



PROJETO DE BACK END

Aberto: domingo, 11 ago. 2024, 00:00

Vencimento: segunda-feira, 19 ago. 2024, 23:00

Descritivo: <https://github.com/digitalcollegebr/projeto-backend>

Requisitos de Avaliação, Critérios, Links úteis para estudo e algumas recomendações.

Requisitos de Avaliação do Projeto

1. *Funcionalidades Essenciais*
 - Implementação correta dos endpoints listados no escopo do projeto.
 - CRUD (Create, Read, Update, Delete) completo para as entidades principais.
 - Autenticação e autorização adequadas (se necessário).
 - Tratamento de erros e respostas apropriadas.
2. *Qualidade do Código*
 - Código limpo e bem comentado.
 - Uso adequado de boas práticas de programação.
 - Organização e estrutura do projeto (padrão MVC, por exemplo).
3. *Banco de Dados*
 - Modelagem correta do banco de dados.
 - Uso adequado de migrations.
 - Consultas eficientes e bem estruturadas.
4. *Documentação*
 - Documentação clara da API (pode ser com Swagger ou outra ferramenta).
 - Instruções de setup do ambiente de desenvolvimento.
 - Guia de uso da aplicação.



Critérios de Avaliação

1. *Completeness*
 - O projeto atende a todos os requisitos funcionais especificados no escopo?
2. *Code Quality*
 - O código está limpo, bem organizado e segue as boas práticas de programação?
3. *Database Design*
 - A modelagem do banco de dados é adequada e eficiente?

Links Úteis para Estudo

1. *Documentação de Linguagens e Frameworks*
 - [Express.js (Node.js)](<https://expressjs.com/>)
 - [Spring Boot (Java)](<https://spring.io/projects/spring-boot>)
 - [Django (Python)](<https://www.djangoproject.com/>)
2. *Conceitos de Banco de Dados*
 - [SQL Databases](<https://www.w3schools.com/sql/>)
 - [NoSQL Databases](<https://www.mongodb.com/nosql-explained>)
3. *Boas Práticas de Programação*
 - [Clean Code](<https://www.oreilly.com/library/view/clean-code/9780136083238/>)
 - [SOLID Principles](<https://en.wikipedia.org/wiki/SOLID>)



- [JUnit (Java)](<https://junit.org/junit5/>)
- [PyTest (Python)](<https://docs.pytest.org/en/6.2.x/>)
- [Jest (JavaScript)](<https://jestjs.io/>)

5. *Documentação de APIs*

- [Swagger](<https://swagger.io/>)
- [Postman](<https://www.postman.com/>)

Recomendações

1. *Planejamento e Organização*

- Divida o projeto em tarefas menores e crie um cronograma.
- Use ferramentas de gerenciamento de projetos como Trello ou Jira.

2. *Colaboração*

- Utilize controle de versão (Git) e repositórios remotos (GitHub, GitLab).
- Promova revisões de código entre os colegas.

3. *Prática Contínua*

- Incentive a prática contínua e a participação em projetos open source.
- Realize coding dojos e hackathons internos.

4. *Aprendizagem Contínua*

- Mantenha-se atualizado com as tendências e tecnologias do mercado.
- Participe de meetups e conferências da área.

Entrega e Avaliação

- *Data de Entrega Final:* 19/08/2024
- *Nota:* De 0 a 10, equivalente a 50% da nota final do curso.
- *Entrega:* Envio do link do repositório no GitHub. Para trabalhos em equipe, um único repositório deve ser entregue com todos os membros identificados.

Comunicação e Suporte

- *Monitores Disponíveis:* De segunda a quinta-feira, das 19h às 21h, os monitores estarão disponíveis para esclarecer dúvidas sobre o projeto de maneira mais direcionada.



Adicionar envio

Status de envio

Status de envio	Nenhum envio foi feito ainda
Status da avaliação	Não há notas
Tempo restante	2 dias 7 horas restando
Última modificação	-
Comentários sobre o envio	▶ Comentários (0)