



Calculadora Anônima de salário

Brunno
Gabriel C.
Gabriel S.

S u m á r i o

- **Contextualização;**
- **Implementação - Original;**
- **Implementação - Aplicada;**
- **Aprendizados;**
- **Motivo do nosso orgulho;**
- **Otimizações;**
- **Apresentação do Código;**

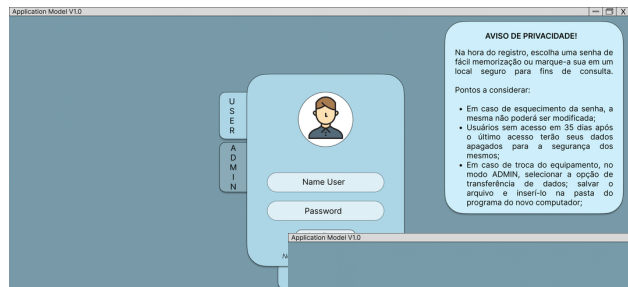


Contextualização

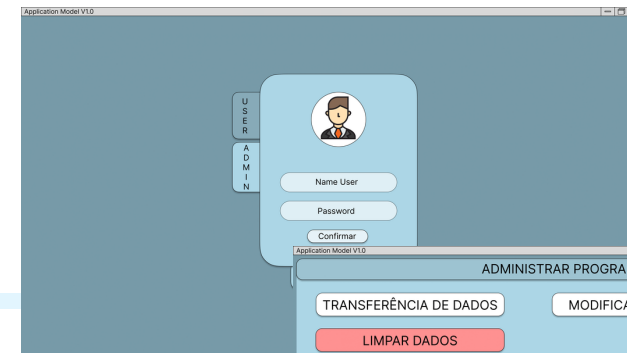
O projeto consiste em um aplicativo que simplifique o processo de cálculo da folha de pagamento para trabalhadores de forma intuitiva e acessível que permitam um fácil entendimento dos descontos e benefícios.



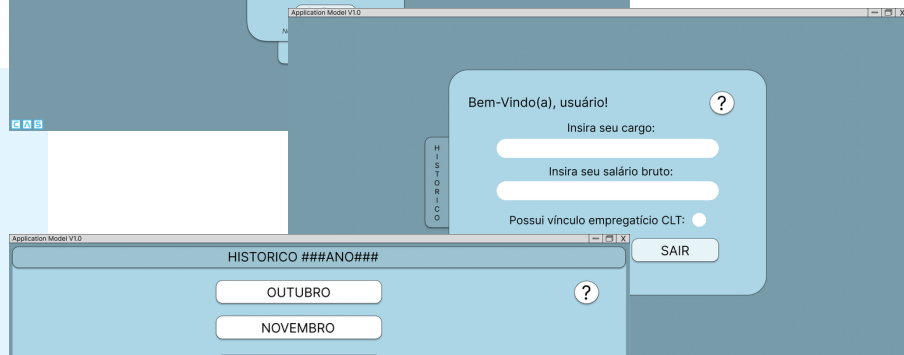
Implementação - Original



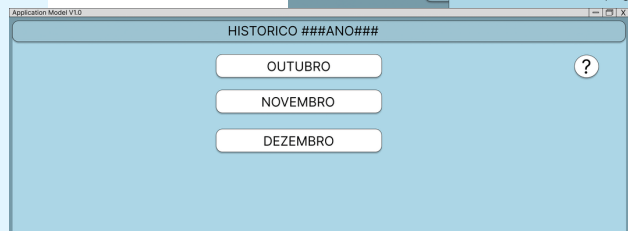
Tela de login do usuário



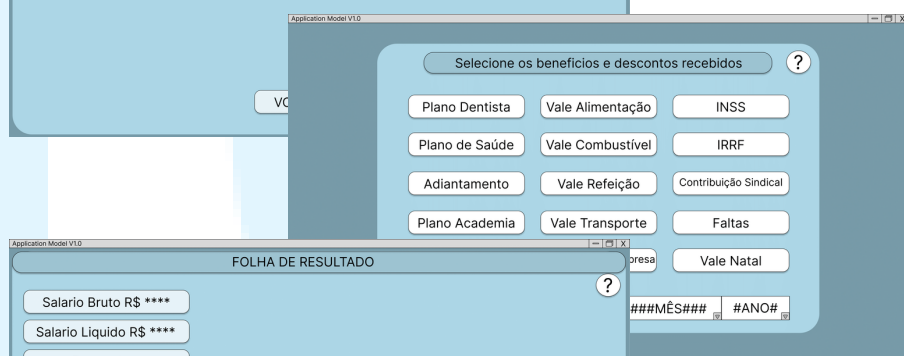
Tela de login do admin



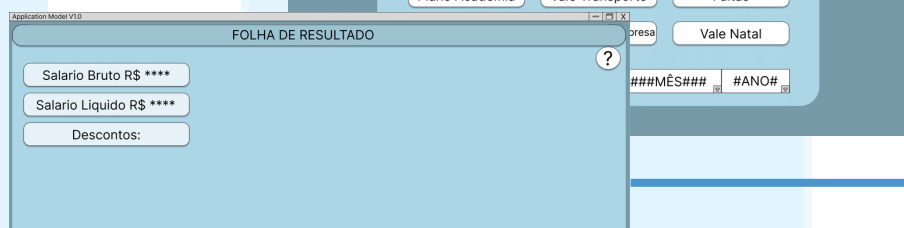
Tela de inserção do salário e **histórico**



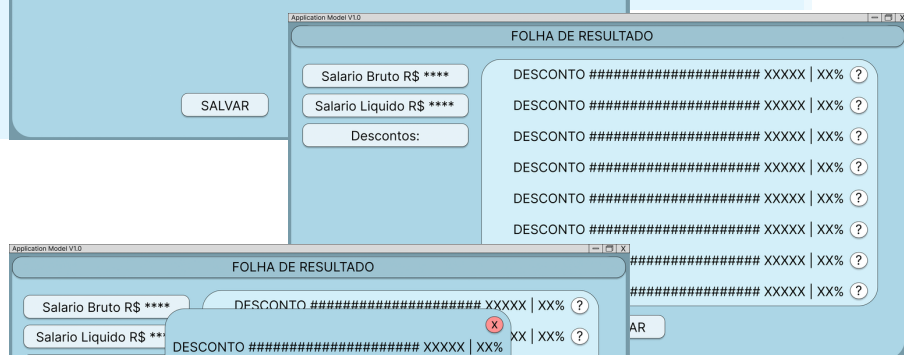
Histórico de salários por meses



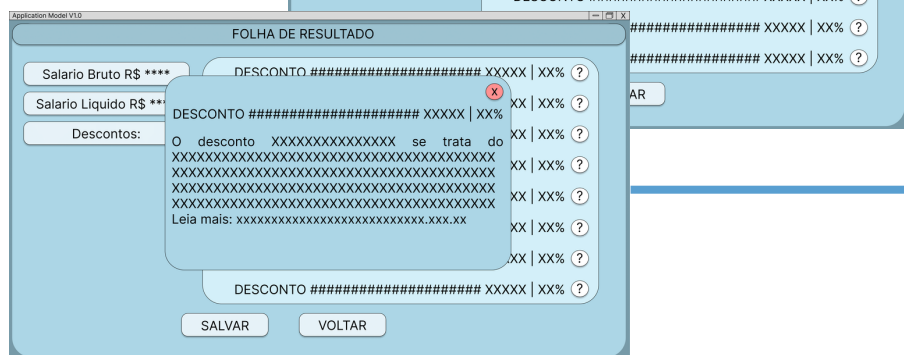
Seleção de descontos



Tela de salários bruto, líquido e descontos



Tela de listagem dos desconto selecionados



Tela de explicação do desconto selecionado



Edição dos descontos

Deletar usuários

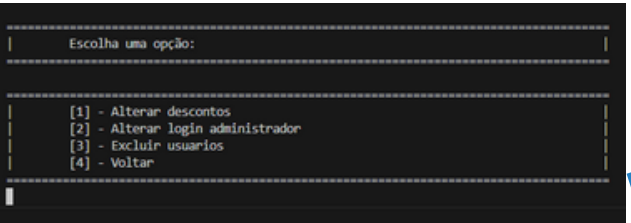
Limpeza de dados

Por terem sido considerados apenas os descontos básicos, não houve necessidade da criação de uma seleção de descontos; mas ainda com a possibilidade de selecionar um desconto: o Vale Transporte.

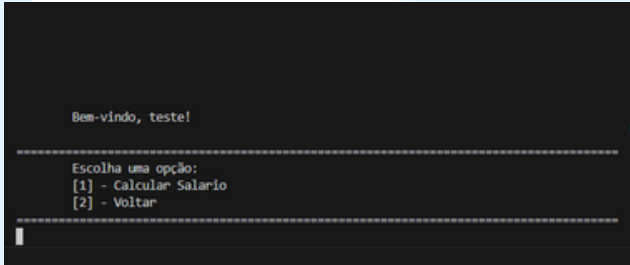
Implementação - Aplicada



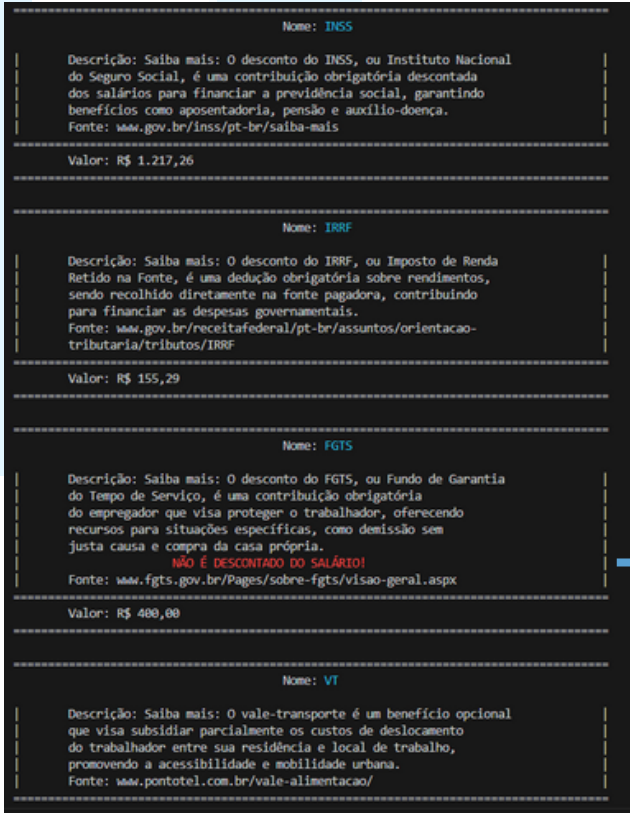
Menu inicial com logins e cadastro



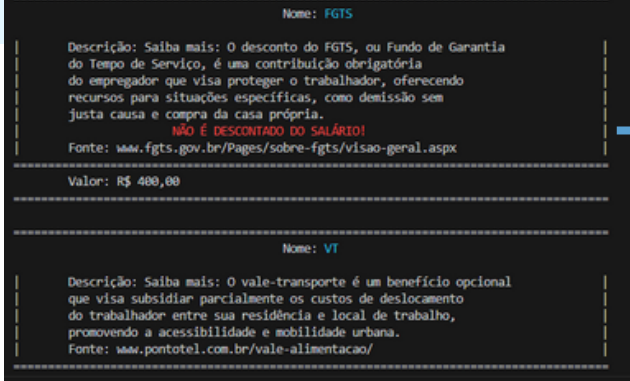
Menu de admin



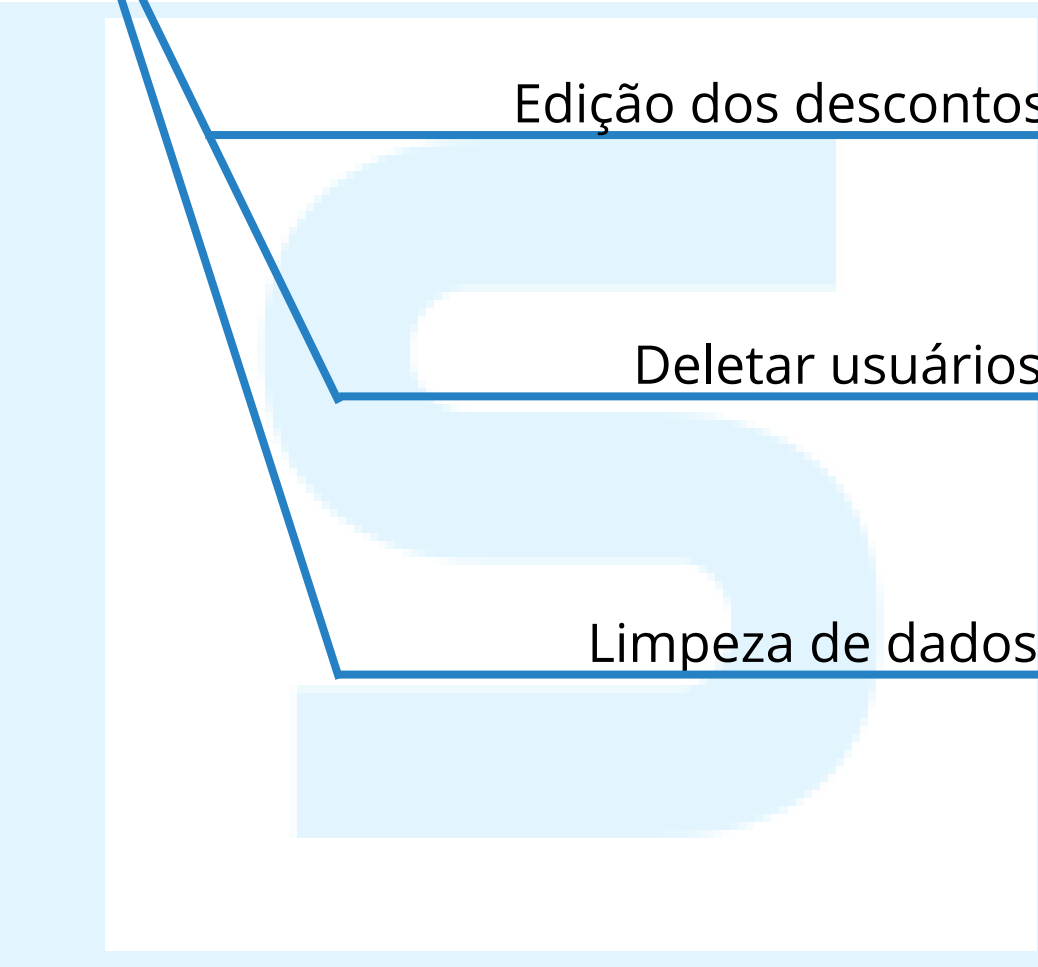
Tela de inserção do salário



Tela de salários bruto, líquido e descontos



Tela de explicação dos descontos básicos



Edição dos descontos

Deletar usuários

Limpeza de dados

Aprendizados

Hard Skills:

- Aperfeiçoamento na lógica de programação em Java;
- Melhoria na compreensão de POO;
- Descobertas de novas ferramentas.



Soft Skills:

- Organização;
- Comunicação;
- Cooperatividade.

Motivo do nosso orgulho:

```
✓ public class LimpaConsole {  
✓  
    public static void limparTela(){  
        for(int clear = 0; clear < 1000; clear++) {  
            System.out.println("\b") ;  
        }  
    }  
}
```

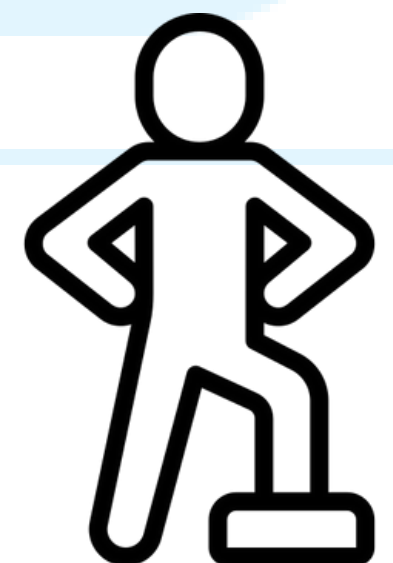
Brunno

```
case 3:  
    // Login Admin  
    LimpaConsole.limparTela();  
    usuarioAuxiliarString = EntradaSaida.solicitarDados("o usuario");  
    senhaAuxiliarString = EntradaSaida.solicitarDados("a senha");  
    LimpaConsole.limparTela();  
    usuarioExistente = admin.verificarAdmin(usuarioAuxiliarString, senhaAuxiliarString);  
    wait(1000);  
    LimpaConsole.limparTela();  
    if (usuarioExistente) {
```

Gabriel C.

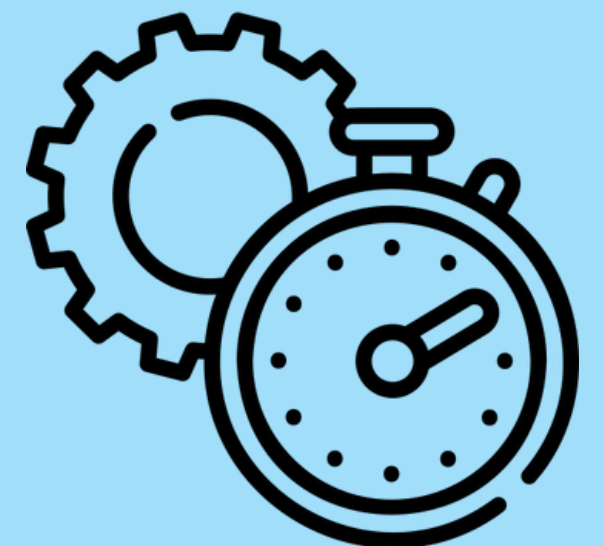
```
do{  
    try{  
        Principal.wait(1000);  
        LimpaConsole.limparTela();  
        System.out.println("\n=====")  
        "        Digite o novo valor do desconto: (Utilize virgula ',,')" +  
        "\n=====")  
        Scanner entradaDez = new Scanner(System.in);  
        entrada = entradaDez.nextDouble();  
    }  
}
```

Gabriel S.



Otimizações

- Recomeços - Melhoria na organização;
- Reelaboração da lógica - (admin, descontos, ...);
- Análise da ideia de implementação;
- Economia de linhas - Recursividade





Dúvidas

???

Obrigado!