

PAULO V. F. ANDRADE  
RAÍSSA J. P. SANTOS  
RITA C. C. CRUZ

Projeto Web I – Internet Banking

**PAULO V. F. ANDRADE  
RAÍSSA J. P. SANTOS  
RITA C. C. CRUZ**

## **Projeto Web I – Internet Banking**

Relatório técnico apresentado à disciplina  
de Desenvolvimento de Sistemas Web I,  
como requisito para obtenção de nota e  
para acompanhamento de projeto.

Universidade Federal do Rio Grande do Norte - UFRN

Instituto Metrópole Digital - IMD

Bacharelado em Tecnologia da Informação

Natal  
2022

# SUMÁRIO

<b>1</b>	<b>Repositório</b>	<b>3</b>
<b>2</b>	<b>Visão de Produto</b>	<b>4</b>
2.1	Objetivo . . . . .	4
2.2	Funcionalidades . . . . .	4
2.3	Benefícios . . . . .	4
2.4	Valor Agregado . . . . .	5
<b>3</b>	<b>Protótipos de Interfaces</b>	<b>6</b>
3.1	Paleta . . . . .	6
3.2	Telas . . . . .	6
<b>4</b>	<b>Escolhas de usabilidade</b>	<b>12</b>
<b>5</b>	<b>Tecnologias Utilizadas</b>	<b>15</b>
<b>6</b>	<b>Organização dos arquivos</b>	<b>17</b>
6.1	O diretório backend . . . . .	17
<b>7</b>	<b>Rodando a aplicação</b>	<b>19</b>
7.1	Criando o banco de dados . . . . .	19
7.2	Executando o servidor . . . . .	19
7.3	Executando a aplicação . . . . .	19

# 1 REPOSITÓRIO

O sistema descrito e implementado neste projeto pode ser encontrado neste link.

## 2 VISÃO DE PRODUTO

### 2.1 Objetivo

O objetivo deste sistema é implementar um Internet Banking, de modo a simplificar operações bancárias na vida das pessoas ao propor um banco eficiente totalmente digital.

### 2.2 Funcionalidades

- Débito
- Crédito
- Pix
- Transferência
- Recarga
- Extrato
- Dashboard do Cliente
- Conta Dependente

### 2.3 Benefícios

O internet banking permite ao usuário realizar operações bancárias pela internet. Dessa forma, os usuários podem acessar contas bancárias tradicionais a partir de um dispositivo conectado à rede para efetuar movimentações como transferências, pagamentos e outros serviços.

Além das vantagens do internet banking, um banco digital se propõe a oferecer os mesmos serviços bancários que os bancos tradicionais, porém com maior comodidade ao usuário, maior segurança, menor burocracia e menores taxas sobre serviços. Para tanto, deve haver pesquisa, análise, investimento e desenvolvimento constantes em tecnologia.

## 2.4 Valor Agregado

Nosso sistema visa usabilidade, comodidade, simplicidade, segurança e eficácia, oferecendo um serviço com interface amigável e diversas funções para atender as necessidades do usuário. Considerando que acesso aos serviços bancários é uma necessidade cotidiana, o sistema aqui proposto vem para ser um facilitador, pois além de diminuir consideravelmente o tempo e a burocracia de realizar tais serviços por meios tradicionais, como agências e terminais de autoatendimento, o sistema se propõe a agregar novas funcionalidades, como a conta dependente e a transferência para múltiplos destinatários, tudo de maneira simplificada e segura.

## 3 PROTÓTIPOS DE INTERFACES

O projeto Figma pode ser acessado neste link.

### 3.1 Paleta



Figura 1: Paleta de cores para o sistema

### 3.2 Telas

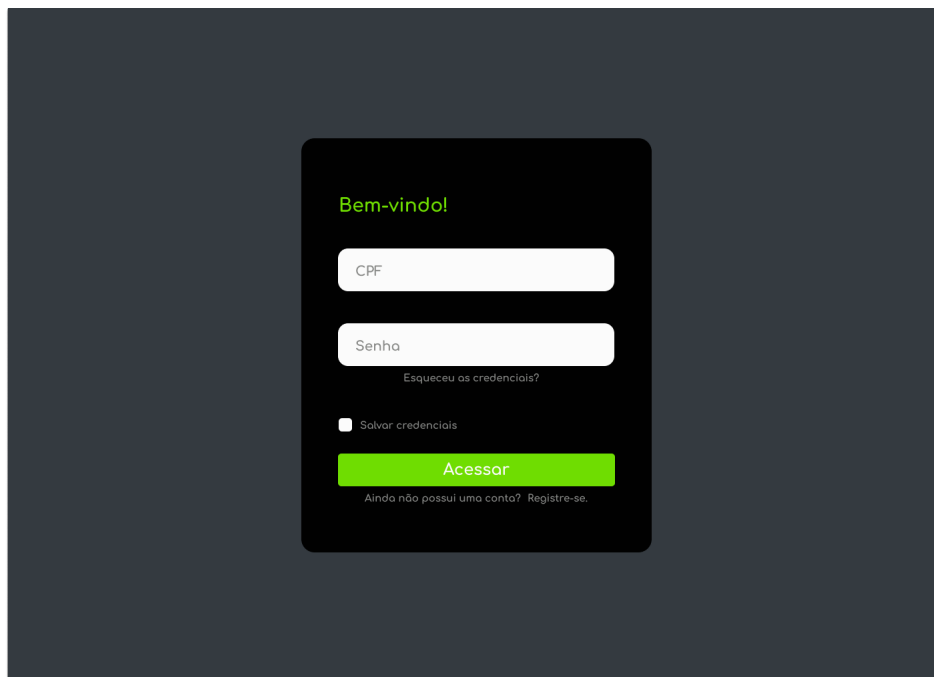


Figura 2: Tela de login

Registration form (Tela de cadastro) with the following fields:

- Nome\*
- Data de nascimento\*
- CPF\*
- Endereço\*
- Telefone\*
- Senha\*
- Email
- Confirmar senha\*

☐ Li e concordo com os termos de uso.

**Cadastrar**

Figura 3: Tela de cadastro

**DBM**

Nome Sobrenome  
123456789-0

- DashBoard
- Transferência
- Pix
- Recarga
- Depósito
- Extrato

**GERENCIAMENTO**

- Configurações da Conta
- Atendimento ao Cliente
- SAIR

**Dashboard**

Saldo disponível  
**R\$2.000,00**

Fatura atual  
**R\$1.000,00**

**Dependentes**

Nome da pessoa	Saldo
Nome da pessoa	Saldo

**Últimos 30 dias**

Nome da pessoa	Transação	Valor	Data e hora
Nome da pessoa	Transação	Valor	Data e hora
Nome da pessoa	Transação	Valor	Data e hora
Nome da pessoa	Transação	Valor	Data e hora
Nome da pessoa	Transação	Valor	Data e hora

Figura 4: Tela principal





Figura 5: Tela de SAC

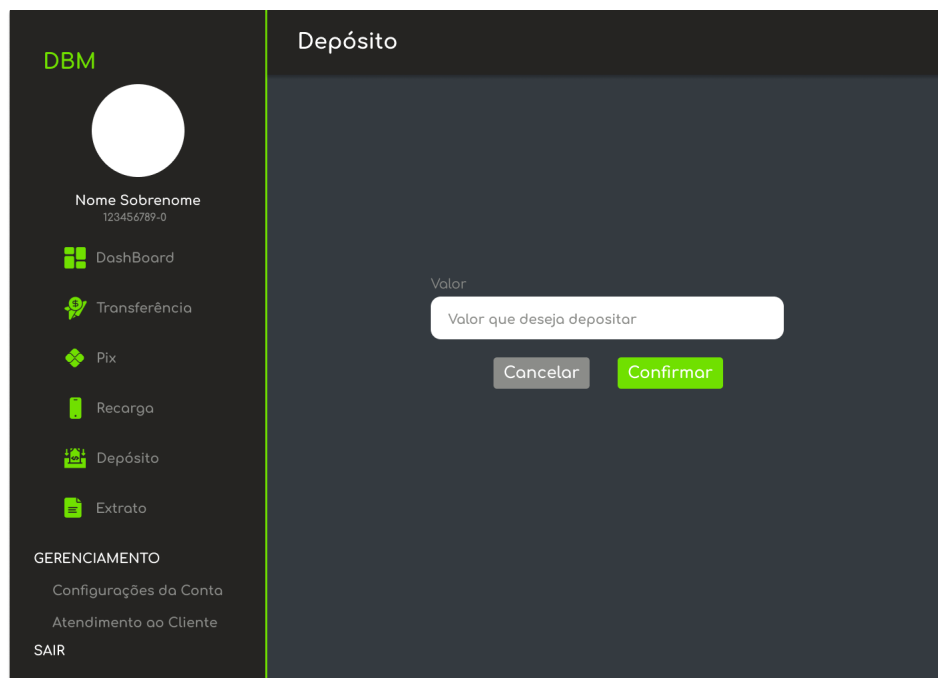


Figura 6: Tela de depósito

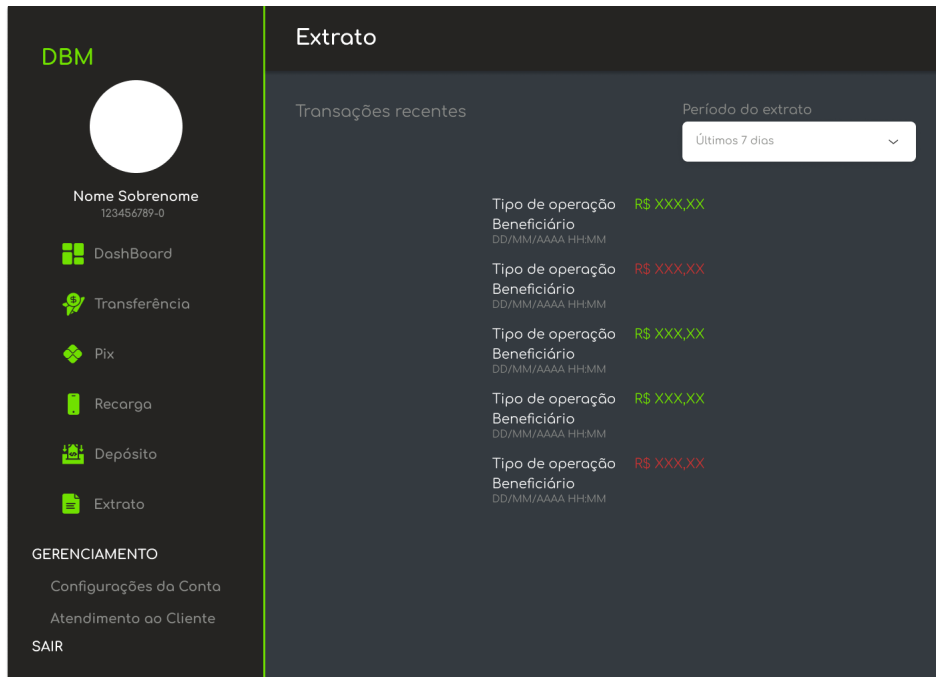


Figura 7: Tela de extrato

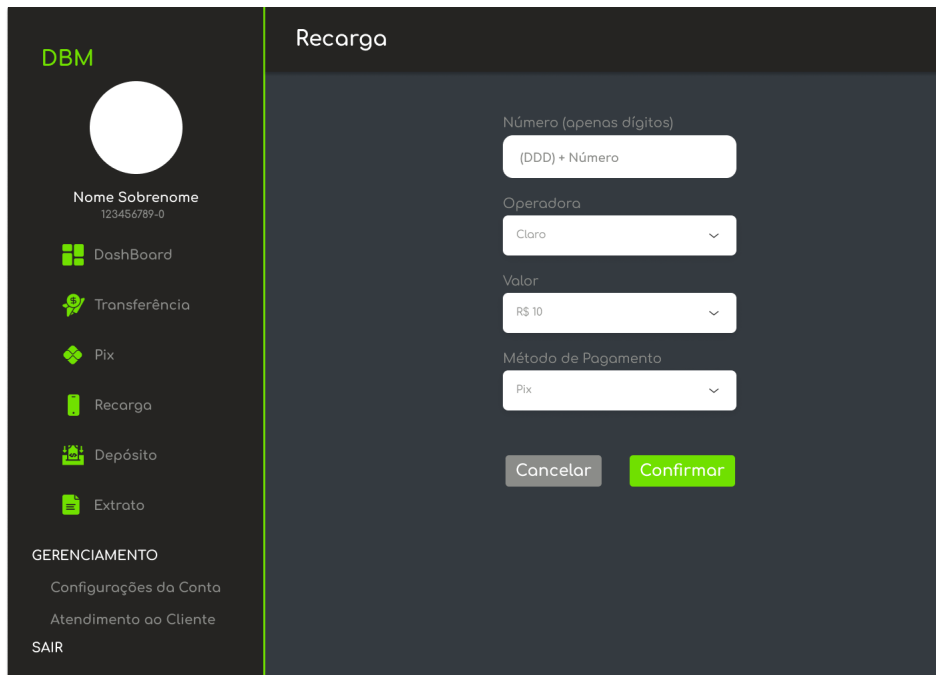


Figura 8: Tela de recarga

DBM

Nome Sobrenome  
123456789-0

DashBoard

Transferência

Pix

Recarga

Depósito

Extrato

GERENCIAMENTO

Configurações da Conta

Atendimento ao Cliente

SAIR

PIX

Chave Pix (apenas números)

Insira o chave pix

Valor

Insira o valor de transferência

Descrição

Mensagem para o destinatário

Cancelar Confirmar

Figura 9: Tela de Pix

DBM

Nome Sobrenome  
123456789-0

DashBoard

Transferência

Pix

Recarga

Depósito

Extrato

GERENCIAMENTO

Configurações da Conta

Atendimento ao Cliente

SAIR

Configurações da Conta

DBM CashBack

9999 9999 9999 9999

08/25 370

CLIENTE C. L. ALEATORIO

DBM CashBack

9999 9999 9999 0000

09/25 380

CLIENTE C. L. ALEATORIO

Limite Total  
R\$1600,00

Disponível  
R\$500,00

Débitos

Nome da Operação	Valor	Data e hora
Nome da Operação	Valor	Data e hora
Nome da Operação	Valor	Data e hora

Figura 10: Tela de configurações da conta

DBM

Nome Sobrenome  
123456789-0

- DashBoard
- Transferência
- Pix
- Recarga
- Depósito
- Extrato

GERENCIAMENTO

- Configurações da Conta
- Atendimento ao Cliente

SAIR

### Transferências

Agência (apenas números)

Insira o agência

Conta (apenas números)

Insira o conta

Tipo de conta

Conta corrente

Valor

Insira o valor a ser transferido

Cancelar Confirmar

Figura 11: Tela de transferência

## 4 ESCOLHAS DE USABILIDADE

O principal elemento css para usabilidade do sistema é a pseudo-classe `:hover`. Ela corresponde ao momento em que o usuário interage com um elemento com um dispositivo apontador, mas não necessariamente o ativa. Geralmente, é acionado quando o usuário passa o mouse sobre um elemento com o cursor.

O atributo modificado pela pseudo-classe é sempre relativo à cor do elemento. Um exemplo é exposto na Figura 12. À esquerda, o elemento em seu estado normal. À direita, quando o usuário posiciona o cursor sobre.



Figura 12: Elemento da sidebar com efeito hover

Na Figura 13, relativa à página SAC, a pseudo-classe `:hover` tem como objetivo sugerir ao usuário que ele pode clicar naquele elemento. Ao clicar, o componente expande, expondo um texto explicativo. Quando expandido, ao passar o mouse sobre o título do componente, a cor deste também muda, indicando que o usuário pode clicar neste componente novamente – desta vez, para retrair-lo.

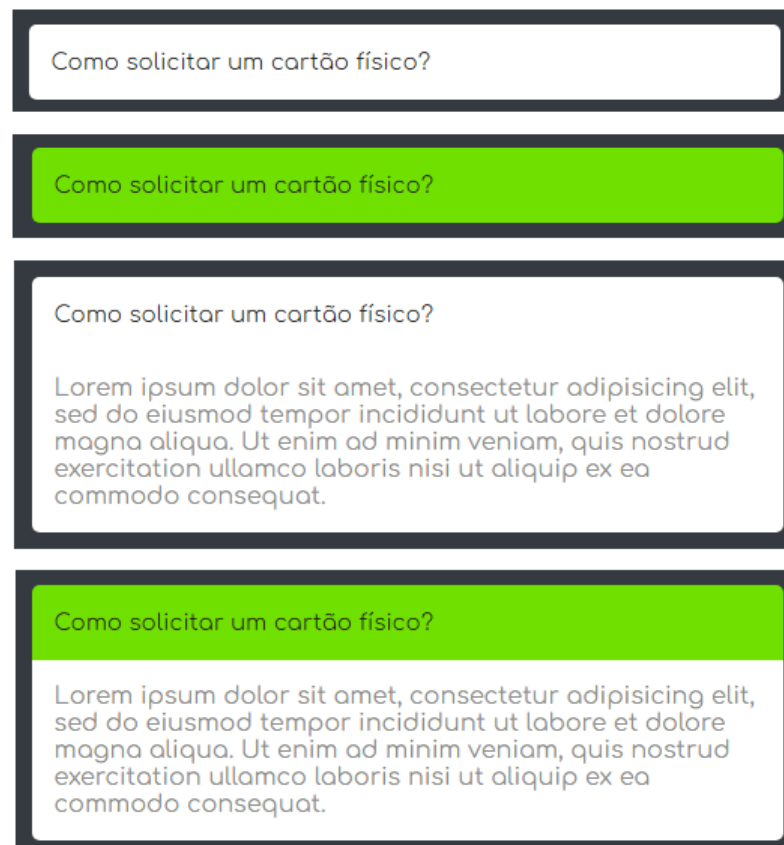


Figura 13: Elemento da página SAC com efeito hover

Outro elemento fundamental para a usabilidade do sistema é o menu em sidebar, exposto na Figura 14. Todas as interfaces internas possuem a mesma sidebar à esquerda, a qual contém um caminho para todos os acessos dentro do sistema.

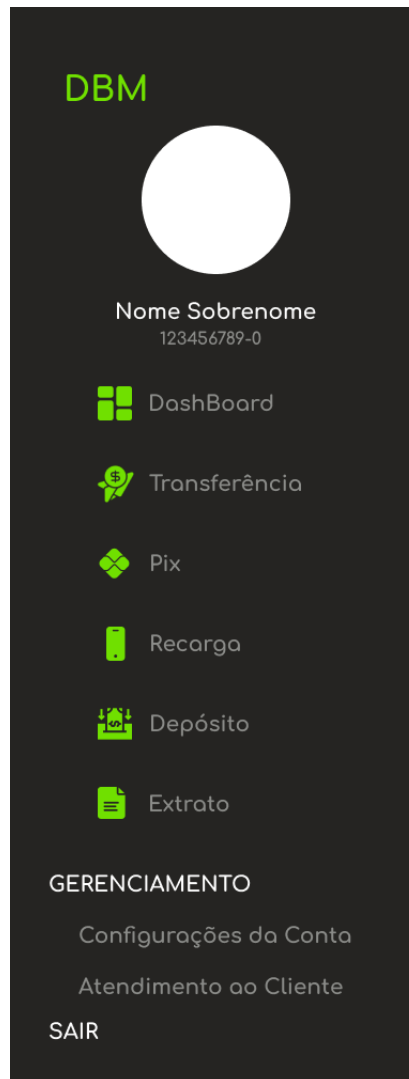


Figura 14: Sidebar

Outras escolhas relativas à usabilidade consistem em utilizar a repetição de padrões ao longo das interfaces, como os botões “Cancelar” e “Confirmar”.

## 5 TECNOLOGIAS UTILIZADAS

- Figma

O Figma é o editor online que foi utilizado na prototipagem das interfaces gráficas.

- HTML

O HTML é a linguagem de marcação utilizada na construção de páginas da aplicação web.

- CSS

O CSS é utilizado para estilizar elementos escritos em HTML.

- Javascript

O JavaScript é a linguagem de programação escolhida no desenvolvimento da aplicação.

- Node.js

O Node.js é o ambiente de execução Javascript no lado do servidor. A instalação foi realizada através do download da página oficial do Node.js.

- Express

A instalação foi realizada através do npm, o gerenciador de pacotes do Node.js. No diretório do projeto, deve ser instalado por meio do comando:

```
$ npm install express --save
```

- Bcrypt

O Bcrypt é uma biblioteca para Node.js utilizada para fazer o hash das senhas dos usuários. A instalação foi realizada através do npm, o gerenciador de pacotes do Node.js. No diretório do projeto, deve ser instalado por meio do comando:

```
$ npm install bcrypt --save
```

- Cors

O cors é um pacote do Node.js que prover um middleware utilizado para habilitar o CORS com várias opções. A instalação foi realizada através do npm, o gerenciador



de pacotes do Node.js. No diretório do projeto, deve ser instalado por meio do comando:

```
$ npm install cors --save
```

- MySQL

O MySQL é o banco de dados utilizado para armazenar as entidades da aplicação. A instalação foi realizada através do npm, o gerenciador de pacotes do Node.js. No diretório do projeto, deve ser instalado por meio do comando:

```
$ npm install mysql2 --save
```

## 6 ORGANIZAÇÃO DOS ARQUIVOS

O projeto está dividido em 5 diretórios principais: app, backend, css, icons e templates.

- **app:** diretório que contém os arquivos .js com funcionalidades utilizadas no HTML.
- **backend:** diretório que contém os arquivos referentes ao servidor.
- **css:** diretório que contém os arquivos dos estilos aplicados na páginas HTML da aplicação.
- **icons:** diretório que contém as imagens utilizadas como ícones na aplicação.
- **templates:** diretório que contém os arquivos com as páginas HTML da aplicação.

### 6.1 O diretório backend

O diretório backend contém um diretório principal: src, e mais 3 arquivos: .gitignore, package.json e package-lock.json.

- **src:** diretório que contém os arquivos que estruturam o servidor.
- **.gitignore:** arquivo que contém os nomes dos arquivos que devem ser ignorados na aplicação.
- **package.json:** arquivo de configuração das dependências da aplicação.
- **package-lock.json:** arquivo de configuração das dependências, assim como o arquivo package.json.

Dentro do diretório src, há 4 subpastas: api, database, services e utils, e também o arquivo app.js.

- **api:** diretório que contém os arquivos referentes aos controllers e routes.
- **database:** diretório que contém os arquivos de conexão com o banco de dados e o esquema do banco de dados.

- **service:** diretório que contém os arquivos que interagem com o banco de dados.
- **utils:** diretório que contém o arquivo que será utilizado em diversas partes da aplicação e serve de propósito geral.
- **app.js:** arquivo responsável pela configuração e inicialização do servidor.

Na subpasta api, há duas subpastas: controllers e routes.

- **controllers:** diretório que contém os arquivos referentes a manipulação e tratamento das requisições e respostas.
- **routes:** diretório que contém os arquivos referentes as rotas da API.

## 7 RODANDO A APLICAÇÃO

### 7.1 Criando o banco de dados

Antes de criar a tabela, é necessário a instalação do MySQL Workbench. Após a instalação completa, abra o MySQL Workbench e execute o script `create-db-dbmoney.sql` na pasta `backend/src/database`. Em seguida, execute o script `initialize-gerente.sql` na mesma pasta.

### 7.2 Executando o servidor

Para rodar o servidor, execute no seu terminal:

```
$ cd backend/
```

```
$ node src/app.js
```

### 7.3 Executando a aplicação

Para rodar a aplicação, abra o arquivo `index.html` na pasta `templates`.