



UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

IGOR MEDEIROS VANDERLEI

MARCOS RAFAEL

MICHEL OLIVEIRA

RELATÓRIO DO PROJETO
BANK OF UFPA

Garanhuns-PE

2022

**IGOR MEDEIROS VANDERLEI
MARCOS RAFAEL
MICHEL OLIVEIRA**

Relatório do projeto de desenvolvimento de software apresentado ao curso de Bacharelado em Ciência da Computação, como requisito parcial para obtenção da aprovação na disciplina Programação Orientada a Objetos.

Orientador: Igor Medeiros Vanderlei

Garanhuns– PE

2022

RESUMO

O resumo é escrito em apenas um parágrafo. Deve apresentar uma breve contextualização (falar do que se trata o sistema). Em seguida, fala-se brevemente dos objetivos deste trabalho. Segue apresentando os métodos utilizados (falar de OO). Por fim, apresentam-se os resultados obtidos. Geralmente, o resumo é escrito após o término do trabalho.

Palavras-Chave:

Conteúdo

1. INTRODUÇÃO.....	5
1.1 VISÃO GERAL DO DOCUMENTO.....	5
1.2 CONVENÇÕES, TERMOS E ABREVIACÕES.....	5
1.2.1 Identificação dos requisitos.....	5
1.2.2 Prioridades dos requisitos.....	5
2. DESCRIÇÃO GERAL DO SISTEMA.....	6
2.1 IDENTIDADE VISUAL.....	6
2.2 SISTEMAS RELACIONADOS.....	6
3. REQUISITOS.....	7
3.1 ATORES DO SISTEMA.....	7
3.2 DIAGRAMA DE CASOS DE USO.....	8
3.3 ESPECIFICAÇÃO DE CASOS DE USO.....	9
3.3.1 [UC001] Selecionar Conta.....	9
3.3.2 [UC002] Gerenciar Cliente.....	9
3.4 REQUISITOS NÃO-FUNCIONAIS.....	11
3.5 CASOS DE TESTE.....	11
4. MODELAGEM.....	13
4.1 ARQUITETURA DA SOLUÇÃO.....	13
4.2 DIAGRAMA DE CLASSES.....	13
4.3 PROTÓTIPO DE TELAS.....	13
4.4 API END POINTS.....	13
5. CRONOGRAMA.....	14
6. RESULTADOS.....	15
7. CONSIDERAÇÕES FINAIS.....	16
8. GLOSSÁRIO.....	17

1. Introdução

Este documento apresenta o relatório do projeto de desenvolvimento de software do sistema **BANK OF UFAPE**, ...

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: ...**
3. <<Incluir a descrição de todas as seções do 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 da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [UC01] ou [NF01] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

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”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **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.
- **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.

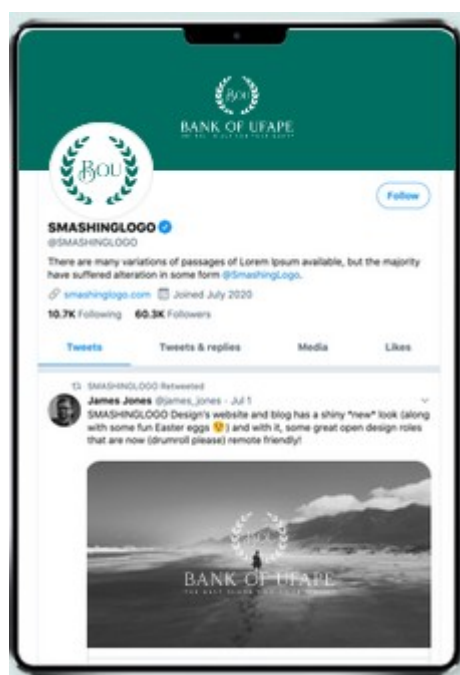
2. Descrição geral do sistema

<< Incluir a descrição geral do sistema, conforme a primeira entrega, com o texto revisado >>

2.1 Identidade Visual

<< Apresentar a logomarca e a paleta de cores a ser utilizada.

Ex: Por se tratar de uma aplicação que trabalha diretamente com os recursos financeiros do cliente, optou-se por utilizar um padrão de comunicação visual sóbrio e *clean*. Com fontes sem serifa, para facilitar a leitura por pessoas com baixa visão. O padrão de cores utiliza o verde, branco e preto. O verde foi escolhido por lembrar a ideia de dinheiro, o branco e o preto foram escolhidos como cores complementares, por possuir um bom contraste com o verde.>>



2.2 Sistemas relacionados

<< Pesquisar dois ou três sistemas (um por membro da equipe) relacionados, destacando suas principais características (plataforma utilizada, se é open source ou proprietário, principais funcionalidades), pontos fortes ou pontos fracos >>

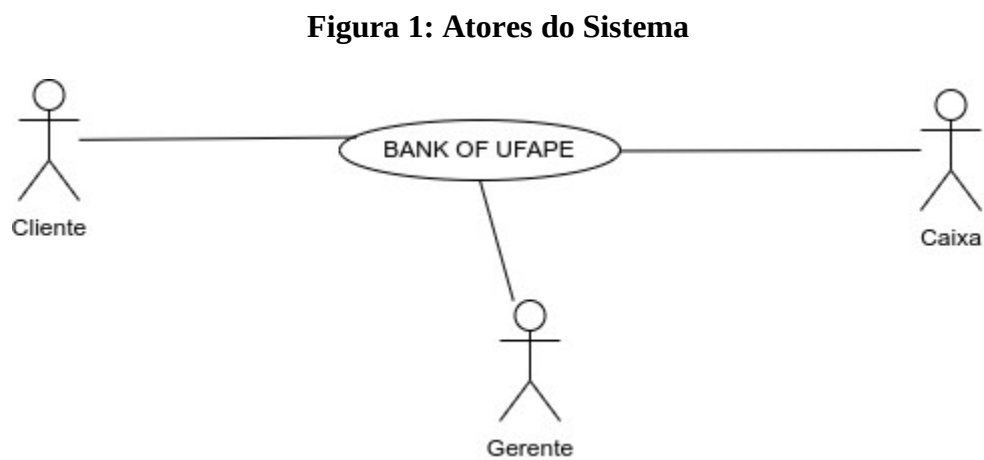
3. Requisitos

<< Esta seção descreve O QUE deve ser feito >>

3.1 Atores do sistema

<< Atores são os perfis de usuário que utilizarão o sistema.

Conforme pode ser observado na Figura 1, os atores usuários do sistema Bank of Ufape, estão divididos nas categorias Cliente, Caixa e Gerente. Segue com uma breve descrição de cada um dos atores.>>



Fonte: Os Autores (2022)

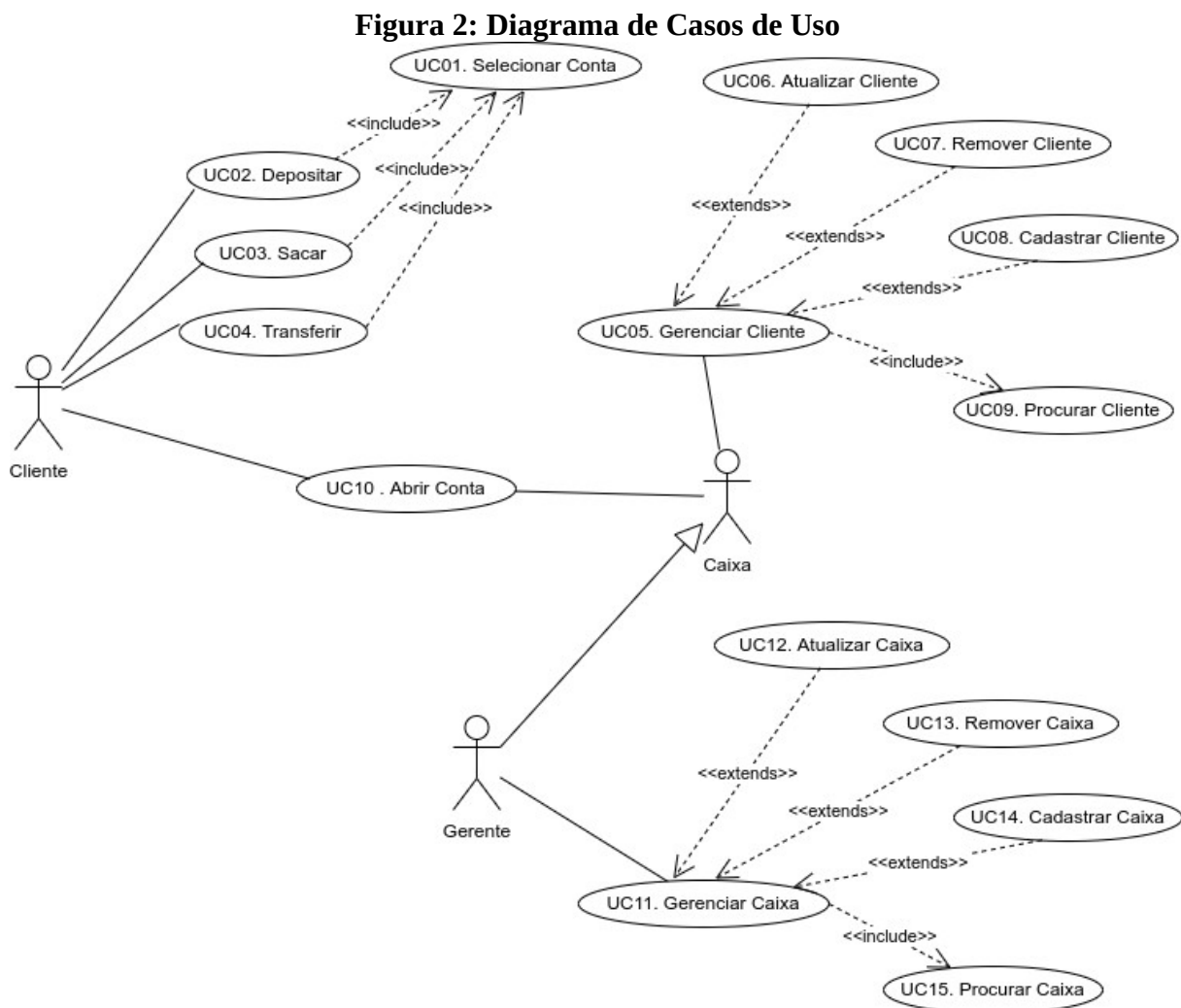
3.2 Diagrama de casos de uso

<< O diagrama de caso de uso apresenta uma visão geral das funcionalidades do sistema, através dos casos de uso (elipses), suas dependências e a sua relação com os atores.

A relação <<include>> indica que um caso de uso possui dependência com outro. No diagrama abaixo “UC02. Depositar” possui dependência com “UC01. Selecionar Conta”, ou seja, para realizar um depósito, o cliente precisa selecionar qual a conta receberá o depósito. Geralmente, o include é utilizado para extrair e reutilizar casos de uso.

A relação <<extends>> indica uma extensão, ou uma ação que poderá ou não ser executada. Quando um Caixa “UC05. Gerenciar Cliente”, ele poderá ou não “UC06. Atualizar Cliente”.

A herança entre os atores indica que um usuário tem permissão de realizar todos os casos de uso de um outro. No diagrama, o Gerente pode fazer tudo que o Caixa faz, além dos casos de uso específicos do próprio gerente.>>



Fonte: Os Autores (2022)

3.3 Especificação de casos de uso

<< Detalhar cada um dos casos de uso. OBS. Dividir adequadamente a tarefa entre os membros da equipe >>

3.3.1 [UC001] Selecionar Conta

Id do Caso de uso:	UC01		
Nome do Caso de uso:	Selecionar Conta		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	

Ator:	Cliente
Descrição:	O caso de uso tem como finalizada selecionar uma conta para que o Cliente possa realizar alguma operação sobre a mesma.
Pré-condições:	Cliente possui alguma conta
Pós-condições:	A conta selecionada está “ativa” no sistema do cliente.
Prioridade:	Alta
Frequência de uso:	Alta
Complexidade	Baixa
Risco	Alto
Fluxo principal:	<ol style="list-style-type: none">1. Imediatamente após o login do usuário, o sistema exibe a lista das contas para que o Cliente selecione qual delas deseja operar.2. O cliente clica na conta que deseja operar.3. O sistema exibe a tela inicial do gerenciamento da conta selecionada.
Exceções:	FE001. Caso o cliente não possua nenhuma conta ativa, o sistema deve exibir uma mensagem para que o mesmo deve ter pelo menos uma conta para poder utilizar as funções.

3.3.2 [UC002] Gerenciar Cliente

Id do Caso de uso:	UC05		
Nome do Caso de uso:	Gerenciar Cliente		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	

Ator:	Caixa, Gerente
Descrição:	O caso de uso tem como objetivo realizar gerenciamento do cadastro dos clientes
Pré-condições:	
Pós-condições:	Cadastro de Clientes atualizado
Prioridade:	Alta
Frequência de uso:	Alta
Complexidade	Alta
Risco	Alta
Fluxo principal:	<ol style="list-style-type: none">1. O usuário seleciona a opção de gerenciar clientes no menu principal do sistema2. O sistema exibe a tela na qual o usuário pode fazer a busca por cliente (UC09. Procurar Cliente)

	<ol style="list-style-type: none"> Após localizar o cliente, o usuário informa se deseja atualizar o cliente (UC06), remover o cliente (UC07) ou cadastrar um novo cliente (UC08) De acordo com a opção escolhida pelo usuário um dos sub-fluxos a seguir é executado.
Sub-Fluxo UC06. Atualizar Cliente [NOTA01]	<ol style="list-style-type: none"> O sistema exibe um formulário contendo todos os dados dos clientes (CPF, NOME, DATA DE NASCIMENTO, ENDEREÇO) O usuário digita as informações atualizadas e clica no botão “atualizar” O sistema verifica os dados digitados. Caso todos os dados estejam corretos, o sistema armazena os dados no banco e exibe uma mensagem de confirmação para o usuário. Caso haja algum dado incorreto, o sistema exibe uma mensagem de erro e volta para o passo 1 (FE02).
Sub-Fluxo UC07. Remover Cliente [NOTA01] [NOTA02][NOTA03]	<ol style="list-style-type: none">
Sub-Fluxo UC08. Cadastrar Cliente [NOTA01]	<ol style="list-style-type: none">
Exceções:	FE002. Caso haja algum dado incorreto, o sistema exibe uma mensagem clara de erro, indicando qual(is) campo(s) apresentam inconformidade com o padrão.
Notas:	<p>NOTA01: A qualquer momento, o usuário pode desistir da operação, clicando no botão cancelar. O sistema então exibe uma mensagem de confirmação, indicando que todos os dados serão perdidos e só efetua o cancelamento mediante confirmação do usuário.</p> <p>NOTA02: Caso a ação Remover Cliente seja realizada por um usuário do tipo Caixa, o sistema deve solicitar a autorização de um Gerente, através da digitação de sua senha.</p> <p>NOTA03: A operação de Remover Cliente se limitar a desativar o cliente para uso do sistema, não apagando suas as informações do banco de dados.</p>

3.4 Requisitos Não-Funcionais

<<Listar características desejáveis do sistema que não se enquadram como funcionalidade. Aqui devem ser apresentados aspectos relacionados à interface, usabilidade, desempenho, segurança, plataformas, etc >>

[RNF001] O design da interface do sistema deve ser responsável, ou seja, adaptável a diversos tamanhos de tela.

Prioridade: Essencial X Importante Desejável

3.5 Casos de Teste

<< Cada membro deverá fazer UM teste unitário, UM teste de integração, UM teste de API e UM teste de interface >>

Id do Caso de Teste:	TC01		
Nome do Caso de uso:	Verificar o funcionamento da operação de depósito		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	
Tipo de Teste:	Unitário		
Caso de Uso:	UC02. Depositar		
Pré-condições:	n/a		
Pós-condições:	n/a		
Fluxo:	<ol style="list-style-type: none">1. Instanciar um objeto do tipo Conta, com saldo inicial conhecido2. Realizar uma operação de depósito, passando como parâmetro um valor conhecido.3. Verificar se após o débito, o objeto encontra-se com o saldo adequadamente atualizado.		
Saída Esperada:	Objeto conta com o saldo atualizado com o valor do saldo inicial + o valor do depósito		

Id do Caso de uso:	TC02		
Nome do Caso de uso:	Verificar se após a operação de depósito, o objeto será atualizado no BD.		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	
Tipo de Teste:	Integração		
Caso de Uso:	UC02. Depositar		
Pré-condições:	Existir ao menos uma conta cadastrada no BD.		
Pós-condições:	Conta atualizada.		
Fluxo:	<ol style="list-style-type: none"> 1. Procurar uma conta cadastrada no sistema (pela fachada) 2. Consultar o saldo desta conta 3. Realizar uma operação de depósito (método da fachada), passando como parâmetro um valor conhecido e o número da conta 4. Procurar novamente a mesma conta(pela fachada) 5. Verificar se o saldo foi adequadamente atualizado 		
Saída Esperada:	Objeto conta com o saldo atualizado com o valor do saldo inicial + o valor do depósito foi atualizado no banco de dados (verificação passos 4 e 5)		

Id do Caso de uso:	TC03		
Nome do Caso de uso:	Verificar a tentativa de realizar um depósito em uma conta inexistente.		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	
Tipo de Teste:	API		
Caso de Uso:	UC02. Depositar		
Pré-condições:	Conta não cadastrada no sistema		
Pós-condições:	n/a		
Fluxo:	<ol style="list-style-type: none"> 1. Gerar uma chamada de API para o endpoint depositar, passando como parâmetro uma conta inexistente 2. Verificar se o retorno da API apresenta o erro. 		
Saída Esperada:	Response com STATUS 400 (BAD REQUEST)		

Id do Caso de uso:	TC04		
Nome do Caso de uso:	Verificar a tentativa de realizar um depósito em uma conta inexistente.		
Criado Por:	Igor Medeiros	Atualizado Por:	
Data da Criação:	23/08/2022	Data da atualização:	
Tipo de Teste:	Interface		
Caso de Uso:	UC02. Depositar		
Pré-condições:	Conta não cadastrada no sistema		
Pós-condições:	n/a		
Fluxo:	<ol style="list-style-type: none"> 1. Acessar o sistema 2. Clicar no link de depósito 3. Informar um número de conta inexistente e um valor qualquer 4. Clicar no botão confirmar 5. Verificar se uma mensagem de erro é exibida na tela 		
Saída Esperada:	A mensagem de erro deve informar claramente sobre o motivo do erro.		

4. Modelagem

<<Esta seção descreve COMO deve ser feito >>

4.1 Arquitetura da Solução

<< Descrever com texto e imagem a arquitetura da solução. Destacar a utilização de Front end e Back end, em cada um deles, incluir a descrição das camadas, linguagens e tecnologias >>

4.2 Diagrama de Classes

<< Incluir somente as classes básicas >>

4.3 Protótipo de Telas

<< Desenho a mão livre das principais telas da aplicação. Fazer referência das telas com os casos de uso >>

4.4 API End Points

<< Descrever os endpoints da API, de acordo com o padrão tinyspec >>

5. Cronograma

<< Elaborar um cronograma das atividades, indicando o prazo e responsável por cada ação. Durante a divisão das tarefas, é importante que todos os membros da equipe trabalhem em todas as partes do sistema. >>

Atividade	Responsável (previsto)	Data (previsto)	Responsável (realizado)	Data (realizado)
1. Documento do projeto	Igor, Rafael, Michel	03/10		
1.1. Apresentação	Igor	23/08	Igor	23/08
1.2. Atores e Casos de Uso	Igor, Rafael, Michel			
1.3. Especificação UC01, UC02, UC03	Igor			
1.4. Especificação UC04, UC05, UC06	Rafael			
1.4. Especificação UC07, UC08, UC09	Michel			
1.5. Requisitos não funcionais	Michel			
1.6. Protótipo tela UC01, UC02, UC03	Igor			
1.7. Protótipo tela UC04, UC05, UC06	Michel			
1.8. Diagrama de Classes				
1.9 Revisão Final				
2. Back end	Igor, Rafael, Michel	03/10		
2.1. Criação e setup do projeto no git				
2.2. Desenvolvimento das classes X, Y, Z				
2.3. Desenvolvimento dos repositórios X, Y, Z				
2.4. Tratamento de exceção A, B e C				
2.5. Fachada ...				
3. Front end	Igor, Rafael, Michel	03/10		
3.1. Definição do template				
3.2. Desenvolvimento da tela inicial				
3.3. Desenvolvimento da tela X				
3.4. Conexão da Tela X com o back				

6. Resultados

<< Descrever o sistema desenvolvido, apresentando prints das telas principais >>

7. Considerações Finais

<< Apresentar as conclusões do trabalho, resgatando os conceitos da programação OO e os seus impactos percebidos durante o projeto >>

8. Glossário

Esta seção é usada para definir a terminologia específica do sistema BANK OF UFAPE. Ela poderá ser expandida durante todo o ciclo de vida do projeto e será usada por toda a equipe de desenvolvimento.

Termo	Definições
Clientes	Toda pessoa física ou jurídica que possui ao menos uma conta no banco.
Caixa	Funcionário do banco que trabalha no atendimento direto ao cliente, tendo a possibilidade de auxiliar o Cliente nas operações de suas contas.
Gerente	Funcionário do Banco responsável pelo bom funcionamento de uma agência. O gerente deve supervisionar o trabalho dos funcionários e realizar o atendimento especializado dos clientes.
Conta Corrente	
Conta Poupança	
Depósito	
Saque	
Transferência	
...	