



UNIVERSIDADE FEDERAL DA PARAÍBA – CIÊNCIA DE DADOS PARA NEGÓCIOS
PROJETO ORIENTADO À OBJETOS

MR
Muscle Rank
SAÚDE E DIVERSÃO

**UM APP DE
GESTÃO DE
TREINOS
GAMEFICADO**

MR

GRUPO:



ALEX
TAVARES



ARTHUR
SILVA



KAIO
VICTOR

PROFESSORA: ADRIANA
DAMASCENO



Já pensou em deixar seu treino mais organizado e ao mesmo tempo mais divertido?

Nós somos a **Muscle Rank**, uma ferramenta dedicada a ajudar academias a terem alunos mais ativos e praticantes, tornando a atividade física mais divertida e competitiva. Através da gamificação, os alunos acumulam pontos ao realizar exercícios e treinos, competindo entre si com base em seus interesses e objetivos, enquanto disputam por recompensas motivadoras. Dessa maneira, estimulamos o engajamento e a constância nos treinos, transformando o hábito de se exercitar em uma experiência envolvente e desafiadora.

Serviços Oferecidos:

1. Adicionar Aluno
2. Adicionar Instrutor
3. Adicionar Exercício ao Treino do Dia
4. Concluir Exercício
5. Exibir Ranking de Alunos
6. Exibir Ranking de Instrutores
7. Exibir Informações da Academia
8. Exibir Alunos da Academia
9. Exibir Instrutores da Academia
10. SAIR

MR

Métodos Aplicados:

Classes, objetos, variáveis de instância e métodos.



Conclusão dos Métodos Aplicados:

A utilização de classes organiza o sistema em entidades bem definidas, facilitando a manutenção e expansão. Objetos armazenam dados específicos de cada entidade, garantindo o encapsulamento e a flexibilidade. Variáveis de instância permitem que cada objeto tenha seu próprio estado, tornando o sistema mais dinâmico. Já os métodos automatizam ações, como a gestão de alunos e ranking, e possibilitam lógicas complexas, tornando a interação mais fluida e eficiente.



Classes:

As classes definem a estrutura e o comportamento dos objetos do sistema. No código, temos: **Instrutor, Aluno, Exercício, Ranking, Academia e Interface.**

Variáveis de Instância:

As variáveis de instância armazenam dados dentro dos objetos e são acessadas através do self.

Alguns exemplos: **self.nome, self.cpf, self.data_nascimento** nas classes Instrutor e Aluno.

self.cnpj, self.endereco na classe Academia.

self.participantes na classe Ranking, que armazena os alunos e instrutores no ranking.



Objetos:

Os objetos são instâncias das classes, criadas para armazenar e manipular dados. No código, vemos exemplos como:

- **academia_top** = Academia("Academia Top", "12.345.678/0001-99", "Rua das Flores, 123") → **Um objeto da classe Academia.**
- **instrutor** = Instrutor("Carlos Silva", "123.456.789-00", "1985-06-15", "INST001", academia_top) → **Um objeto da classe Instrutor.**
- **aluno1** = Aluno("Ana Souza", "987.654.321-00", "2000-04-22", "ALU001", instrutor, academia_top) → **Um objeto da classe Aluno.**
- **ranking** = Ranking() → **Um objeto da classe Ranking.**
- **menu** = Interface(academia_top, ranking) → **Um objeto da classe Interface.**

Métodos:

Os métodos são funções definidas dentro das classes para manipular dados e realizar ações. O código usa métodos para:

Manipulação de dados:

- academia_top.inserir_aluno(aluno1): Método para adicionar um aluno à academia.
- ranking.adicionar_participante(instrutor): Método para adicionar um participante ao ranking.

Interação com o usuário:

- menu.exibir_menu(): Método da classe Interface que exibe as opções para o usuário.

Estruturação do fluxo do programa:

- O loop while executando: mantém o programa rodando até que o usuário saia do sistema.

Dúvidas e Dificuldades:

Durante o projeto encontramos...

- Dificuldades na extração de informações de uma Classes para outra por conta da utilização do repositório do GH.
- Conflito de Classes na hora da confecção do menu.
- Dúvidas sobre o escopo do software.





Obrigado pela
atenção!

[Clique aqui para obter o código!](#)