# FACULDADE INTERNACIONAL DA PARAÍBA - FPB

**Análise e Desenvolvimento de Sistemas Italo Gabriel Martins - 1362223191**

# Mirelly Ketilly Melo de Oliveira - 1362220814

**MODELO DE DE REQUISITOS**

# João Pessoa 2024

**PROJETO A3**

Projeto apresentado ao curso Análise e Desenvolvimento de Sistemas, na Unidade Curricular Gestão e Qualidade de Software, para obtenção da nota A3, sob orientação dos professores Pablo Ramon e Reinaldo Carlos.

# João Pessoa 2024 SUMÁRIO

1. **INTRODUÇÃO**

O projeto em questão consiste em um site que possui funcionalidades essenciais como login, senha, cadastro de usuários, exibição de lista de cadastros e exclusão de registros. Todas as funcionalidades foram inovadoras com base nos conhecimentos adquiridos durante o semestre.

Através deste projeto, buscamos aplicar os conceitos de análise de sistemas, arquitetura de software, segurança de dados, entre outros, desenvolvendo um site que atenda aos requisitos funcionais e não funcionais propostos, além de ser intuitivo, seguro e eficiente. Este modelo de requisitos serve como um guia para o desenvolvimento do sistema, detalhando as funcionalidades, fluxos e critérios necessários

A execução deste projeto permitiu consolidar o aprendizado do semestre e fornecer uma visão prática de como planejado, projetar e desenvolver um sistema completo, desde a análise de requisitos até a implementação das funcionalidades. O presente documento é uma das etapas importantes do processo de desenvolvimento e serve para garantir que todas as expectativas e necessidades sejam atendidas.

# Estimativa de Esforço e Planejamento

**Funcionalidades Básicas:**

Cadastro de clientes: Permitir a inserção de novos clientes com informações como nome, email, etc.

Edição de clientes: Possibilitar a alteração dos dados de um cliente já cadastrado.

Listagem de clientes: Apresentar uma lista com todos os clientes cadastrados, permitindo a busca por filtros.

Exclusão de clientes: Permitir a remoção de clientes da base de dados.

# Organização do Trabalho:

Divisão de tarefas: Realizamos boa parte do projeto por ligação, exceto o banco de dados.

Reuniões regulares: Estabelecemos momentos para discutir o progresso, desenvolver, solucionar problemas e tomar decisões.

Ferramentas de colaboração: GitHub e Discord para acompanhar o andamento do

projeto.

# Modelos de Qualidade

Escolha do Modelo:

O MPS.BR devido à sua adaptabilidade a diferentes contextos e ao foco em melhoria contínua. Ele oferece um conjunto de níveis de maturidade que podem ser alcançados gradualmente, permitindo que o projeto evolua de forma controlada.

Implementação:

Definir processos: Estabelecemos processos claros para cada fase do desenvolvimento, como requisitos, projeto, desenvolvimento, testes e implantação.

# Estratégias de Teste

Testes Unitários:

Cobertura de código: Escrever testes unitários para cobrir a maior parte do código, garantindo que cada função ou método funcione corretamente de forma isolada.

Testes de borda: Criação de casos de teste que verifiquem o comportamento do sistema em situações extremas ou inesperadas.

# Evolução do Software

Manutenção:

Documentação: Manter a documentação do sistema atualizada para facilitar futuras alterações.

Modularização: Desenvolvido o sistema de forma modular, facilitando a identificação e a correção de problemas.

Testes de regressão: Execução testes de regressão após cada alteração para garantir que novas funcionalidades não afetem as existentes.

Melhorias:

Novas funcionalidades: Adicionar novas funcionalidades de acordo com as necessidades dos usuários.

Otimização de desempenho: Melhorar a performance do sistema através de técnicas de otimização de código e banco de dados.

Melhoria da interface: Aprimorar a interface do usuário para torná-la mais intuitiva e agradável.

# Controle de Versão

Git:

Criação do repositório: Criar um repositório no GitHub ou em outra plataforma de hospedagem de código.

Commits regulares: Fazer commits frequentes para registrar as alterações no código.

Branching: Utilizar branches para desenvolver novas funcionalidades de forma isolada.

Merge requests: Criar merge requests para integrar as alterações no branch principal.

Configuração:

.gitignore: Configurar o arquivo .gitignore para excluir arquivos temporários e configurações locais.

README: Criar um arquivo README.md com informações sobre o projeto.

Considerações Adicionais:

Linguagem de programação: PHP Banco de dados: MySQL

Segurança: criptografia e validação de dados.