Especificação de Requisitos do Sistema de Gerenciamento de Usuários

1. Introdução

Este documento descreve os requisitos necessários para o desenvolvimento de um sistema de gerenciamento de usuários, utilizando Programação Orientada a Objetos (POO) e o padrão arquitetônico Model-View-Controller (MVC) em Java.

O objetivo é fornecer uma visão clara das funcionalidades, características e restrições do sistema, garantindo alinhamento entre todos os envolvidos no projeto.

2. Objetivos

• Permitir o cadastro, edição, exclusão e listagem de usuários.

• Garantir segurança no acesso e manipulação dos dados.

• Oferecer uma interface amigável e de fácil uso.

• Estruturar o código de forma modular e escalável.

3. Escopo

O sistema abrangerá:

• Cadastro de novos usuários.

• Edição e exclusão de usuários existentes.

• Listagem de todos os usuários cadastrados.

• Autenticação de usuários via login e senha.

• Armazenamento dos dados em banco de dados relacional via JDBC.

Não estão incluídas:

• Integrações com redes sociais.

• Funcionalidades de recuperação de senha por e-mail.

• Aplicativos móveis.

4. Requisitos Funcionais (RF)

• RF01 – O sistema deve permitir o cadastro de novos usuários.

• RF02 – O sistema deve permitir a edição dos dados de um usuário.

• RF03 – O sistema deve permitir a exclusão de usuários.

• RF04 – O sistema deve listar todos os usuários cadastrados.

• RF05 – O sistema deve autenticar usuários com login e senha.

• RF06 – O sistema deve armazenar e recuperar dados de um banco de dados via JDBC.

5. Requisitos Não Funcionais (RNF)

• RNF01 – O sistema deve ser desenvolvido em Java, utilizando POO e MVC.

• RNF02 – O sistema deve ter interface responsiva e intuitiva.

• RNF03 – O sistema deve garantir segurança na autenticação e no armazenamento de dados.

• RNF04 – O sistema deve ser modular e de fácil manutenção.

• RNF05 – O sistema deve responder às ações do usuário em até 2 segundos.

6. Glossário

• POO – Programação Orientada a Objetos.

• MVC – Model-View-Controller.

• CRUD – Create, Read, Update, Delete (Criar, Ler, Atualizar, Excluir).

• JDBC – Java Database Connectivity, API para conexão com banco de dados.

7. Vídeos

• (Espaço reservado para links de vídeos explicativos ou tutoriais do sistema)

8. Arquivos / Links

• (Espaço reservado para anexos, scripts SQL, documentação técnica)

Perfeito, Jefferson — vou montar para você um **documento estruturado** com tudo que já revisamos até agora, no formato de **esqueleto de documentação**.  
A ideia é que ele já sirva como base para a entrega final da A3, e você só precise ir completando as partes que ainda faltam.

**📄 Documentação do Projeto – Sistema de Gestão de Projetos e Equipes**

**1. Capa**

* **Título:** Sistema de Gestão de Projetos e Equipes
* **Disciplina:** Programação de Soluções Computacionais
* **Avaliação:** A3
* **Integrantes:**
  + Jefferson (Líder técnico e desenvolvedor)
  + Maria Liliane Vasconcelos Barros (Auxiliar de desenvolvimento e documentação)
* **Data:** Setembro/2025

**2. Sumário**

1. Introdução
2. Contexto da UC
3. Apresentação da Proposta
4. Definição do Escopo
5. Formação do Grupo e Análise da Proposta
6. Requisitos do Sistema
7. Tecnologias Utilizadas
8. Aplicação do Paradigma Orientado a Objetos
9. Plano de Ação e Cronograma
10. Modelagem do Sistema
    * Diagrama de Classes UML
    * Diagrama Entidade-Relacionamento (ER)
11. Impacto Humano
12. Critérios de Avaliação da A3
13. Conclusão e Próximos Passos

**3. Introdução**

Breve apresentação do objetivo do projeto e sua importância para a disciplina, destacando que será desenvolvido em **Java (POO + MVC)** com persistência em **MySQL**.

**4. Contexto da UC**

* **Metas de compreensão** (resumo das metas máximas e intermediárias da UC).
* **Competências** (identificar, planejar e desenvolver soluções computacionais, organizar dados em banco, etc.).
* **Tópicos geradores** (tipos de dados, paradigmas, lógica de programação, POO, SGBDs, interface gráfica).

**5. Apresentação da Proposta**

* **Caso:** Sistema de Gestão de Projetos e Equipes para clientes Oracle.
* **Problema:** Dificuldade no controle de múltiplos projetos e equipes.
* **Solução:** Ferramenta para cadastro, acompanhamento e relatórios de projetos, equipes e tarefas.

**6. Definição do Escopo**

* **Funcionalidades principais:**
  + Cadastro e manutenção de usuários, projetos, equipes e tarefas.
  + Alocação de equipes a projetos.
  + Relatórios e dashboards.
  + Autenticação com perfis de acesso.
* **Requisitos explícitos:** (lista já definida com itens a–p).
* **Requisitos implícitos:** (questões sobre vínculos, permissões, histórico, logs, etc.).

**7. Formação do Grupo e Análise da Proposta**

* **Integrantes e papéis**.
* **Estratégia inicial** (definição de tecnologias, criação de repositório GitHub, divisão de tarefas).

**8. Requisitos do Sistema**

* **Funcionais:** CRUD de usuários, projetos, equipes e tarefas; autenticação; relatórios.
* **Não funcionais:** Java 17+, MVC, segurança, desempenho, usabilidade.

**9. Tecnologias Utilizadas**

* **Linguagem:** Java 17+
* **Interface:** Swing ou JavaFX
* **Banco de Dados:** MySQL
* **Bibliotecas:** JDBC, JUnit
* **Controle de Versão:** Git/GitHub

**10. Aplicação do Paradigma Orientado a Objetos**

* **Identificação de objetos reais:** Usuário, Projeto, Equipe, Tarefa, Relatório.
* **Encapsulamento:** atributos privados, getters/setters.
* **Herança:** classe base Pessoa.
* **Polimorfismo:** métodos com comportamentos diferentes por perfil.
* **Abstração:** interfaces para padronizar operações CRUD.

**11. Plano de Ação e Cronograma**

| **Tarefa** | **Responsável** | **Início** | **Fim** |
| --- | --- | --- | --- |
| Levantar requisitos | Jefferson | 02/09 | 05/09 |
| Modelar classes UML | Maria | 06/09 | 09/09 |
| Implementar Model | Jefferson | 10/09 | 14/09 |
| Implementar Controller | Maria | 15/09 | 19/09 |
| Implementar View | Maria | 20/09 | 26/09 |
| Testes unitários | Jefferson | 27/09 | 02/10 |
| Testes com usuários | Maria | 03/10 | 05/10 |
| Documentação final | Maria | 06/10 | 08/10 |
| Apresentação | Equipe | 09/10 | 09/10 |

**12. Modelagem do Sistema**

* **Diagrama de Classes UML:** representação das classes, atributos, métodos e relacionamentos.
* **Diagrama ER:** entidades, atributos, chaves primárias e estrangeiras, relacionamentos.

**13. Impacto Humano**

Resumo dos impactos para usuários finais, desenvolvedores, analistas, DBA e suporte técnico.

**14. Critérios de Avaliação da A3**

* Valor total: 40 pontos (Etapas 1 a 6).
* Aprovação: média ≥ 70 (A1 + A2 + A3) e frequência ≥ 75%.
* Entregas via DreamShaper, penalidade para atrasos.

**15. Conclusão e Próximos Passos**

* Síntese do que foi desenvolvido até agora.
* Próximas etapas: prototipagem de telas, implementação completa, testes finais, entrega e apresentação.

Se quiser, Jefferson, posso já **preencher esse esqueleto com o texto detalhado de cada seção** usando tudo que já escrevemos, para que você tenha um documento pronto para entrega.  
Quer que eu já faça essa versão completa?