

- 1) Escreva uma função capaz de retornar, a partir de uma lista de aulas, aquelas que se iniciam em minuto ímpar.

Formato da representação de aula recebida:

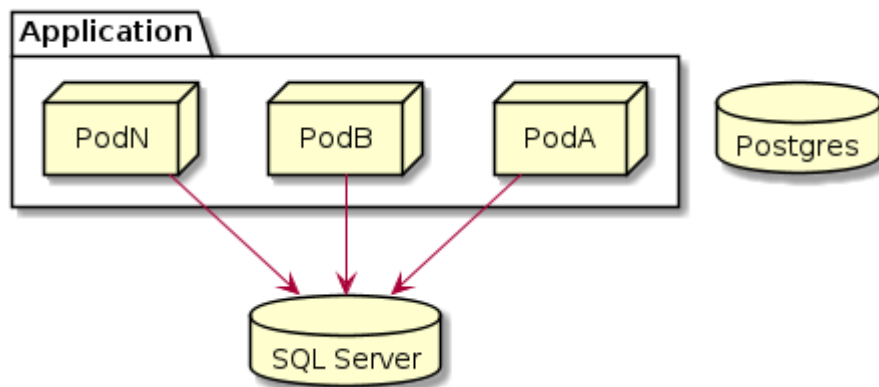
```
[  
  {  
    discipline: "math",  
    startHour: 11,  
    startMinute: 0  
  },  
  {  
    discipline: "biology",  
    startHour: 12,  
    startMinute: 15  
  },  
]
```

- 2) Escreva uma função que retorne o número de ocorrências de cada palavra em uma frase.

Frase de exemplo: *“Atirei o pau no gato, mas o gato não morreu.”*

- 3) Suponha que você tem uma API que roda 24/7 e que atende a uma enorme quantidade de requisições. Para suportar o tráfego, existem várias instâncias desta aplicação rodando num cluster Kubernetes.

Atualmente, esta aplicação utiliza SQL Server como forma de persistir seus dados:

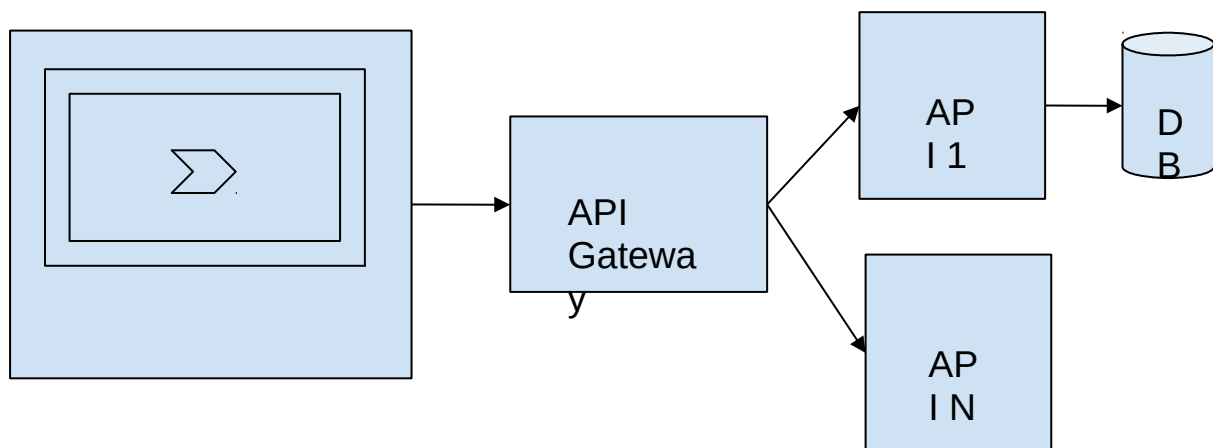


No entanto, o time de engenharia deseja migrar de banco para o PostgreSQL. Como você faria para migrar a aplicação sem downtime e sem perda de dados?

(Obs.: para esse problema, assuma que o schema é completamente compatível entre os dois SGBDs e nenhum tipo de conversão ou tratamento é necessário; precisamos apenas garantir que a aplicação passe de um banco para outro sem downtime e sem perda de dados.)

- 4) Suponha que você esteja trabalhando na API de um microsserviço, melhorando um endpoint que grava em que ponto de um vídeo um usuário está. Atualmente, essa informação é gravada no mesmo banco de dados onde as demais informações dos vídeos são armazenadas (nome, descrição, matéria, professor etc).

Com o aumento do número de usuários, o grande volume de escritas exigidas por esse endpoint tem impactado os demais endpoints da API. Ainda assim, o time de produto deseja aumentar a frequência do tracking de 15s para 5s, ou seja, guardaremos as posições dos usuários nos vídeos a cada 5s. Que soluções você veria para atender a esse requisito sem causar a piora do tempo de resposta do resto da API?



5) Utilizando a plataforma StackBlitz (<https://stackblitz.com>) e a API Swapi (<https://swapi.dev/api/people/>), implemente uma UI com React que permita visualizar e manipular uma lista de 10 personagens de Star Wars ordenados alfabeticamente, seguindo os passos abaixo:

- Criar componente Personagem com props Nome e Cor
 - Cor do Texto deve ser da cor do olho do personagem
- Adicionar um botão que remove o nome de um personagem da lista
 - Passo1: Uma lista com 1 botão por nome
 - Passo2: Um dropdown com a lista de nomes e um botao
- Escrever um teste unitário para garantir que estamos renderizando 10 componentes Personagem (pseudo código)
- Estilizar a lista com Flexbox ou CSS Grid (justificar a escolha). Centralizar no Desktop e alinhar à esquerda no breakpoint mobile (usar width: 768px)