

APP Fitness: GymBro

Grupo 4

Turma Java 82



Integrantes



Ângelo dos Santos



Cleiton Reis



Eduardo Tosta



Iago Zandone



Victor Hugo Ferreira



Gabriel da Silva Julio



Vinicius Balbino

Introdução ao GymBro

Objetivo

Facilitar treinos personalizados sugerindo exercícios recomendados.

Motivação

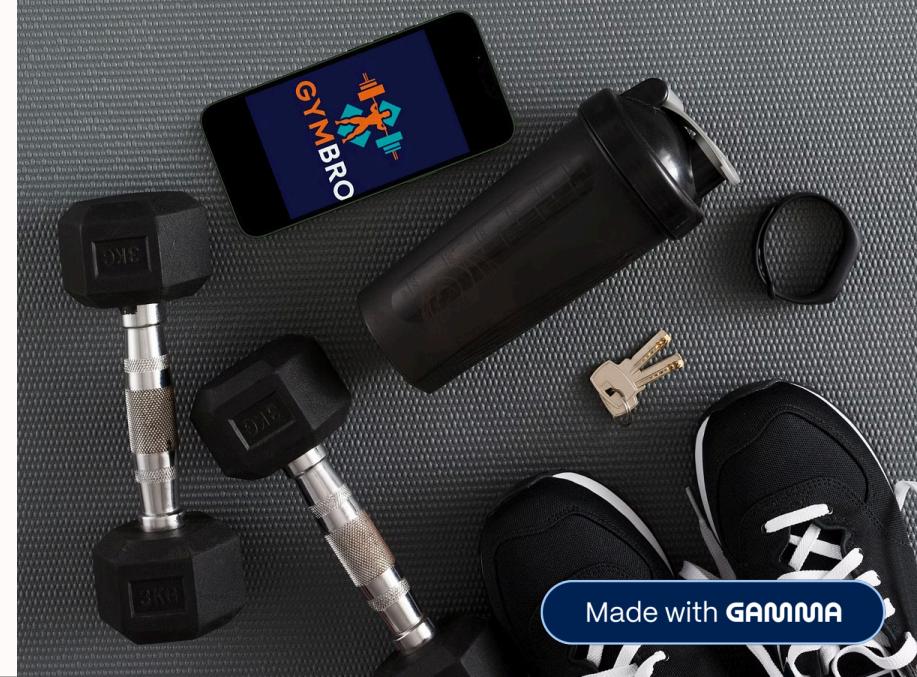
Promover saúde e disciplina por meio da tecnologia.

Público-alvo

Academias e seus Alunos.



A EVOLUÇÃO COMEÇA NO SEU
BOLSO

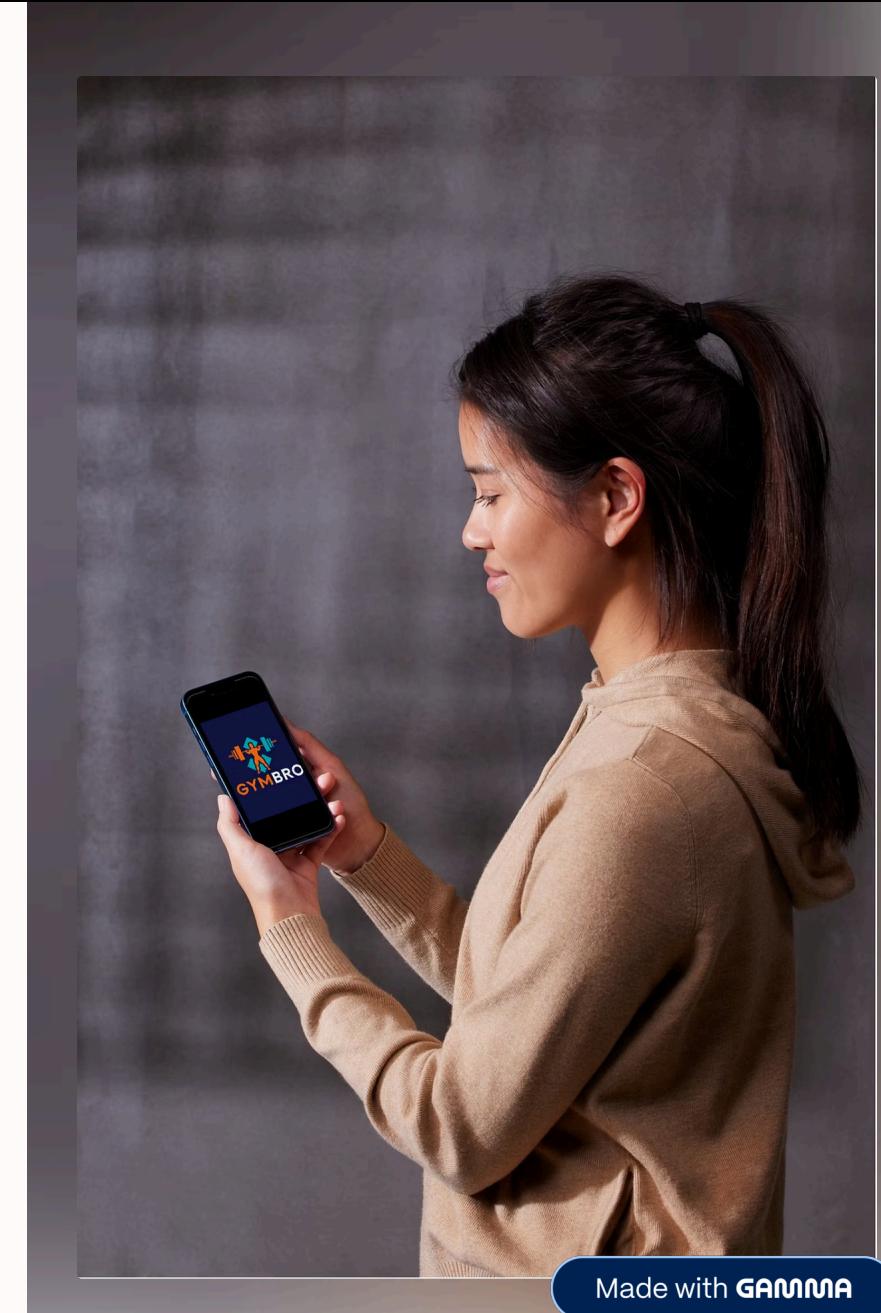


Made with **GAMMA**

Descrição do Sistema

Funcionalidades:

CRUD completo dos exercícios (Cadastro, Visualização, Alteração e Exclusão), Incluindo Pesquisa por nome.



Arquitetura do Sistema

Diagramas

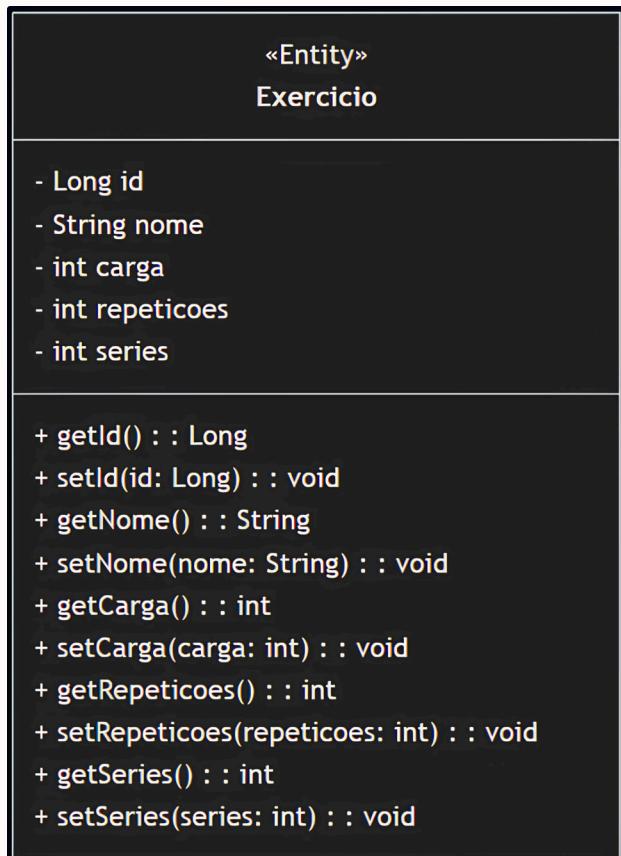


Diagrama de Classes

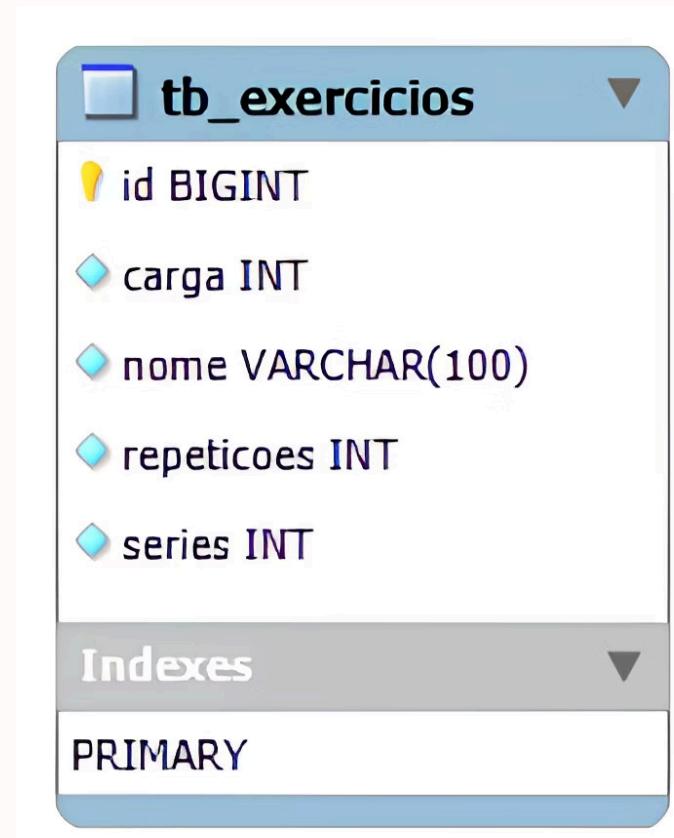


Diagrama Entidade-Relacionamento

Tecnologias Utilizadas



Java



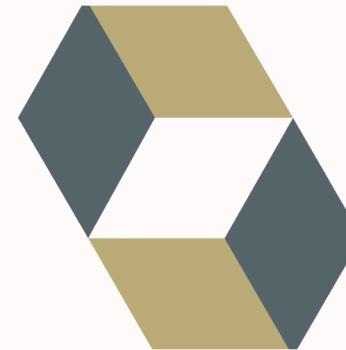
Spring Framework



MySQL



Insomnia



Hibernate

Testes Realizados

Testes Funcionais

Verificação das funcionalidades finais

The screenshot shows a mobile application interface with four main sections: **GET EXERCICIOS**, **POST EXERCICIOS**, **UPDATE EXERCICIOS**, and **DELETE EXERCICIOS**. Each section contains a description of the endpoint and a JSON representation of the data being manipulated.

- GET EXERCICIOS**:
Description: GET /exercicio - Recuperar uma lista de todos exercícios.
Data:

```
[  
  {  
    "id": 1,  
    "nome": "Treino Ombro",  
    "carga": 20,  
    "repeticoes": 12,  
    "series": 3  
  },  
  {  
    "id": 3,  
    "nome": "Treino teste",  
    "carga": 20,  
    "repeticoes": 15,  
    "series": 3  
  }  
]
```
- POST EXERCICIOS**:
Description: POST /exercicio - Registra um novo exercício no aplicativo
Data:

```
{  
  "nome": "Treino Braço e perna",  
  "carga": 20,  
  "repeticoes": 12,  
  "series": 3  
}
```
- UPDATE EXERCICIOS**:
Description: PUT /exercicio - Atualiza um exercício existente
Data:

```
{  
  "id": 32,  
  "nome": "Treino teste",  
  "carga": 20,  
  "repeticoes": 15,  
  "series": 3  
}
```
- DELETE EXERCICIOS**:
Description: DELETE /exercicio - Deleta um exercício existente
Data:

```
{  
  "id": 32  
}
```

Desafios e Soluções



Escalabilidade

Otimizações no banco e cache implementadas



Segurança

Autenticação e criptografia reforçadas



Usabilidade

Design responsivo e feedbacks constantes

```
25
26     @Autowired
27     private ExercicioRepository exercicioRepository;
28
29     @GetMapping
30     public ResponseEntity<List<Exercicio>> getAll(){
31         return ResponseEntity.ok(exercicioRepository.findAll());
32     }
33
34     @GetMapping("/{id}")
35     public ResponseEntity<Exercicio> getById(@PathVariable Long id){
36         return ResponseEntity.ok(exercicioRepository.findById(id)
37             .map(resposta -> ResponseEntity.ok(resposta))
38             .orElse(ResponseEntity.notFound().build()));
39     }
40
41     @GetMapping("/name")
42     public ResponseEntity<List<Exercicio>> getAllByName(@PathVariable String nome){
43         return ResponseEntity.ok(exercicioRepository.findAllByNameContainingIgnoreCase(nome));
44     }
45
46     @PostMapping
47     @Transactional
48     public ResponseEntity<Exercicio> postExercicio(@Id @RequestBody Exercicio exercicio){
49         return ResponseEntity.status(HttpStatus.CREATED).body(exercicioRepository.save(exercicio));
50     }
51
52     @PutMapping
53     @Transactional
54     public ResponseEntity<Exercicio> putExercicio(@Id @RequestBody Exercicio exercicio) {
55         if (exercicio.getId() == null) {
56             return ResponseEntity.badRequest().build();
57         }
58
59         return exercicioRepository.findById(exercicio.getId())
60             .map(respostaExistente -> {
61                 return ResponseEntity.ok().body(exercicioRepository.save(exercicio));
62             })
63             .orElse(ResponseEntity.notFound().build());
64     }
65
66     @DeleteMapping("/{id}")
67     @Transactional
68     public ResponseEntity<Void> deleteExercicio(@PathVariable Long id) {
69         return exercicioRepository.deleteById(id)
70             .map(resposta -> {
71                 exercicioRepository.deleteById(id);
72                 return ResponseEntity.status(HttpStatus.NO_CONTENT).ok.build();
73             })
74             .orElse(ResponseEntity.notFound().build());
75     }
76 }
```

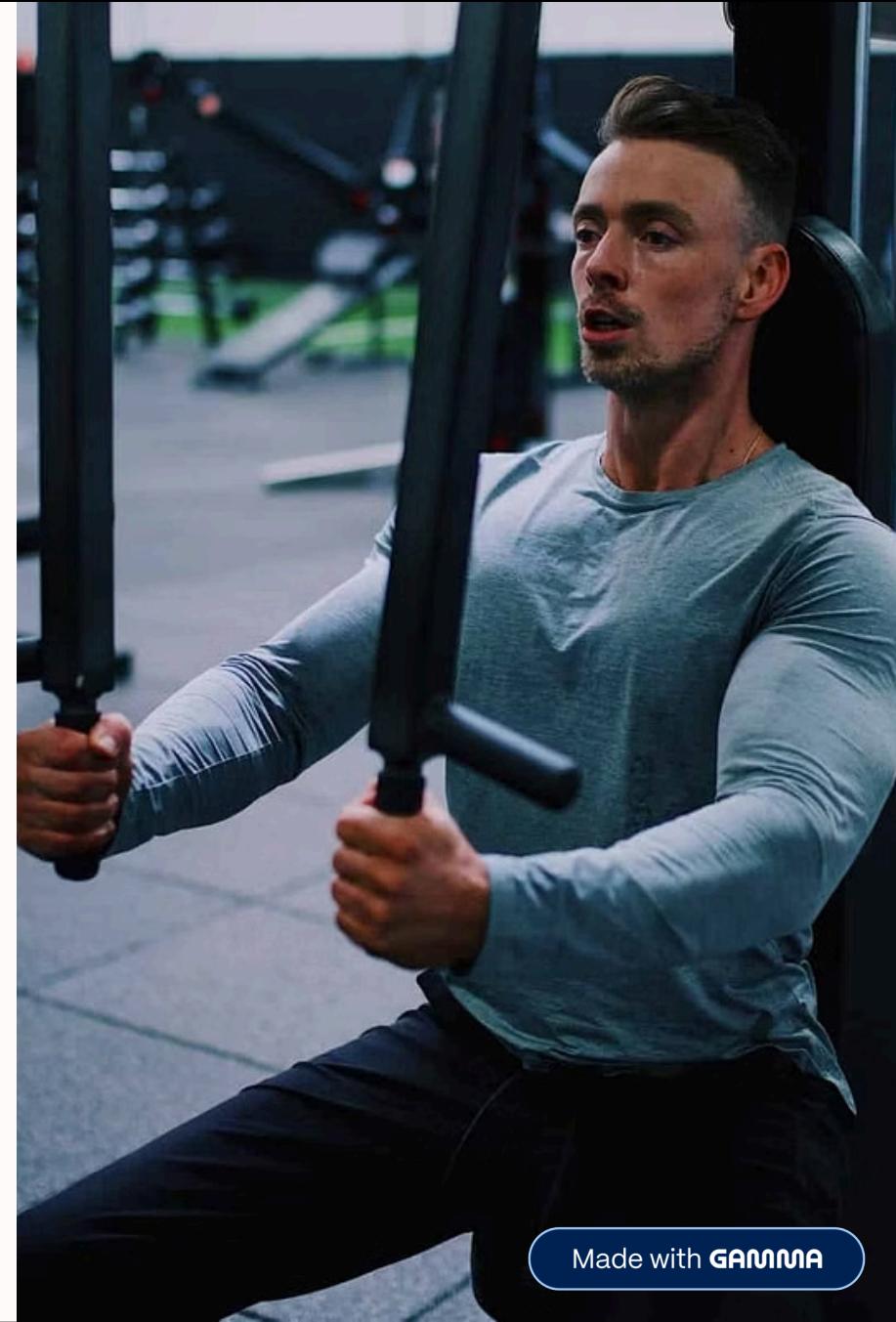
Impactos do Projeto

Ganhos para o cliente

- Redução do processos manuais e retrabalho
- Acesso rápido a dados de alunos e treinos
- Possibilidade de personalizar treinos com mais agilidade

Ganhos para o usuário

- Acesso fácil ao próprio treino
- Experiência mais moderna e conectada.
- Maior acompanhamento e motivação



Implementações Futuras

Planejadas

- Autenticação completa (Admin e Usuários)
- Dashboard de estatísticas
- Comunidade para usuários
- Plano de treino personalizado
- Inclusão de planos alimentares, monitoramento de progresso e IA.

Resumo

GymBro integra tecnologia e saúde, promovendo uma experiência inteligente e acessível para o bem-estar físico.

Com uma base sólida e boas práticas, o projeto tem alto potencial de crescimento e integração futura.



Conclusão

🤝 Obrigado pela atenção!

Projeto GymBro – Grupo 04 | Java 82

Quer saber mais ou nos acompanhar?

➡️ **Conecte-se com a gente no LinkedIn!**

👉 (Aponte a câmera para o QR Code ao lado) 👉

