**Modelo 1 – Relatório Formal (Word/Google Docs)**

**Título**: Relatório – Sistema de Cadastro de Clientes para Loja

**1. Dados Gerais**

* **Nome do projeto**: Sistema de Cadastro de Clientes para Loja
* **Cenário escolhido**: Sistema de Cadastro de Clientes
* **Integrante(s)**: [Seu nome ou dupla]

**2. Pesquisa sobre Java Desktop**

* **Java Desktop**: Aplicações Java com interface gráfica.
* **Java Swing**: Biblioteca tradicional do Java para GUIs; fácil, mas interface menos moderna.
* **JavaFX**: Framework mais moderno; suporte a gráficos enriquecidos, animações e estilos CSS.
* **Diferenças principais**: Swing é consolidado e leve, JavaFX é visualmente superior e mais flexível.
* **Exemplos de aplicações**: sistemas de estoque, PDV, agendas, calculadoras.
* **Vantagens**: multiplataforma, comunidade forte, integração com bancos de dados.
* **Desvantagens**: Swing parece antiquado; JavaFX pode demandar bibliotecas extras.

**3. Planejamento da Interface (GUI)**

* **Layout descrito**: janela “Cadastro de Clientes” com campos (Nome, Endereço, Telefone, E-mail, CPF), botões (Cadastrar, Atualizar, Excluir, Listar) e tabela de exibição.
* **Usabilidade**: botões abaixo dos campos; tabela ocupa metade da janela; confirmações visuais após ações.
* *(Adicionar foto do desenho em papel aqui)*

**4. Diagrama de Classes**

* **Cliente**
  + Atributos: nome, endereço, telefone, email, cpf
  + Métodos: cadastrar(), atualizar(), excluir(), listar()
* **SistemaCadastro**
  + Atributo: listaClientes (ArrayList<Cliente>)
  + Métodos: adicionarCliente(Cliente), removerCliente(String cpf), buscarCliente(String cpf), exibirTodos()

**5. SDLC (Ciclo de Vida de Desenvolvimento de Software)**

* **Fases**: análise de requisitos, design (planejamento), implementação, testes, implantação.
* **Nesta atividade**: estamos na fase de “Design (Planejamento)”.
* **Importância**: evita retrabalho, guia a implementação e assegura que o sistema atenderá ao que foi definido.

**6. Conclusão / Reflexão**

* Esta atividade reforça que o desenvolvimento não é só codificar — o planejamento (especificação, GUI, classes) é essencial.
* Ajuda a pensar na usabilidade, arquitetura e estruturação antes de programar, economizando tempo e erros.