CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS VII - UNIDADE TIMÓTEO
Curso: Engenharia de Computação
Disciplina: Desenvolvimento de Sistemas
Professor: Odilon Corrêa

Nota

Valor: 12 Pontos

JavaScript - Exercício Avaliativo II

Objetivo

O exercício tem como objetivo é aprimorar os conhecimentos na área de desenvolvimento de aplicações web utilizando as tecnologias HTML, CSS e JavaScript.

Exercício

O exercício consiste em desenvolver um sistema bancário que permita aos clientes gerenciar suas contas correntes. O sistema deve suportar operações bancárias essenciais, como abertura de contas, depósitos, saques e consulta de extratos.

Sua tarefa é implementar os requisitos pendentes na aplicação web, cuja base já está disponível em um repositório do GitHub (https://github.com/odiloncorrea/ds-exercicio-avaliativo-02).

Requisito Funcional	Situação
RF-01: Cadastrar Cliente	Implementado
RF-02: Autenticar Cliente	Implementado
RF-03: Exibir Menu de Acesso	Implementado
RF-04: Cadastrar Conta	-
RF-05: Realizar Operação Financeira	-
RF-06: Consultar Extrato	-
RF-07: Controlar Acesso e Segurança	-

Critérios de avaliação:

- RF-04: Cadastrar Conta — 2.5 pontos
- RF-05: Realizar Operação Financeira — 2.5 pontos
- RF-06: Consultar Extrato — 2.5 pontos
- RF-07: Controlar Acesso e Segurança — 2.5 pontos
- RNF-001: Responsividade e Tecnologia — 2.0 pontos

Testes unitários

O discente deve utilizar o arquivo disponibilizado pelo professor no sistema acadêmico para testar os *endpoints* do API.

Entrega

O discente deve enviar um arquivo **compactado** com todos os códigos da solução implementada através do SIGAA até o dia **25/11**.

Caso o discente disponibilize também os arquivos em um repositório no **GitHub** e informe o link no envio, receberá uma **bonificação de 2 pontos**.

ANEXO I

1. Descrição geral do sistema

O sistema bancário tem como objetivo oferecer uma plataforma digital segura, intuitiva e funcional para gerenciamento de contas correntes. Ele permite que clientes realizem operações essenciais como abertura de contas, saques, depósitos e consulta de extratos. O sistema visa garantir a praticidade no controle financeiro bancário e fornecer ao cliente autonomia sobre suas movimentações.

2. Usuários e ambiente

O público-alvo é composto por clientes bancários em geral, que desejam gerenciar suas contas de forma autônoma. O sistema deve atender usuários com diferentes níveis de familiaridade com tecnologia, oferecendo uma interface amigável e acessível. O sistema deve ser acessível a partir de qualquer dispositivo com conexão à internet.

3. Requisitos Funcionais

3.1. RF-01: Cadastrar Cliente

O sistema deve oferecer funcionalidade para o cadastro de novos clientes, exigindo, obrigatoriamente, os seguintes dados: nome, CPF e senha. O CPF deve ser único, não sendo permitido o registro de clientes com CPF duplicado.

3.2. RF-02: Autenticar Cliente

O acesso ao sistema é restrito a usuários previamente cadastrados. A autenticação deve ser realizada por meio de CPF e senha.

3.3. RF-03: Exibir Menu de Acesso

Após a autenticação, o sistema deve identificar o cliente e exibir seu primeiro nome, além do saldo total de todas as suas contas, juntamente com as funcionalidades disponíveis. As opções acessíveis ao cliente são:

- Contas
- Extrato
- Operação financeira
- Sair

3.4. RF-04: Cadastrar Conta

O sistema deve permitir que o cliente visualize suas contas existentes, exibindo o número da conta e o respectivo saldo. Também deve ser possível cadastrar novas contas. Ao criar uma nova conta:

- O saldo inicial é definido como zero.
- O número da conta é gerado automaticamente, seguindo o formato AA-999999, em que:
  - AA corresponde às duas primeiras letras do nome do cliente (em maiúsculas);
  - 999999 é um número aleatório de seis dígitos.

Antes de registrar a nova conta, o sistema deve verificar se o número gerado é único na base de dados. Caso o número já exista, um novo valor deve ser gerado até que a unicidade seja garantida.

3.5. RF-05: Realizar Operação Financeira

O sistema deve permitir que o cliente realize operações financeiras em suas contas, abrangendo depósitos e saques. Para efetuar a operação, o cliente deve selecionar uma de suas contas cadastradas, informar o tipo de operação (saque ou depósito) e o valor desejado.

Em operações do tipo saque, o sistema deve validar previamente se o saldo disponível é suficiente para o valor solicitado. Caso contrário, a operação deve ser recusada.

3.6. RF-06: Consultar Extrato

O sistema deve permitir que o cliente consulte o extrato de uma conta. Para isso, o cliente deve selecionar uma de suas contas cadastradas. A consulta deve ser restrita às contas pertencentes ao cliente autenticado e deve exibir a operação que originou cada lançamento.

3.7. RF-07: Controlar Acesso e Segurança

O sistema deve restringir o acesso de usuários não autenticados a todas as páginas, exceto login e cliente, e encerrar automaticamente a sessão após 1 minuto de inatividade ou ausência de requisições.

4. Requisitos Não Funcionais

4.1. RNF-001: Responsividade e Tecnologia

O sistema deve ser acessível a partir de qualquer navegador moderno e apresentar design responsivo, garantindo boa usabilidade e adaptação correta da interface em diferentes dispositivos, incluindo computadores, notebooks, tablets e smartphones.

4.2. RNF-002: Armazenamento em Nuvem

O sistema deve realizar o armazenamento os dados na nuvem, utilizando a API disponibilizada pelo professor para comunicação e operações de leitura e escrita de dados.

- Cliente – Login

Método	URL	Descrição
GET	http://18.229.132.2:8888/api/clientes/login	Autenticar o cliente

  - O login e a senha devem ser enviados no corpo da requisição para autenticação do cliente. Em caso de sucesso, a estrutura JSON ao lado apresenta os dados retornados do cliente autenticado.

Requisição

```
{
    "cpf": "111.111.111-111",
    "senha": "111"
}
```

Resposta

```
{
    "id": 1,
    "nome": "João Silva",
    "cpf": "111.111.111-11"
}
```

- Cliente – CRUD

Método	URL	Descrição
GET	http://18.229.132.2:8888/api/clientes	listar todos os clientes
GET	http://18.229.132.2:8888/api/clientes/{id}	buscar cliente por ID
POST	http://18.229.132.2:8888/api/clientes	cadastrar novo cliente
PUT	http://18.229.132.2:8888/api/clientes/{id}	atualizar cliente existente
DELETE	http://18.229.132.2:8888/api/clientes/{id}	remover cliente
GET	http://18.229.132.2:8888/api/clientes/exists?cpf=111.111.111-11	verificar CPF

Requisição

```
{
    "id": 1,
    "nome": "João Silva",
    "cpf": "111.111.111-111",
    "senha": "111"
}
```

Resposta

```
{
    "id": 1,
    "nome": "João Silva",
    "cpf": "111.111.111-11"
}
```

- Conta – CRUD

Método	URL	Descrição
GET	http://18.229.132.2:8888/api/contas	listar todas as contas
GET	http://18.229.132.2:8888/api/contas/{id}	buscar conta por ID
POST	http://18.229.132.2:8888/api/contas	cadastrar nova conta
PUT	http://18.229.132.2:8888/api/contas/{id}	atualizar conta existente
DELETE	http://18.229.132.2:8888/api/contas/{id}	remover conta
GET	http://18.229.132.2:8888/api/contas/cliente/{idCliente}	listar todas as contas pelo ID do cliente
GET	http://18.229.132.2:8888/api/contas/exists?numero=JS-000001	verificar número

Requisição	Resposta
{ "id": 1, "numero": "JS-000001", "saldo": 0, "idCliente": 1 }	{ "id": 1, "numero": "JS-000001", "saldo": 1800.0, "idCliente": 1 }

- Lançamento – CRUD

Método	URL	Descrição
GET	http://18.229.132.2:8888/api/contas	listar todos os lançamentos
GET	http://18.229.132.2:8888/api/contas/{id}	buscar lançamento por ID
POST	http://18.229.132.2:8888/api/contas	cadastrar novo lançamento
PUT	http://18.229.132.2:8888/api/contas/{id}	atualizar lançamento existente
DELETE	http://18.229.132.2:8888/api/contas/{id}	remover lançamento
GET	http://18.229.132.2:8888/api/lancamentos/conta/{idConta}	listar todos os lançamentos pelo ID da conta

  - O tipo do lançamento deve ser obrigatoriamente SAQUE ou DEPÓSITO.
  - Ao registrar o lançamento, a API atualiza automaticamente o saldo da conta

Requisição	Resposta
{ "id": 1, "valor": 200.0, "tipo": "SAQUE", "idConta": 1 }	{ "id": 9, "valor": 200.0, "tipo": "DEPOSITO", "idConta": 2 }

5. Diagrama de Casos de Uso

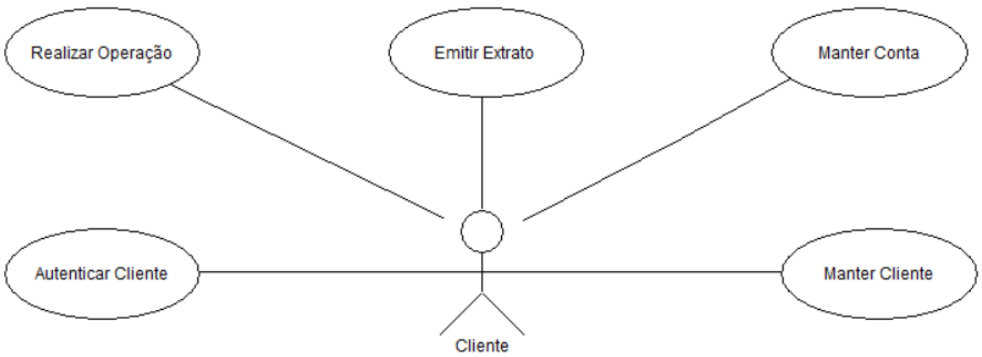

Figura 01 – Diagrama do Caso de Uso

6. Estrutura do Banco de Dados

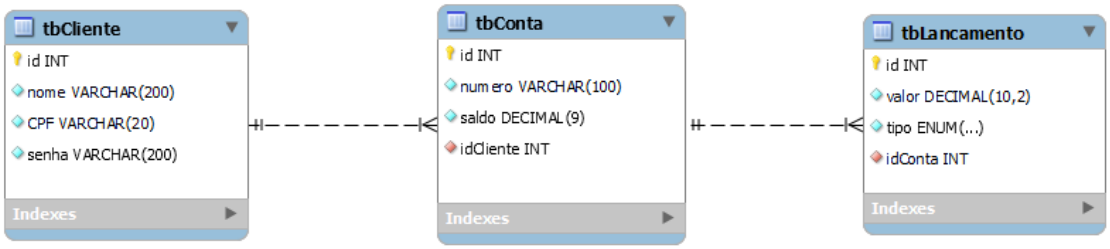

Figura 02 – Esquema do Banco de Dados

7. Protótipos

Os protótipos apresentados a seguir têm como objetivo ilustrar os principais requisitos do sistema, servindo como referência para facilitar sua compreensão e orientar o processo de desenvolvimento. É importante destacar que os discentes têm total liberdade para propor e implementar soluções alternativas, desde que atendam aos requisitos funcionais e não funcionais estabelecidos. A estrutura, o layout e os componentes utilizados nos protótipos não são restritivos, permitindo adaptações conforme as decisões de projeto adotadas durante a implementação.

1.1. PT-01: Cadastrar e Autenticar Cliente

Figura 03 – Protótipos dos RF-01 e RF-02

1.2. PT-02: Exibir Menu de Acesso

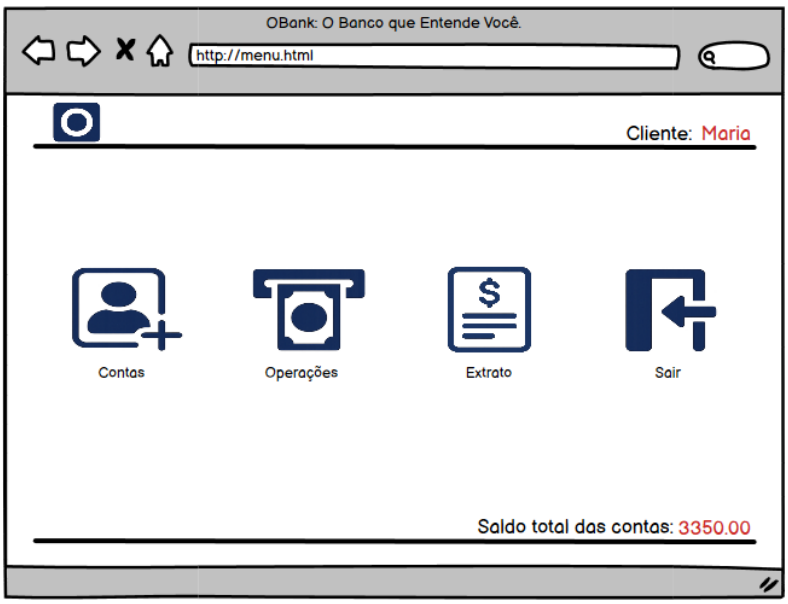

Figura 04 – Protótipo do RF-03

1.3. PT-03: Cadastrar Conta

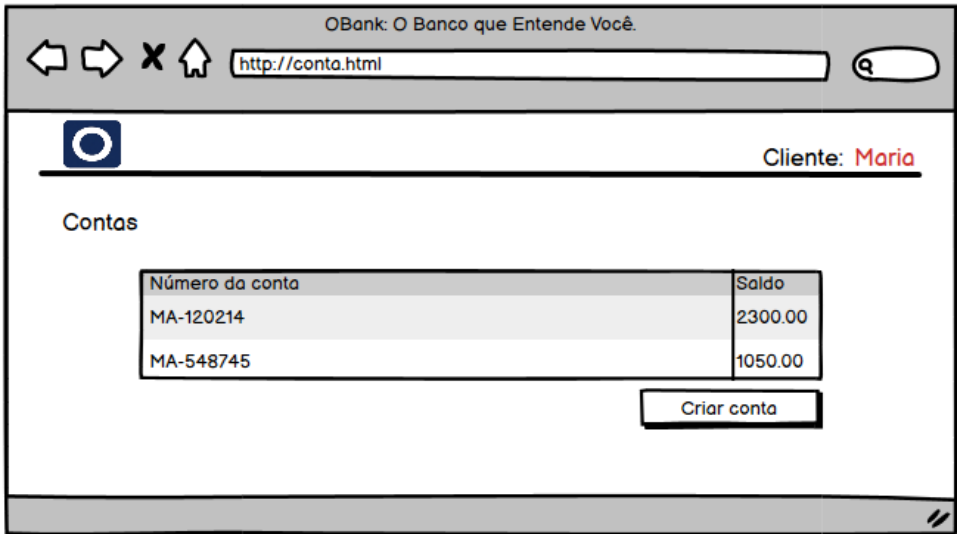

Figura 05 – Protótipo do RF-04

1.4. PT-03: Realizar Operação Financeira

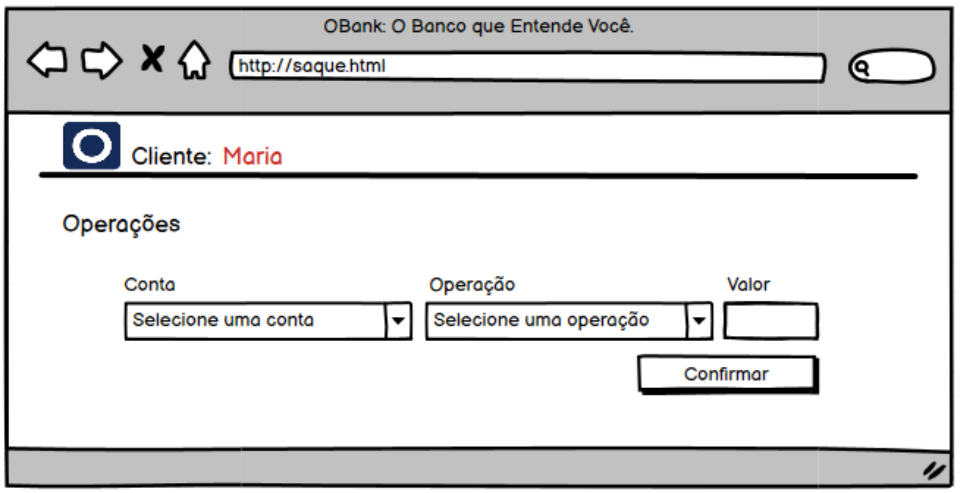

Figura 06 – Protótipo do RF-05

1.5. PT-04: Consultar Extrato

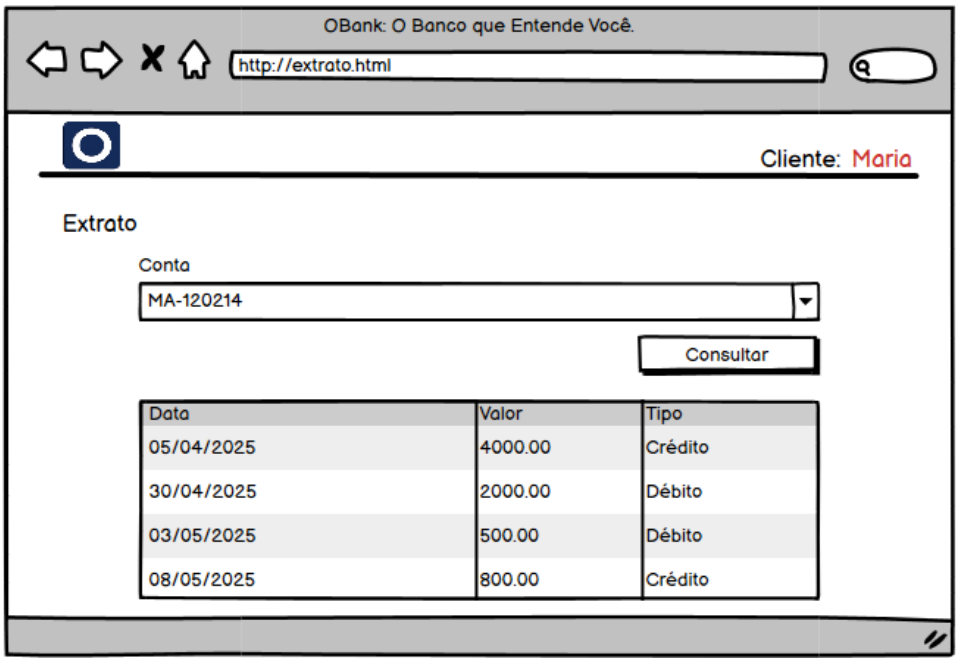

Figura 07 – Protótipo do RF-06