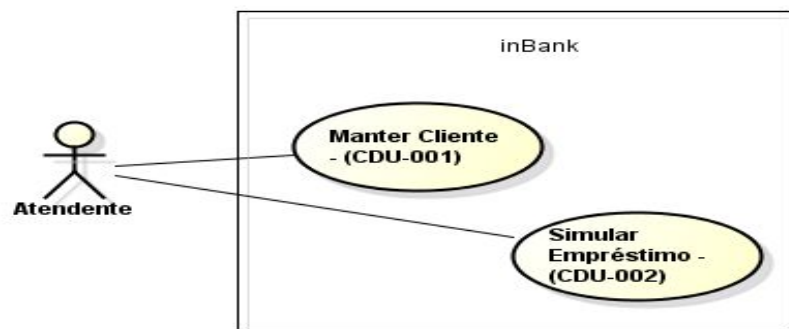
Especificação de caso de uso

Sistema: inBank

Introdução

- Diagrama de casos de uso

I



Descrição do sistema

O diagrama de casos de uso do sistema inBank é constituído por dois casos de uso: Manter Cliente (CDU – 001) e Simular Empréstimo (CDU – 002). Estes dois casos de uso representam as funcionalidades do sistema e serão detalhados nas próximas seções. O sistema de uma maneira geral tem a função de simular empréstimos para aposentados,

funcionários públicos e pensionistas com base nos modelos PRICE e SAC, partindo de uma taxa fixa.

□ Manter Cliente (CDU – 001)

Usuários: Atendente.

Descrição do caso de uso: O caso de uso “Manter Cliente” (CDU – 001) tem como responsabilidade as operações realizadas com os dados dos clientes. Salvar, editar, listar e excluir clientes são responsabilidades deste caso de uso. Os dados do cliente serão mantidos em uma base de dados administrada pelo próprio sistema – inBank.

Fluxo principal:

- Na tela que apresenta uma lista com os clientes, o usuário deve apertar o botão “novo” para adicionar novo cliente.
- Após aparecerá uma tela com os campos para serem preenchidos.
- Depois de preencher estes campos o usuário deve apertar em salvar.

Fluxo secundário:

- Na tela que apresenta uma lista com os clientes, o usuário deve apertar o botão “excluir cliente” no cliente que este deseja deletar.
- Aparecerá uma mensagem de confirmação e o usuário será deletado do sistema.

Fluxo secundário:

- Na tela que apresenta uma lista com os clientes, o usuário deve apertar o botão “manter cliente” no cliente que este deseja manipular.
- Logo após o sistema listará os dados do cliente já nos campos de edição. Para editar, basta alterar dentro destes campos as informações pertinentes ao cliente.
- O usuário, ao terminar suas edições, deverá clicar no botão “salvar”, e suas alterações serão realizadas.

□ Simular Empréstimo (CDU – 002)

Usuário: Atendente.

Descrição do caso de uso: O caso de uso “Simular Empréstimo” (CDU – 002) é responsável pelo cálculo da simulação do empréstimo. A partir de dados fornecidos pelo cliente o sistema deverá calcular o valor das parcelas assim como o prazo do financiamento. Após o cálculo o sistema deve apresentar os resultados obtidos ao usuário.

Fluxo principal:

- Na tela que apresenta uma lista com os clientes, o usuário deve apertar o botão “Simular” de acordo com cliente selecionado.

- Após o sistema dará a possibilidade ao usuário de informar qual modelo (SAC ou PRICE) deseja seguir.
- Logo após o usuário informar o sistema retornará os resultados obtidos.

□ Regras de negócio

RN-01: O campo “cpf” pertinente ao cliente deve receber no máximo onze caracteres, ignorando os pontos e o ítem.

RN-02: A simulação só deve ser autorizada se o valor das parcelas da simulação, mais o valor das parcelas de outros financiamentos for menor que 30% do salário líquido do cliente.

Sistema inbank

ESPECIFICAÇÃO DAS CLASSES

Classes de entidade

Referências da classe: ClienteControl, ClienteDao

Descrição: A classe “Cliente” é responsável por gerir dados pertinentes ao cliente. Ela possui todas as informações necessárias para o sistema manipular os clientes.

Atributos:

- nome : String
- cpf : String
- salLiquido : Double
- idade : int

| Tipo | Atributo | Descrição |
|--------|------------|--|
| String | nome | Mantém o nome do cliente |
| String | Cpf | Mantém o cpf do cliente – chave primária |
| Double | salLiquido | Mantém o salário líquido do cliente |
| Int | Idade | Mantém idade do cliente |

Métodos: Contrutor, getters e setters.

Classe: Simulacao

Referências da classe: SimulacaoControl

Descrição: A classe “Simulacao” é responsável por gerir dados pertinentes à simulação. Ela possui todas as informações necessárias para o sistema manipular os dados das simulações.

Atributos:

- qtdParcelas : int
- valorParcela : Double

| Tipo | Atributo | Descrição |
|--------|--------------|--|
| int | qtdParcelas | Define quantas parcelas terão no financiamento |
| Double | valorParcela | Define o valor de cada parcela |

Métodos: Construtor, getters e setters

Classes persistência

Classe: ClienteDao

Referências da classe: ClienteControl, Cliente

Descrição: A classe “ClienteDao” é responsável por gerir dados pertinentes ao cliente

junto ao sistema de banco de dados. É esta classe que realiza toda a comunicação, pertinente a cliente, entre o sistema e o SGBD

Métodos:

- void salvar(Cliente c): este método recebe através da classe “ClienteControl” os dados de uma instância da classe “Cliente” e salva no banco de dados.
- void alterar(Cliente c): este método altera no banco de dados as informações de determinado cliente previamente identificado no sistema.
- void excluir(Cliente c): este método exclui do banco de dados as informações de determinado cliente previamente identificado no sistema.
- ArrayList listar(): este método retorna um array list com todos os elementos presentes no banco de dados.

Classes de controle

Classe: ClienteControl

Referências da classe: Cliente, ClienteDao

Descrição: A classe “ClienteControl” é responsável por controlar toda a interação entre as classes de tela e a manipulação de dados do cliente interna do sistema.

Atributos:

- cliente : Cliente

- clienteDao : ClienteDao

| Tipo | Atributo | Descrição |
|------------|------------|--------------------------------|
| Cliente | cliente | Instância da classe Cliente |
| ClienteDao | clienteDao | Instância da classe ClienteDao |

Métodos:

- registrarCliente(): Este método solicita ao objeto clienteDao que crie um novo cliente na base de dados. Antes ele instancia a classe cliente e armazena nessa instância os dados do novo cliente.

- editarCliente(): Este método solicita ao objeto clienteDao que edite um cliente já existente na base de dados. Este método passa o identificador do cliente para que seja possível encontra-lo no banco de dados.

- excluirCliente(): Este método solicita ao objeto clienteDao que exclua um cliente já existente na base de dados. Este método passa o identificador do cliente para que seja possível encontra-lo no banco de dados.

- listarTodos(): Este método solicita ao objeto clienteDao uma lista contendo todos os clientes cadastrados no sistema.

Classe: SimulacaoControl

Referências da classe: Simulacao

Descrição: A classe "SimulacaoControl" é responsável por controlar toda a interação entre as classes de tela e a manipulação de dados da simulacao interna do sistema.

Atributos:

- simulacao : Simulacao

| Tipo | Atributo | Descrição |
|-----------|-----------|---|
| Simulacao | simulacao | Este atributo é uma instância da classe Simulacao |

Métodos:

- calcularPossiveisFinanciamentos(): este método é responsável pela realização dos cálculos referentes às simulações.

- calcularTaxaFinan(): este método é responsável pelo cálculo que avalia se é possível realizar o financiamento(30% do salário).

Classes de visão

Classe: TelaListaCliente

Referências da classe: ClienteControl e SimulacaoControl

Descrição: A classe “TelaListaCliente” é a tela principal do sistema. É responsável por capturar as ações do usuário e dar continuidade às ações.

Métodos:

- novoCliente(): o método é um botão que inicia a inserção de um novo usuário.
- editarCliente(): o método é um botão que inicia o processo de edição dos dados de um cliente.
- excluirCliente(): o método é um botão que inicia o processo de exclusão dos dados de um cliente.
- calcularSimulacao(): o método é um botão que inicia o processo de simulação de empréstimo.

Classe: TelaExibirSimulacao

Referências da classe: SimulacaoControl

Descrição: A classe “TelaExibirSimulacao” é a tela do sistema responsável por exibir a mensagem final ao usuário. Esta mensagem pode ser a simulação ou algum problema encontrado no processo.

Métodos:

- exibirSimulacao(): este método é responsável por exibir a mensagem final na tela.

Diagramas :

Diagrama de classe

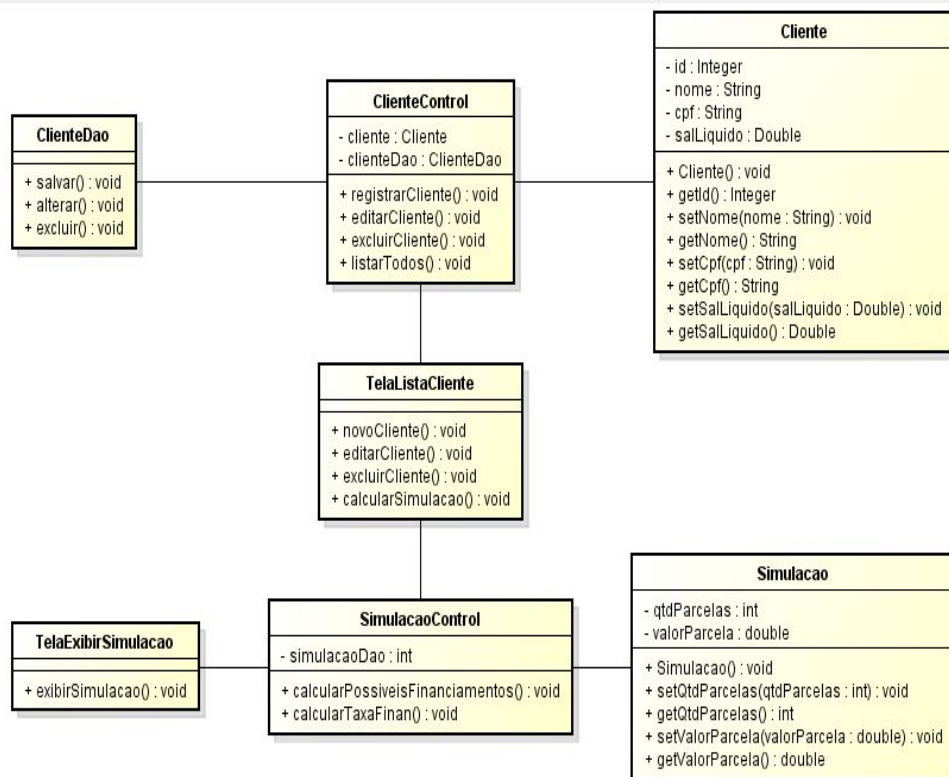
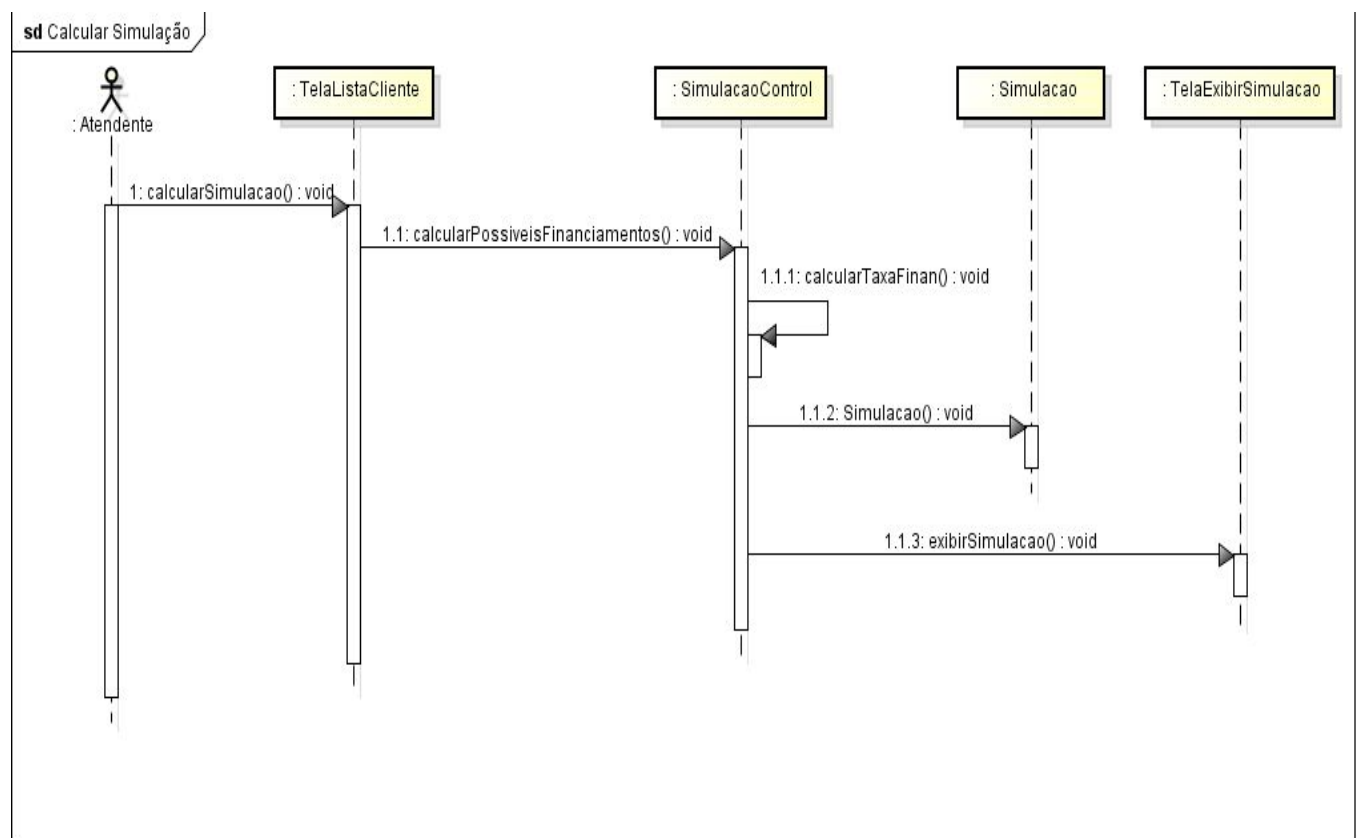
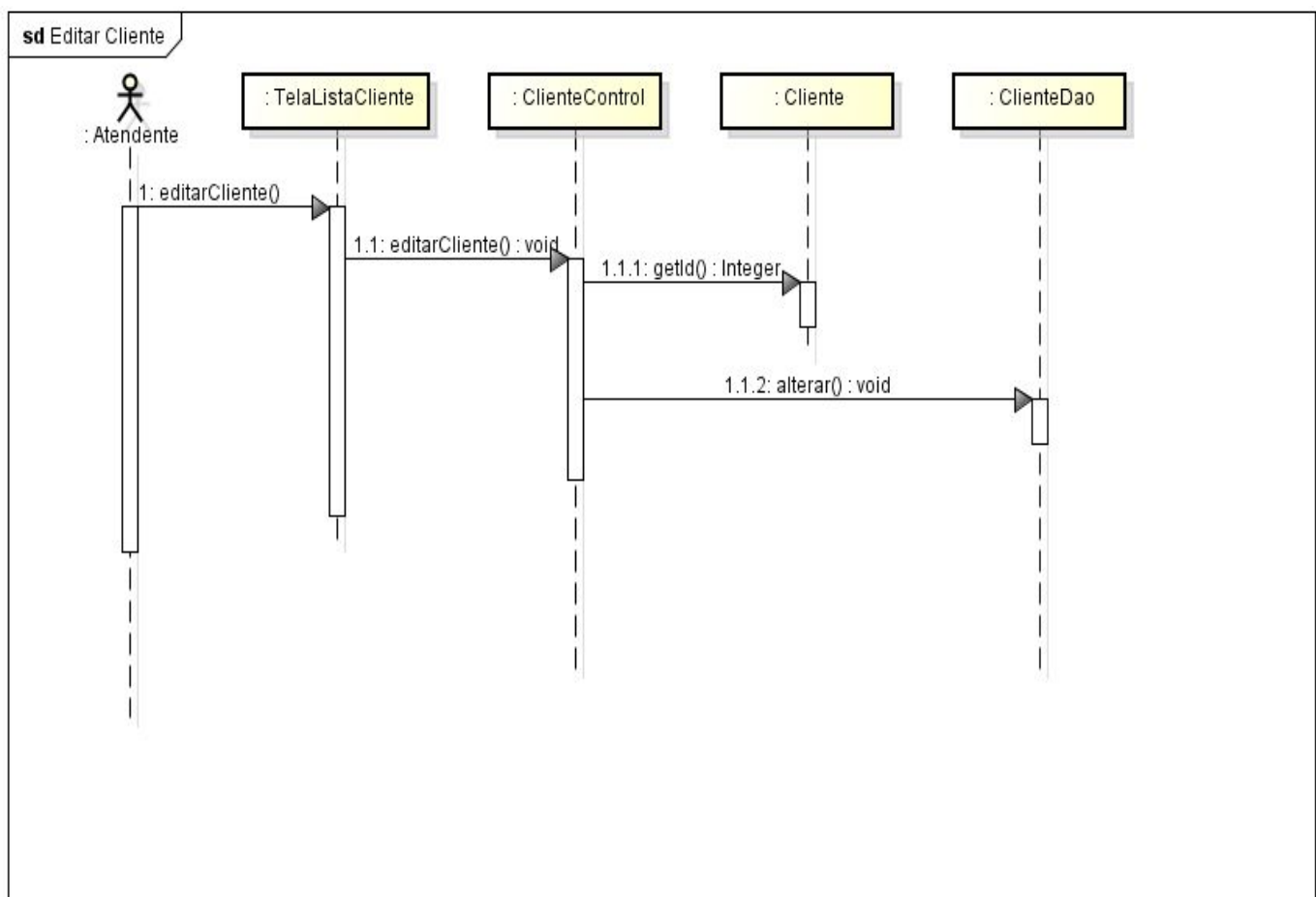


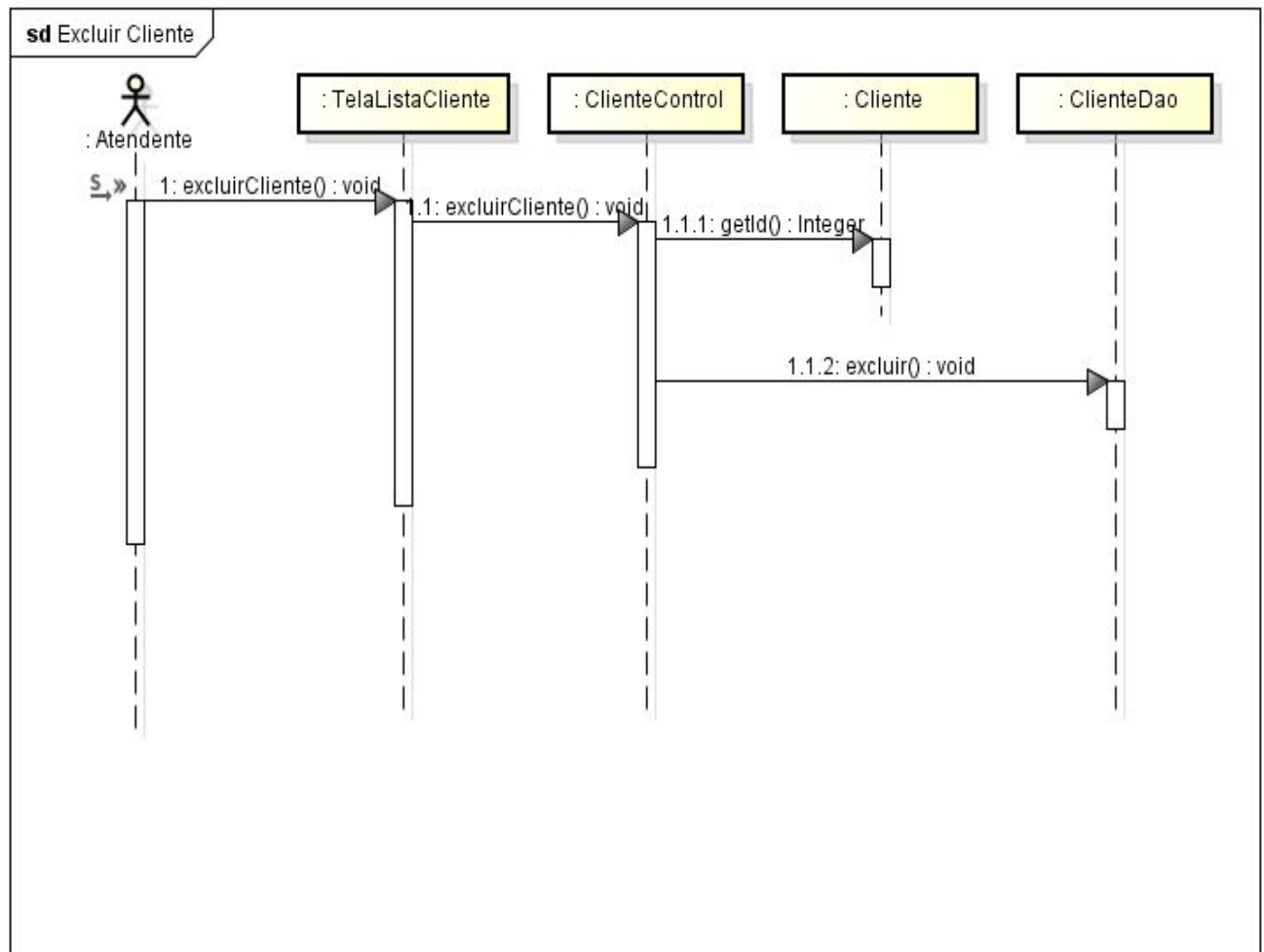
Diagrama de sequência



Editar cliente



Excluir cliente



Registrar cliente

sd Registrar Cliente

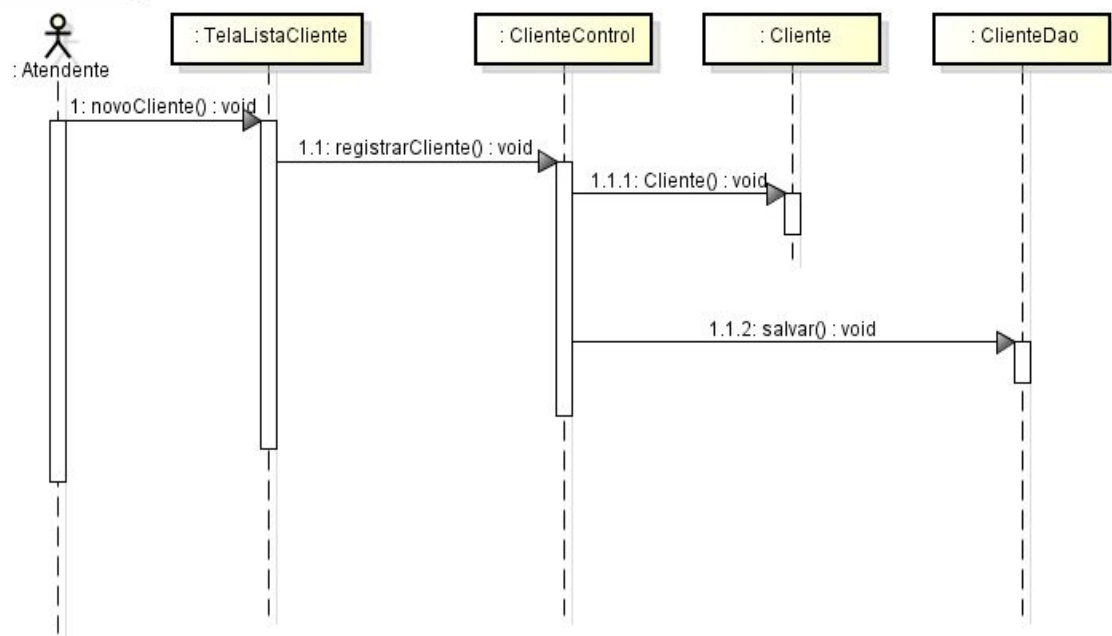


Diagrama de Atividade

