

PROJETO INTERDISCIPLINAR

Dev Bank

Banco Digital

Alunos:

RGM	Nome
29604745	Gabriel Vianna
29527996	Douglas Prado Cantuario
29685729	José Ronaldo de Azevedo Júnior
30146470	Lucas Monari
29708672	Douglas Zucolotto
29660351	Fernando Macena Araújo Justo
30165709	Gabriel dos santos
29684994	Guilherme Henrique Rodrigues dos santos

UNIVERSIDADE CRUZEIRO DO SUL

PROJETO INTERDISCIPLINAR

Dev Bank

Banco Digital

Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.

Orientadores: Prof. Agnaldo e Prof. Fabio

São Paulo
2023

Sumário

1. Apresentação:	4
1.1 Justificativa e Motivação	4
1.2 Dados do Sistema.	4
2. Requisitos de Engenharia de Software & Análise e Projetos de Sistemas ...	5
3 Consideração finais	23
4 Referencias	23
APENSO 1 – Cronograma de entrega de atividades.	24

1. APRESENTAÇÃO:

1.1 Justificativa e Motivação

Criação de um programa baseado em uma startup do setor financeiro no qual realiza processos como transferências TED e PIX, abrir conta, realizar depósitos, saques, pagamentos e consultar extratos.

1.2 Dados do Sistema.

O programa desenvolvido pode ter muitas características diferentes. No entanto, algumas características comuns incluem:

- Acesso fácil e conveniente aos serviços bancários
- Uso de tecnologia para automatizar processos e reduzir custos
- Capacidade de gerenciar contas bancárias e cartões de crédito
- Capacidade de fazer pagamentos e transferências

é um banco digital que permite que os usuários gerenciem suas contas bancárias e realizem transações financeiras. Isso pode incluir recursos como:

- Verificar saldos de conta
- Fazer pagamentos e transferências
- Gerenciar cartões de crédito
- Depositar cheques
- Rastrear despesas e orçamentos

A utilização de um sistema bancário pode ser muito relevante para muitas pessoas, pois oferece uma maneira conveniente e fácil de gerenciar suas finanças.

2 REQUISITOS DE ENGENHARIA DE SOFTWARE & ANÁLISE E PROJETOS DE SISTEMAS

Escopo:

“Você está abrindo uma nova Startup do setor financeiro e possui planos de crescimento rápido dessa Startup que irá realizar todas as operações de bancos tradicionais, como saques, transferências, DOCS/TEDS, PIX, Comprovantes e demais funções. Esta startup deve manter sempre um repositório de clientes, um repositório de operações bancárias, além de oferecer ao cliente histórico de todas as transações, bem como informes de seus valores aplicados em conta.

Esta aplicação voltada para o mercado financeiro deve possuir capacidade de realizar a integração com outros sistemas bancários para operações.

Além disso, esta conta trata-se de uma conta digital, ou seja, sem agência física. Sendo assim, o sistema também deve prover uma capacidade de interação instantânea para o envio e recebimento de mensagens como canais de atendimento.”

Foi realizado o levantamento dos requisitos para a criação do sistema em questão baseado no escopo citado acima.

Requisitos Funcionais:

- O sistema deve permitir que os usuários façam depósitos em suas contas.
- O sistema deve permitir que os usuários retirem dinheiro de suas contas.
- O sistema deve permitir que os usuários transfiram dinheiro entre contas.
- O sistema deve permitir que os usuários paguem contas.
- O sistema deve permitir que os usuários visualizem o saldo de suas contas.
- O sistema deve permitir que os usuários visualizem o histórico de transações de suas contas.
- O sistema deve permitir que os usuários cancelem suas contas

Requisitos não funcionais:

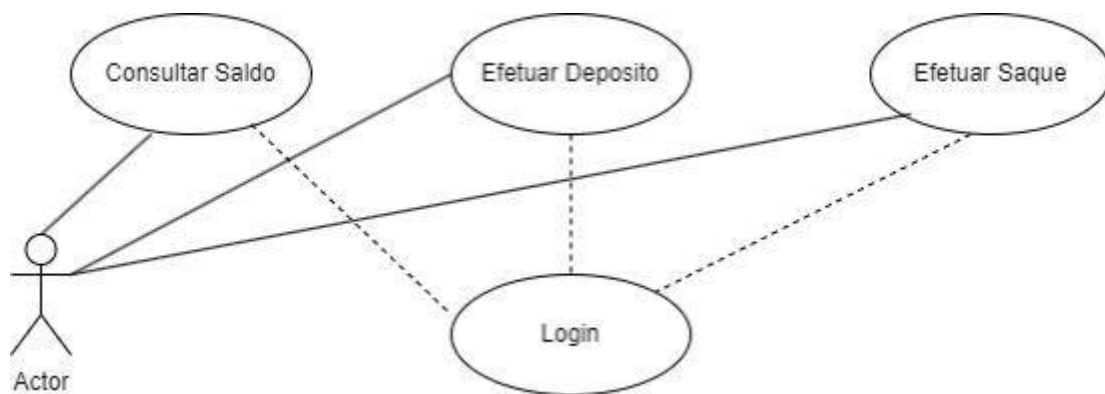
- O sistema deve ser capaz de lidar com um grande número de transações simultâneas sem diminuir o desempenho.
- O sistema deve ser capaz de lidar com picos de tráfego sem diminuir o desempenho.
- O sistema deve ser seguro e protegido contra acesso não autorizado.
- O sistema deve estar disponível 24 horas por dia, 7 dias por semana, sem tempo de paragem, se possível.
- O sistema deve ser fácil de usar, acessível e intuitivo para todos os usuários.
- O sistema deve ser compatível com todos os navegadores e dispositivos móveis populares.
- O sistema deve ser capaz de lidar com grandes quantidades de dados sem diminuir o desempenho.
- O sistema deve ser escalável e capaz de crescer à medida que a base de usuários cresce.

Regras de negócio:

- O banco deve manter o sigilo das informações dos clientes.
- O banco deve cumprir as leis e regulamentações aplicáveis.
- O banco deve fornecer informações precisas e atualizadas aos clientes.
- O banco deve fornecer serviços de qualidade aos clientes.
- O banco deve ter um sistema de segurança robusto para proteger os dados dos clientes.
- Apenas maiores de 18 anos podem abrir uma conta na plataforma
- O banco deve ter um sistema de backup para garantir a disponibilidade dos dados dos clientes.
- O banco deve ter um sistema de recuperação de desastres para garantir a continuidade dos serviços em caso de falha do sistema.
- O banco deve ter um sistema de gerenciamento de riscos para minimizar os riscos associados às atividades bancárias.
- O banco deve ter um sistema de gerenciamento de fraude para detectar e prevenir fraudes.

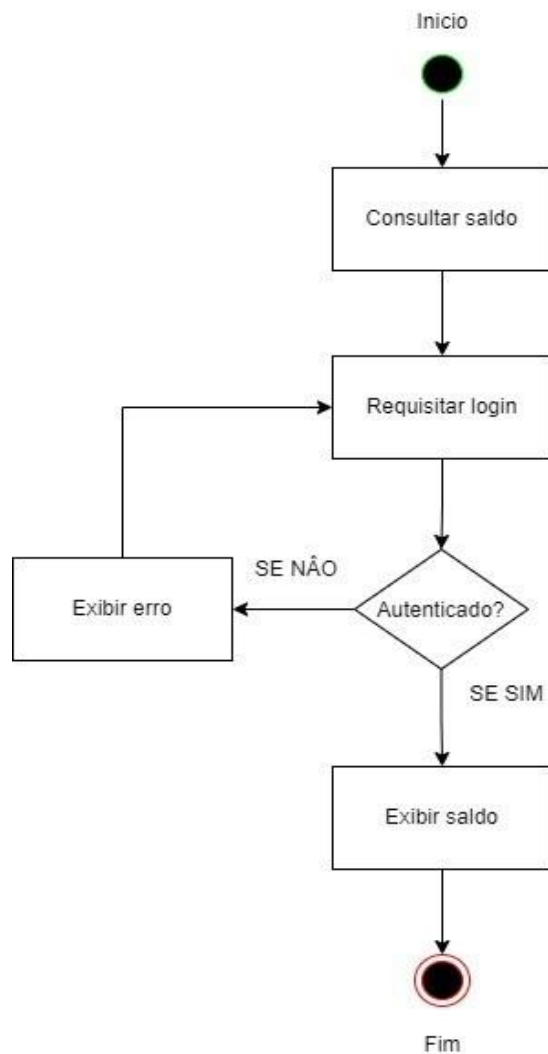
Construção de diagramas UML:

Diagramas de caso de uso:



Representam as funções em que o usuário poderá interagir no sistema com efetuar depósito, saque, transferências e pagamentos.

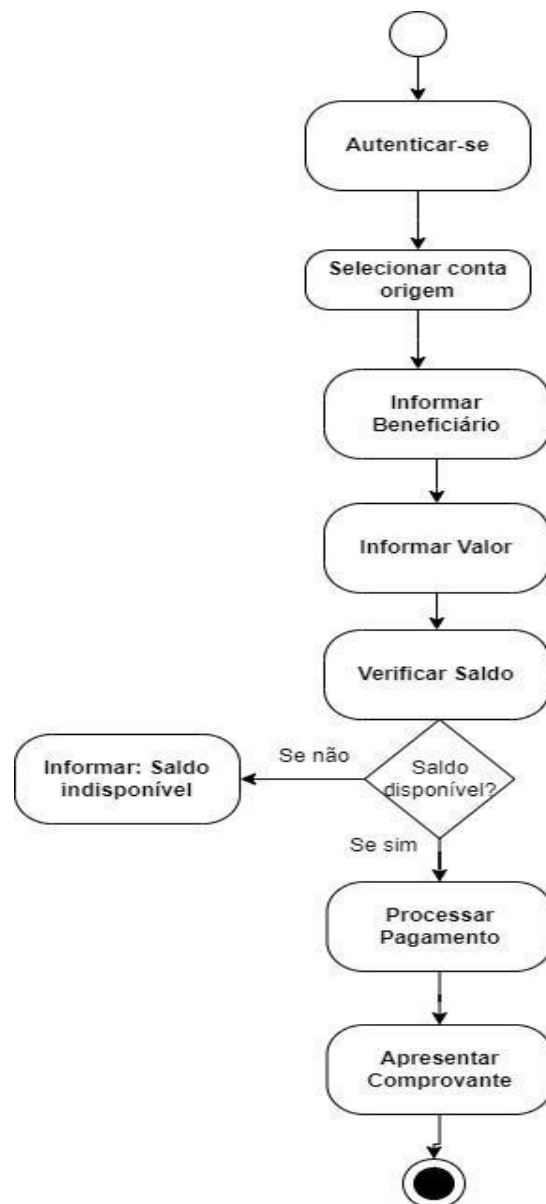
Diagramas de atividades - Consultar saldo:



O cliente opta por consultar o saldo da sua conta, onde seleciona o que deseja, o programa requisita login, caso não esteja realizado, se for autenticado ele prosseguirá e será exibido o saldo.

Caso erre a autenticação será exibida uma mensagem de erro e voltará para o que login seja feito novamente.

Diagramas de atividades – Realizar pagamentos:



O programa solicitará a autenticação, logo após ser autenticado, o usuário entrará na conta de origem, informará o beneficiário e o valor do pagamento.

O sistema irá verificar o saldo e se estiver disponível processará o pagamento e apresentará o comprovante.

Diagramas de atividades – Realizar transferência:



Logo após a autenticação e selecionar a conta de origem, o cliente escolherá a forma de transferências disponível, entre PIX, TED e DOC. Irá fazer a escolha da conta que destinará a transferência, informará o valor que irá transferir, o programa verificará o saldo e se aprovado, processará e emitirá o comprovante de transferência.

Diagramas de atividades – Consultar extrato:



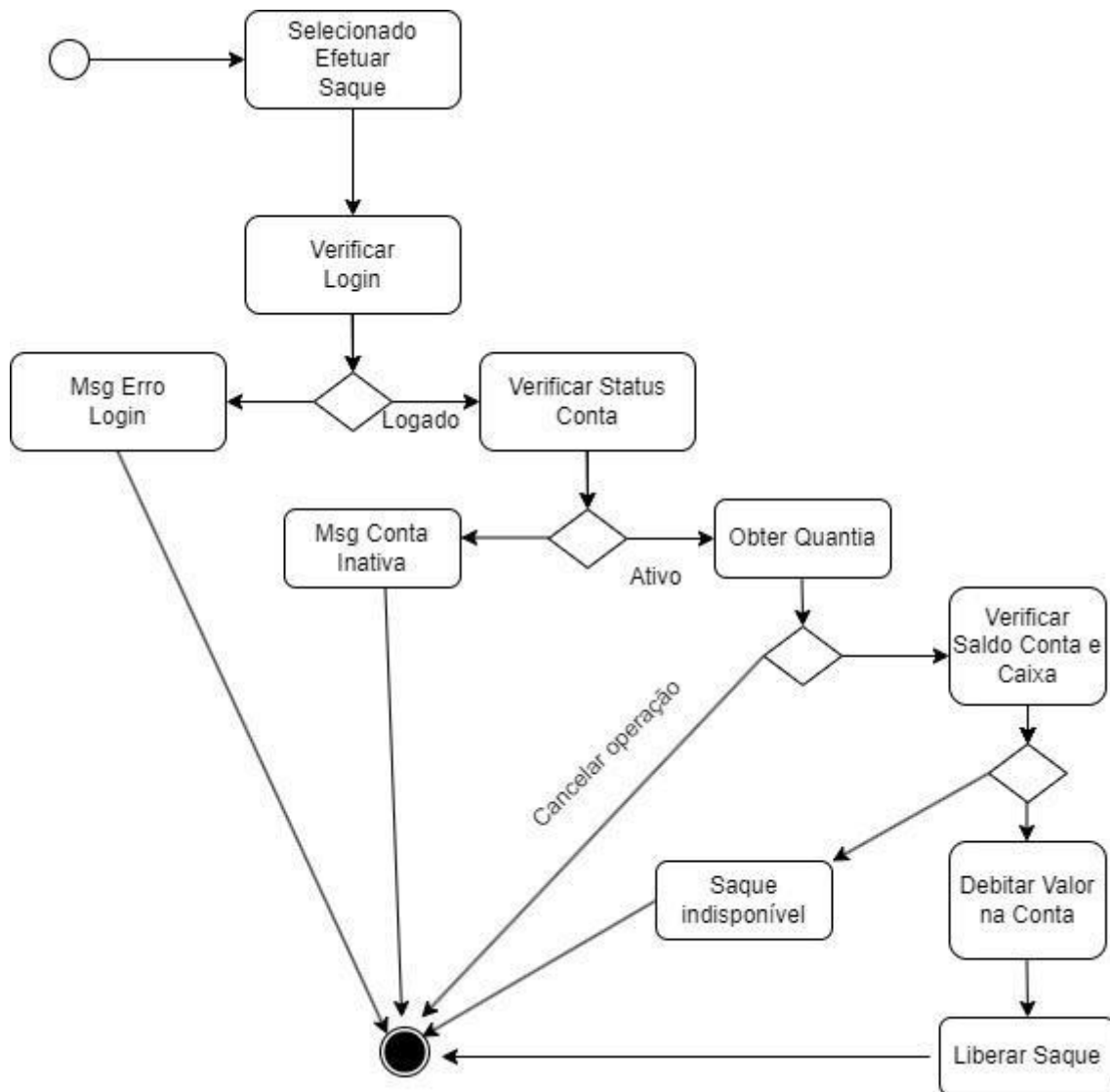
O Programa solicitará a autenticação, o usuário seleciona “Solicitar Extrato”, e o programa irá processar a solicitação, e depois será gerado o extrato e apresentado ao cliente.

Diagramas de atividades – Realizar depósito:



Será selecionado a opção “Deposito”, e o programa solicitará o valor do depósito, e verificará o valor, se disponível será validado, e apresentará a Agência e a Conta para qual será depositado, se não houver erro o programa prosseguirá, e a mensagem e o comprovante serão realizados.

Diagramas de atividades – Realizar saque:

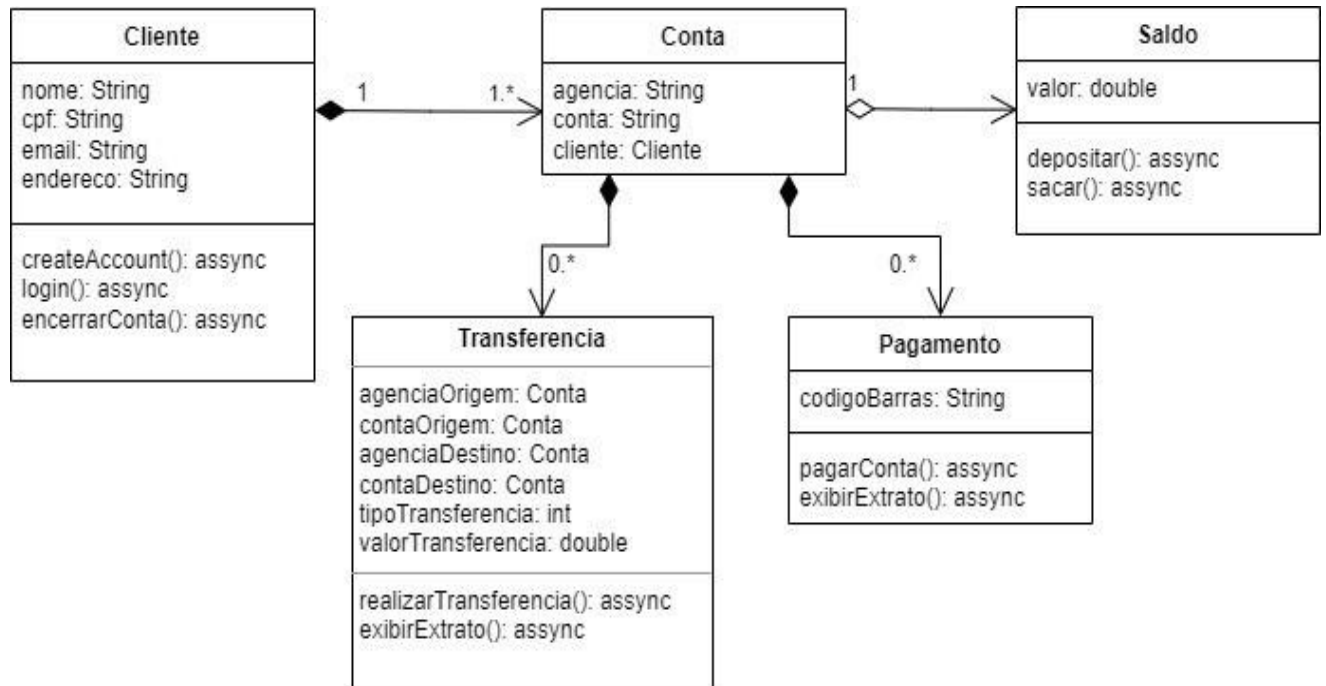


Selecionado “Saque”, irá verificar login, caso esteja certo prosseguira, caso não esteja exibira uma mensagem de erro.

Se o programa continuar, irá verificar o status da conta e irá obter a quantia e será verificado o saldo, caso seja indisponível exibirá uma mensagem de erro.

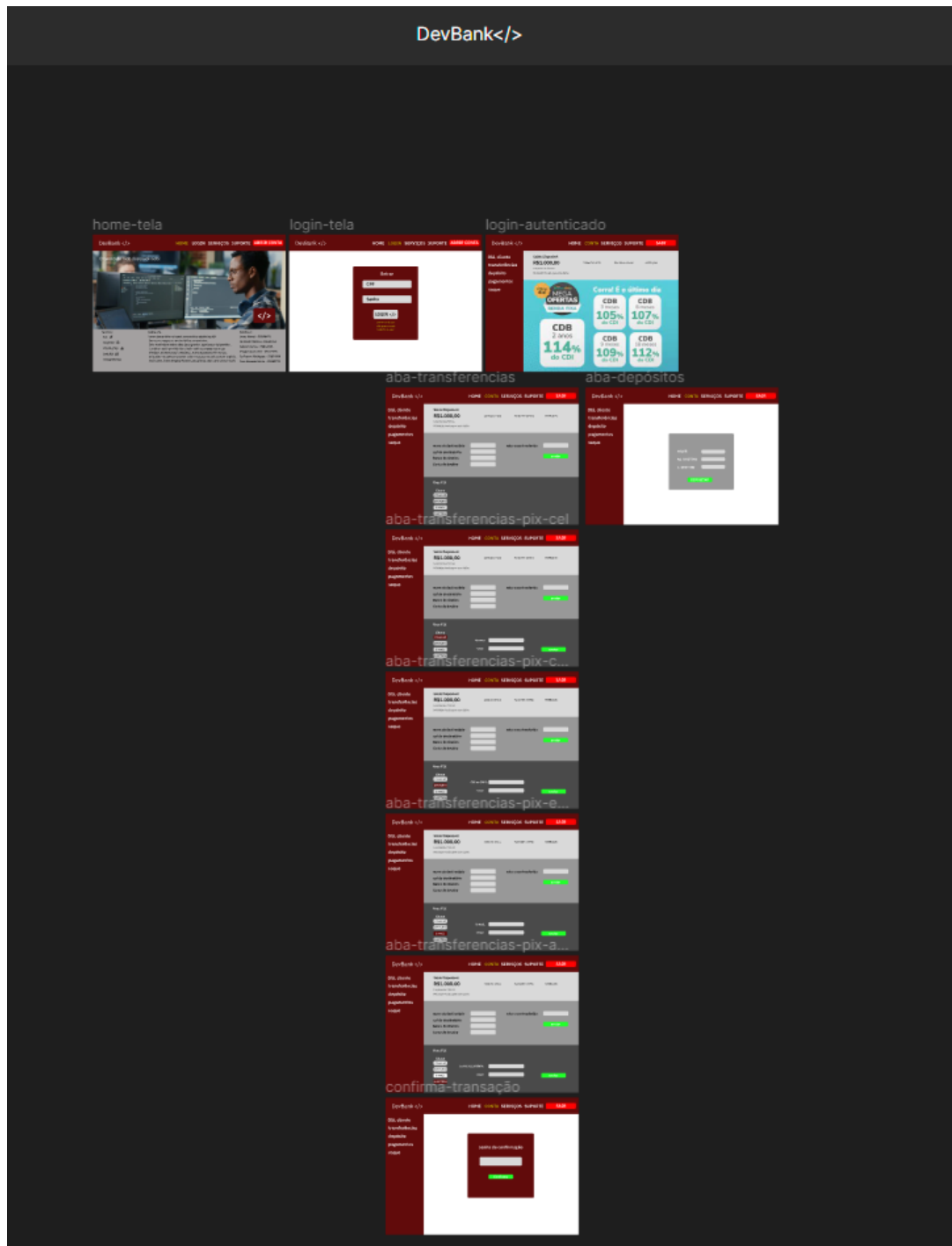
Se não, irá debitar da conta o valor e o saque serão liberados.

Diagramas de classes:



Diagramas de classes, onde é realizado e apresentado a classificação dos objetos cliente, conta, saldo, transferência e pagamento.

Construção do Protótipo - Figma:



Foi criado um protótipo para uma simulação e pré-visualização do sistema antes de sua criação através da ferramenta Figma:

<https://www.figma.com/proto/u81HCyY1GtMfrT0q0dMdHW/DevBank%3C%2F%3E?type=design&node-id=0-1&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=1%3A2>

Construção do Protótipo - Figma:



Aqui é apresentado a página principal do banco digital Dev Bank, onde é apresentado os seguintes links: Login, abrir conta, serviços e suporte.

Construção do Protótipo - Figma:



Menu principal após o cliente realizar o login em sua conta. Podemos verificar o saldo, consultar extrato, atividades recentes e as notificações.

Ao lado superior esquerdo encontra-se os seguintes links para os serviços de: Transferências, Depósito, Pagamento e Saque.

Construção do Protótipo - Figma:

DevBank </>

HOME **CONTA** SERVIÇOS SUPORTE

SAIR

Olá, cliente

transferências

depósito

pagamentos

saque

Saldo Disponível

R\$1.000,00

Consultar Extrato

Atividades recentes

Notificações

Lançamentos futuros:

R\$ 350,84 Previsto para o dia 20/04

nome do destinatário

cpf do destinatário

Banco de destino

Conta de destino

valor a ser transferido

enviar

Àrea PIX

Chave

CELULAR

CPF/CNPJ

E-MAIL

ALEATÓRIA

Acima temos a aba de transferências, onde o cliente fará suas transferências sejam via TED/DOC ou PIX através da área PIX inserindo a chave desejada.

Construção do Protótipo - Figma:

The image shows a Figma prototype of a banking application interface. It features a dark red header bar at the top. On the left side of the header, the text 'DevBank </>' is displayed. On the right side, there are navigation links: 'HOME', 'CONTA' (highlighted in yellow), 'SERVIÇOS', 'SUPORTE', and a red button labeled 'SAIR'. Below the header, on the left, is a dark red sidebar containing the text 'Olá, cliente' followed by a list of menu items: 'transferências', 'depósito' (highlighted in white), 'pagamentos', and 'saque'. The main content area is white and contains a gray rectangular box representing a deposit form. This form has three input fields labeled 'VALOR', 'AG. DESTINO', and 'C. DESTINO', each followed by a light gray input field. Below these fields is a green button labeled 'DEPOSITAR'.

Acima temos a aba de deposito no qual o cliente irá inserir os dados e valor na conta que deseja depositar.

Codificação do projeto:

O projeto todo foi codificado em HTML5, CSS3 e JavaScript com a utilização do framework Vue.JS para o desenvolvimento.

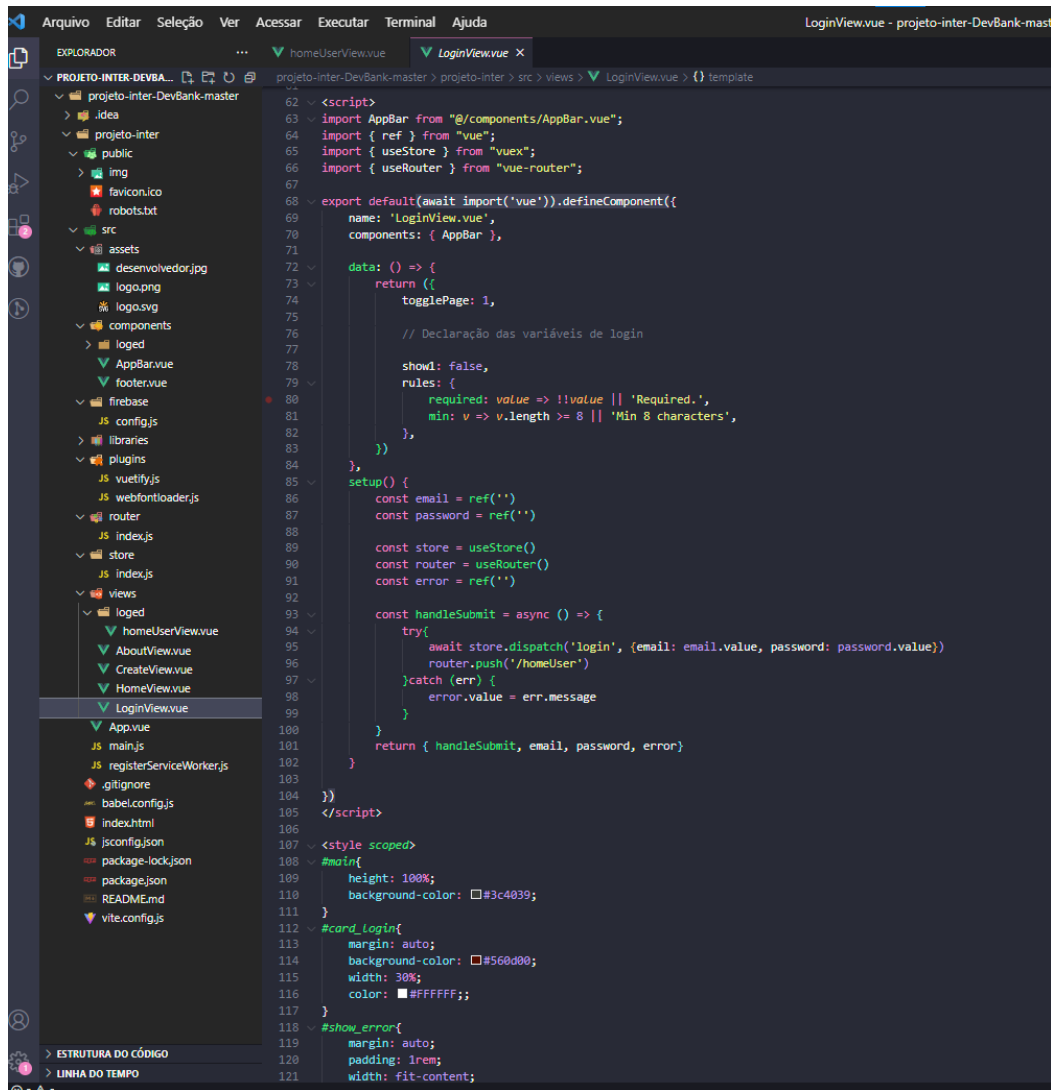
Tela para criação de conta :

The image shows a Visual Studio Code editor with a project named 'projeto-inter-DevBank-master'. The Explorer sidebar on the left displays the project structure, including folders like 'public', 'src', 'assets', 'components', 'firebase', 'plugins', 'router', 'store', 'views', and 'logged', along with various JavaScript and JSON files. The main editor area shows the 'CreateView.vue' file, which is a Vue component template. The template includes a root element with an 'id' of 'main', a container, a card with an 'id' of 'card_create' and 'align' of 'center', and a form with a submit button. The form contains four text input fields with labels 'Nome completo', 'E-mail', 'CPF', and 'Senha', each with specific validation rules and a password hint. The template also includes a card title 'Abrir conta' and a subtitle.

```

1 <template>
2   <AppBar/>
3   <v-main
4     id="main"
5   >
6     <v-container></v-container>
7     <v-card
8       id="card_create"
9       align="center"
10    >
11       <v-card-title>
12         Abrir conta
13       <v-card-subtitle>
14         <hr>
15       </v-card-subtitle>
16     </v-card-title>
17     <v-form @submit.prevent="handleSubmit">
18       <v-container>
19         <v-card-item>
20           <v-text-field
21             v-model="fullName"
22             :rules="[rules.required]"
23             Label="Nome completo"
24             variant="underlined"
25           ></v-text-field>
26           <v-spacer></v-spacer>
27           <v-text-field
28             v-model="email"
29             type="email"
30             :rules="[rules]"
31             Label="E-mail"
32             variant="underlined"
33           ></v-text-field>
34           <v-spacer></v-spacer>
35           <v-text-field
36             v-model="cpf"
37             Label="CPF"
38             variant="underlined"
39             :rules="[rules.required]"
40             v-mask="[ '###.###.###-##' ]"
41           ></v-text-field>
42           <v-text-field
43             v-model="password"
44             :append-icon="show1 ? 'mdi-eye' : 'mdi-eye-off'"
45             :rules="[rules.required, rules.min]"
46             :type="show1 ? 'text' : 'password'"
47             variant="underlined"
48             Label="Senha"
49             hint="At least 8 characters"

```



```
62 <script>
63 import AppBar from "@components/AppBar.vue";
64 import { ref } from "vue";
65 import { useStore } from "vuex";
66 import { useRouter } from "vue-router";
67
68 export default(await import('vue')).defineComponent({
69   name: 'LoginView.vue',
70   components: { AppBar },
71
72   data: () => {
73     return {
74       togglePage: 1,
75
76       // Declaração das variáveis de login
77       show1: false,
78       rules: {
79         required: value => !!value || 'Required.',
80         min: v => v.length >= 8 || 'Min 8 characters',
81       },
82     })
83   },
84   setup() {
85     const email = ref('')
86     const password = ref('')
87
88     const store = useStore()
89     const router = useRouter()
90     const error = ref('')
91
92     const handleSubmit = async () => {
93       try{
94         await store.dispatch('login', {email: email.value, password: password.value})
95         router.push("/homeUser")
96       }catch (err) {
97         error.value = err.message
98       }
99     }
100     return { handleSubmit, email, password, error}
101   }
102 </script>
103
104 <style scoped>
105 #main{
106   height: 100%;
107   background-color: #3c4039;
108 }
109 #card_login{
110   margin: auto;
111   background-color: #560d00;
112   width: 30%;
113   color: #FFFFFF;
114 }
115 #show_error{
116   margin: auto;
117   padding: 1rem;
118   width: fit-content;
119 }
```

Codificação da tela de login para que o cliente possa acessar os serviços disponíveis no banco digital Dev Bank.

3 CONSIDERAÇÃO FINAIS

O projeto foi feito em etapas onde, foi realizada a programação com uma dificuldade mediana, pois os conhecimentos adquiridos em sala foram de suma importância. Já a parte escrita foi mais delicada pois houve um pouco de conflito de ideias, assim sendo, gerou uma certa dificuldade, mas a equipe chegou em comum acordo conseguindo concluir o projeto de forma louvável.

4 BIBLIOGRAFIA

Material didático apresentado nas aulas das seguintes matérias: Engenharia de Software e Análise e Projetos de Sistemas.

APENSO 1 – CRONOGRAMA DE ENTREGA DE ATIVIDADES.

