**-------------------------------------------------------------------------------------------**

**Planejamento e Arquitetura de um Sistema Desktop com Java**

**Aluno(a): Erik Eike Silva de Paiva**

**Data de entrega: 08/08/2025**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Nome do Projeto e Cenário Escolhido**
2. Pesquisa sobre Java Desktop, Swing e JavaFX

**2.1** Java Desktop

**2.2** Swing

**2.3** JavaFX

**2.4** Vantagens do Java

**2.5**Desvantagens do Java

1. Planejamento da Interface Gráfica (GUI)
2. Diagrama de Classes
3. Protótipo da Interface
4. Ciclo de Vida do Desenvolvimento de Software (SDLC)
5. Considerações Finais
6. **Nome do Projeto e Cenário Escolhido**

**Projeto: Sistema de Cadastro de Clientes**

Descrição: Aplicação para cadastrar e gerenciar clientes, incluindo informações como nome, endereço, telefone e histórico de compras.

1. Pesquisa sobre Java Desktop, Swing e JavaFX
   1. Java Desktop

Tecnologia utilizada para criar aplicações gráficas simples e estáveis, amplamente usada no passado para softwares locais.

* 1. Swing

Biblioteca gráfica do Java que permite criar janelas, botões, tabelas e outros elementos visuais. É mais antiga, mas ainda funcional e compatível com versões atuais do Java.

3.3 JavaFX

Biblioteca mais moderna e visualmente mais rica que o Swing, com suporte para estilos CSS, efeitos gráficos e animações.

* 1. Vantagens do Java

Multiplataforma (Windows, Linux e Mac).

Extensa biblioteca de recursos prontos.

Linguagem orientada a objetos, o que melhora a organização.

3.5 Desvantagens do Java

Pode ser mais pesado em máquinas antigas.

Swing pode gerar interfaces com visual ultrapassado.

1. Planejamento da Interface Gráfica (GUI)

Classe Cliente

Atributos: nome, endereço, telefone, históricoDeCompras.

Métodos: cadastrar(), atualizar(), excluir(), pesquisar().

Classe SistemaCadastro

Atributos: listaDeClientes.

Métodos: adicionarCliente(), editarCliente(), removerCliente(), buscarCliente().

1. Diagrama de Classes (UML Simplificado)

|  |
| --- |
|  |

| Cliente |

|  |
| --- |
|  |

| - nome |

| - endereco |

| - telefone |

| - historico |

|  |
| --- |
|  |

| + cadastrar() |

| + atualizar() |

| + excluir() |

| + pesquisar() |

|  |
| --- |
|  |

▲

|

|  |
| --- |
|  |

| SistemaCadastro |

|  |
| --- |
|  |

| - listaDeClientes |

|  |
| --- |
|  |

| + adicionarCliente() |

| + editarCliente() |

| + removerCliente() |

| + buscarCliente() |

|  |
| --- |
|  |

1. Protótipo da Interface (Rascunho Sugerido)

| Sistema de Cadastro de Clientes |

| Nome: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| Endereço: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| Telefone: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| [Cadastrar] [Atualizar] [Excluir] |

| Lista de Clientes: |

|  |
| --- |
|  |

| | Nome | Telefone | Endereço | |

|  |
| --- |
|  |

1. Ciclo de Vida do Desenvolvimento de Software (SDLC)

O SDLC é dividido nas seguintes fases:

1. Análise de Requisitos
2. Design
3. Implementação
4. Testes
5. Implantação
6. Manutenção

Importância do Planejamento:

Evita erros no desenvolvimento.

Economiza tempo.

Melhora a organização.

Garante que o sistema atenda às necessidades do usuário.