**Portfólio para Dev Back-end**

**O que é analisado**

* Bem documentado
* Bem estruturado

**Ideia de projeto Back-end**

* API
* Gerar Login
* Validador de senhas
* Criptografar senha
* Estrutura de armazenamento (banco de dados)
* Autenticação (Token)
* Arquiteturas (MVC)

**Sobre o que fazer**

* Dificuldades do dia a dia
* Solucionar problemas reais
* Testes (Unitário)

**o que precisa ter**

* Funcionalidades
* Tecnologias utilizadas

**Referencias:**

<https://github.com/OneBitCodeBlog/restaurant-reviews-api>

<https://github.com/OneBitCodeBlog/onebitlife>

<https://mecanicashoptire.com.br/blog/>

*“Pretendo que o projeto se inicie sendo desenvolvido o back-end trazendo as funcionalidades, bem padronizado, boas práticas, separação de services etc, banco de dados, armazenamento na nuvem e testes”.*

Melhorias futuras:

Avaliações de peças ou popular

Esgotamento de item

Comentários

O que o mecânico vai fazer:

CRUD da peça (registrar, visualizar e deletar)

Colocar valor na peça

Colocar descrição

Colocar imagem

Visualizar comentários (o ADM podendo deletar)

O que o Cliente vai fazer:

Visualizar as peças

Visualizar o valor

Criar um pedido/carrinho

Adicionar comentário/ Excluir comentário

**Aqui estão algumas dicas adicionais para a documentação no README:**

Guia de Instalação:

Forneça instruções claras sobre como instalar e configurar o seu projeto. Inclua dependências necessárias e os passos específicos para colocar o sistema em funcionamento.

Exemplos de Uso:

Demonstre exemplos práticos de como usar sua API. Isso pode incluir exemplos de solicitações e respostas HTTP, bem como casos de uso típicos.

Configuração do Ambiente:

Se houver configurações específicas do ambiente, como variáveis de ambiente ou configurações do banco de dados, inclua essas informações para que outros desenvolvedores possam configurar o ambiente facilmente.

Contribuições:

Se você estiver aberto a contribuições externas, forneça orientações sobre como outros desenvolvedores podem contribuir para o projeto. Isso pode incluir informações sobre como relatar problemas, enviar solicitações de pull, etc.

Licença:

Especifique claramente a licença sob a qual o seu projeto é lançado. Isso ajuda a esclarecer como os outros podem usar, modificar e distribuir o seu código.

Exemplos de Telas (se aplicável):

Se houver uma interface do usuário associada ao projeto, considere incluir capturas de tela ou GIFs animados para fornecer uma visão rápida das funcionalidades.

Endpoints da API:

Liste todos os endpoints disponíveis na sua API, juntamente com uma breve descrição de cada um e exemplos de solicitações e respostas.

Instruções de Testes Unitários:

Se possível, forneça instruções sobre como os desenvolvedores podem executar os testes unitários. Isso pode incluir requisitos específicos e comandos para executar os testes.