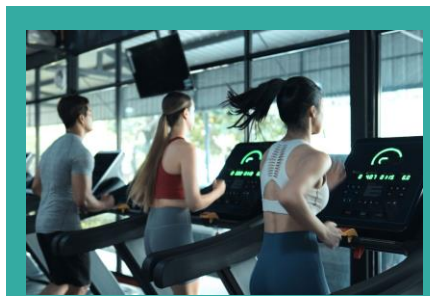


Relatório de elaboração Do Projeto 2025

15/01

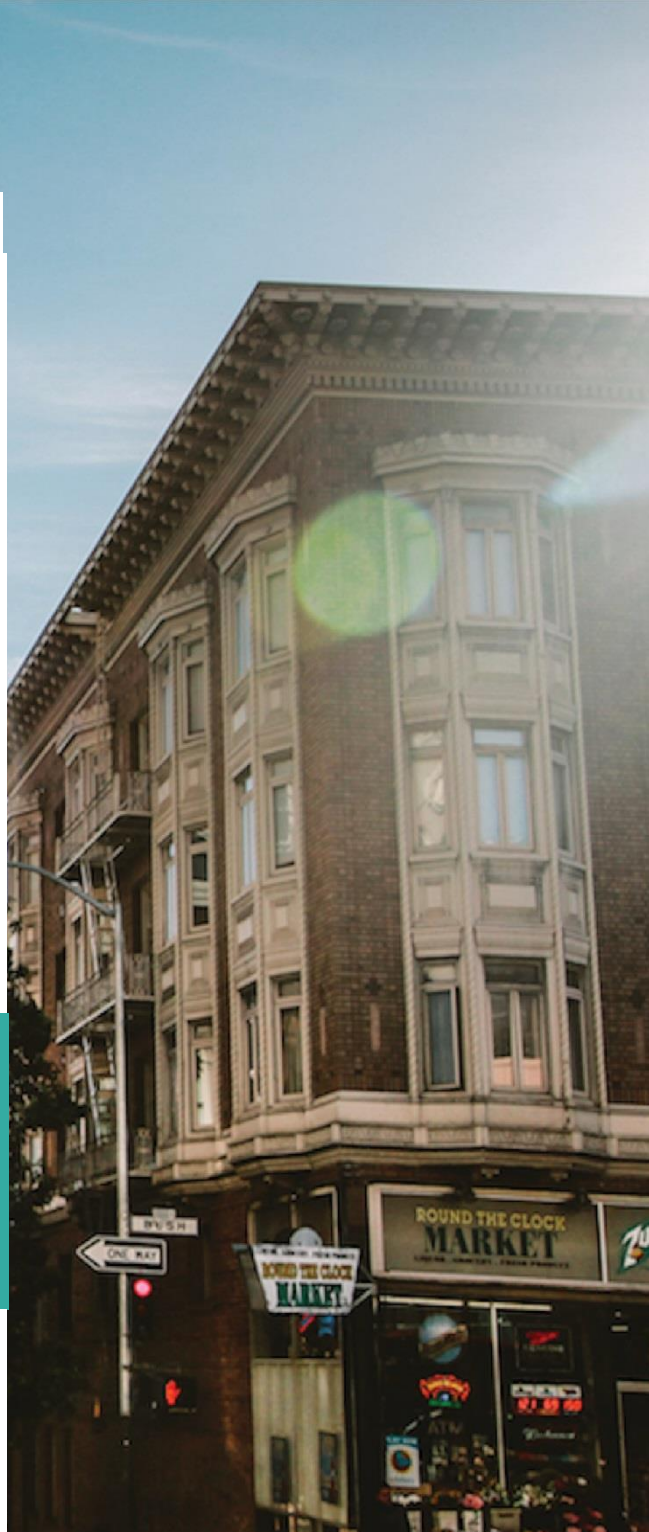
Healthy Life



Da autoria de: Jéssica Rebello Cunha

Sistema de Gerenciamento para Ginásio

“O projeto Sistema de Gerenciamento para Ginásio foi idealizado para atender às necessidades de organização e automação de informações relacionadas ao gerenciamento de academias.”



Sumário

1. Introdução

2. 1.1 Contextualização

3. 1.2 Objetivo do Relatório

4. Descrição do Projeto

2.1 Visão Geral

2.2 Escopo

2.3 Objetivos

5. Levantamento de Requisitos

3.1 Metodologia de Recolha

3.2 Requisitos Funcionais

3.3 Requisitos Não Funcionais

6. Análise de Sistemas

4.1 Modelagem

4.2 Casos de Uso

4.3 Análise de Riscos

7. Design do Sistema

5.1 Arquitetura do Sistema

5.2 Design de Componentes

5.3 Interface do Utilizador

5.4 Base de Dados

8. Implementação

6.1 Plano de Implementação

6.2 Tecnologias Utilizadas

6.3 Estrutura de Código

6.4 Controle de Versão

9. Testes

7.1 Plano de Testes

7.2 Casos de Teste

7.3 Resultados dos Testes

10. Implantação

8.1 Plano de Implantação

8.2 Formação

8.3 Suporte

11. Conclusão

9.1 Resumo dos Resultados

9.2 Próximos Passos

9.3 Reflexão

1. Introdução

1.1 Contextualização

O projeto Sistema de Gerenciamento para Ginásio foi idealizado para atender às necessidades de organização e automação de informações relacionadas ao gerenciamento de academias. Este projeto busca otimizar o registro de clientes, gestão de planos e pagamento, oferecendo soluções tecnológicas eficientes.

1.2 Objetivo do Relatório

Este relatório destina-se a apresentar o detalhamento do projeto, incluindo objetivos, requisitos, design e planejamento de implementação. Ele é voltado para stakeholders e equipes técnicas envolvidas no desenvolvimento e execução.

2. Descrição do Projeto

2.1 Visão Geral

O Sistema de Gerenciamento para Ginásio é uma plataforma desenvolvida para facilitar a gestão de clientes, planos e pagamentos, promovendo maior eficiência nas operações administrativas de academias.

2.2 Escopo

Inclui:

- Cadastro de clientes e planos.
- Consulta, edição e exclusão de dados.
- Geração de relatórios básicos.

Não Inclui:

- Integração com sistemas de pagamento automático.
- Recursos de marketing ou CRM.

2.3 Objetivos

- Melhorar a organização dos dados dos clientes.
- Reduzir erros no gerenciamento de planos.
- Agilizar o acesso às informações.

3. Levantamento de Requisitos

3.1 Metodologia de Recolha

Os requisitos foram levantados por meio de entrevistas com gestores de academias, questionários aplicados a usuários e workshops com a equipe de desenvolvimento.

3.2 Requisitos Funcionais

1. Cadastro de clientes com nome, idade, telefone e plano.
2. Consulta e edição de informações cadastradas.
3. Exclusão de registros.
4. Geração de relatórios simples.

3.3 Requisitos Não Funcionais

1. Interface intuitiva e de fácil navegação.
2. Banco de dados seguro e eficiente.
3. Escalabilidade para futuras implementações.

4. Análise de Sistemas

4.1 Modelagem

A modelagem do sistema foi realizada utilizando diagramas UML (Unified Modeling Language) para representar os fluxos de trabalho, entidades e interações do sistema. Os principais diagramas utilizados incluem:

- Diagrama de Casos de Uso para identificar as interações do usuário com o sistema.
- Diagrama de Classes para estruturar as entidades do banco de dados.

4.2 Casos de Uso

Os principais casos de uso do sistema incluem:

1. Cadastro de Cliente: O usuário registra um novo cliente com os dados necessários.
2. Consulta de Cliente: Busca por clientes utilizando filtros como nome ou ID.
3. Atualização de Dados: Modificação de informações previamente cadastradas.

4.3 Análise de Riscos

Os principais riscos identificados incluem:

-
- **Falhas de Segurança:** Acesso não autorizado aos dados do cliente.
Mitigação: Implementação de autenticação e controle de permissões.
 - **Inconsistência de Dados:** Possibilidade de erros ao manipular informações.
Mitigação: Validações no frontend e backend.

5. Design do Sistema

5.1 Arquitetura do Sistema

O sistema segue uma arquitetura cliente-servidor com as seguintes camadas:

- **Frontend:** Interface gráfica para interação do usuário.
- **Backend:** API desenvolvida em Python para manipulação dos dados.
- **Banco de Dados:** MySQL como gerenciador de banco de dados.

5.2 Design de Componentes

Os principais componentes incluem:

- **Módulo de Gerenciamento de Clientes.**
- **Módulo de Gerenciamento de Planos.**
- **Módulo de Relatórios.**

5.3 Interface do Utilizador

A interface foi projetada para ser simples e intuitiva, com:

- **Formulários** para cadastro e atualização de dados.
- **Menus** para navegação rápida entre funcionalidades.

5.4 Base de Dados

A modelagem do banco de dados segue o modelo relacional, com as seguintes tabelas principais:

- **Clientes:** Armazena dados pessoais e contato.
- **Planos:** Contém informações sobre os tipos de planos disponíveis.
- **Assinaturas:** Registra a vinculação entre clientes e planos.

6. Implementação

6.1 Plano de Implementação

A implementação será dividida em três fases:

1. **Desenvolvimento do Backend.**
2. **Desenvolvimento da Interface de Usuário.**

3. Integração e Testes.

6.2 Tecnologias Utilizadas

- **Linguagem de Programação: Python (Flask/Django).**
- **Banco de Dados: MySQL.**
- **Frontend: HTML, CSS, JavaScript.**

6.3 Estrutura de Código

O código será organizado em:

- **Diretórios para o backend (modelos, controladores).**

6.4 Controle de Versão

O repositório será gerenciado no GitHub, com uso de branches para diferentes funcionalidades e pull requests para revisões.

7. Testes

7.1 Plano de Testes

Os testes serão realizados em três níveis:

- **Testes Unitários: Garantir o funcionamento de funções individuais.**
- **Testes de Integração: Verificar a comunicação entre os módulos.**
- **Testes de Aceitação: Confirmar que o sistema atende às necessidades dos usuários.**

7.2 Casos de Teste

- 1. Validação de Cadastro: Verificar a entrada de dados obrigatórios.**
- 2. Pesquisa de Cliente: Confirmar que os filtros funcionam corretamente.**
- 3. Relatórios: Garantir a geração precisa de relatórios.**

7.3 Resultados dos Testes

Os testes iniciais identificaram erros menores, como validações ausentes, que foram corrigidos antes da implantação.

8. Implantação

8.1 Plano de Implantação

O sistema será implantado em um ambiente local para testes finais e, posteriormente, em produção. A transição será realizada fora do horário comercial para minimizar impactos.

8.2 Formação

Treinamentos serão realizados para funcionários, com manuais de uso e sessões práticas.

8.3 Suporte

Uma equipe de suporte estará disponível para resolver problemas durante o período inicial de uso.

9. Conclusão

9.1 Resumo dos Resultados

O projeto alcançou seus objetivos principais, entregando um sistema funcional e eficiente para a gestão de ginásios.

9.2 Próximos Passos

- Expandir o sistema para incluir funcionalidades como integração com sistemas de pagamento e relatórios avançados.
- Implementar Frontend: HTML, CSS, JavaScript.

9.3 Reflexão

O projeto demonstrou a importância de uma comunicação eficaz entre as partes interessadas e uma abordagem iterativa para o desenvolvimento. Lições aprendidas serão aplicadas em projetos futuros.

Link do Git Hub para ter acesso ao código: https://github.com/jessica-re88/Projeto_final.git