



Teste Técnico

Desenvolvedor Backend

Python Pleno - Agriness

Expresse seus conhecimentos através da criação de soluções centradas no usuário!

Teste prático para a vaga de Desenvolvedor Backend na Agriness.

Objetivo do Teste

Desenvolver uma API de integração para o registro de nascimentos de leitões. A API deve permitir o cadastro de lotes de nascimento, onde o ID do lote, status e a quantidade de leitões nascidos são armazenados. A quantidade de leitões deve ser gerenciada utilizando Redis, sendo atualizada a cada modificação ou deleção do lote. O uso de Fila para o gerenciamento de mensagens.

Especificações Técnicas

1. Backend:

- Utilize Django ou Flask para desenvolver a API.
- A API deve permitir as seguintes operações:
 - **Cadastro de Lote de Nascimento:** Registrar um novo lote de nascimentos, especificando o ID do lote, status inicial (ex.: "created") e a quantidade de leitões nascidos.
 - **Consulta de Lotes:** Listar todos os lotes cadastrados, com possibilidade de filtrar por status.
 - **Atualização de Lote:** Permitir a atualização dos dados de um lote existente, especialmente a quantidade de leitões nascidos.
 - **Deleção de Lote:** Permitir a deleção de um lote existente.

2. Fila (por exemplo uso do RabbitMQ/Celery):

- Configure a Fila para enfileirar mensagens de integração referentes ao registro, atualização e deleção de lotes de nascimentos.
- A API deve enviar uma mensagem para a fila sempre que um lote for criado, atualizado ou deletado.
- Crie um consumidor que processe as mensagens e realize as operações correspondentes no banco de dados e no Redis.

3. Redis:

- Utilize Redis para armazenar a quantidade de leitões por lote.
- Sempre que um lote for atualizado ou deletado, a quantidade de leitões deve ser atualizada no Redis para refletir as mudanças.

4. Banco de Dados:

- Utilize PostgreSQL para persistir os dados dos lotes de nascimentos.

5. Docker (Opcional):

- Crie um Dockerfile para a aplicação.
- Utilize Docker Compose para orquestrar os serviços (API, Fila, Redis, banco de dados).

6. Testes:

- Implementar testes unitários e de integração para as principais funcionalidades da API, incluindo o gerenciamento de quantidade de leitões no Redis e o processamento de mensagens com Fila.

7. Documentação:

- Forneça um README com instruções detalhadas sobre como configurar e executar o projeto, incluindo a configuração do ambiente e execução dos testes.

Critérios de Avaliação

- Clareza e organização do código.
- Arquitetura da solução e boas práticas.
- Implementação correta das funcionalidades de integração, com destaque para o gerenciamento de quantidade de leitões no Redis.
- Capacidade de utilizar Filas para o processamento de mensagens.
- Implementação e cobertura de testes.
- Documentação clara e completa.
- Uso opcional de Docker.

Entrega

- Submeta o código em um repositório Git (GitHub, GitLab, etc.).
- Certifique-se de que o repositório esteja público ou forneça acesso para revisão.
- Compartilhe o link do repositório para avaliação.
- Tempo para desenvolvimento e entrega: 3 dias.

Boa sorte!