CooperCarga – Jairo Bankhardt

1. Qual a saída do algoritmo?

A saída são 1000 números primos divididos em 5 páginas. Cada página é composta por 50 linhas, 4 colunas e 200 números primos.

2. Você julga que este código é limpo? Aponte quais erros o programador cometeu que prejudicaram a qualidade do código. Obs: não existe nenhum bug escondido no código.

Este não é um código limpo. Boas práticas incluem:

- As variáveis devem sem nomeadas adequadamente. Uso de camelCase.
- Variáveis com escopo local para uso dentro de seu próprio método.
- Particionamento do código em módulos.

3. Refatore o código do arquivo utilizando conceitos de Clean Code, de maneira que o código se torne mais limpo, legível e de fácil manutenção.

Código refatorado. Ver arquivo mainrefacted.js.

4. Explique como o conceito de middlewares no Express.js pode ser utilizado para evitar repetição de código.

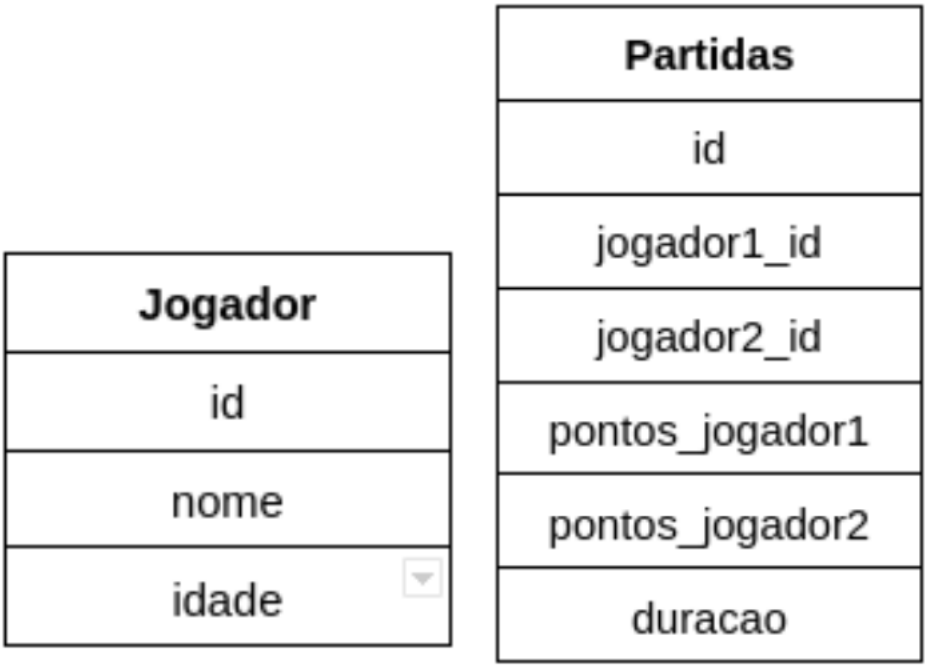
Os middlewares basicamente são funções que podem ser chamadas interceptando rotas específicas ou podem ser usadas em toda a aplicação. Assim, ao invés de repetir determinado código em várias requisições, é possível invocar um middleware para realizar a tarefa. O exemplo clássico é o uso de um middleware para autenticação/permissão de usuários.

5. Tendo em vista duas abordagens de backend: uma utilizando um ORM (como Prisma e Sequelize) e outra utilizando apenas um query builder (como o Knex), quais as vantagens e desvantagens de cada abordagem?

ORMs como Prisma e Sequelize oferecem uma abstração sobre o banco de dados através de objetos, eliminando a tarefa de escrever SQL diretamente. Isso facilita e simplifica ao precisar escrever um CRUD, por exemplo.

Já query builder oferecem maior controle sobre as consultas e melhor desempenho.

Para a pergunta 6, considere as tabelas:



6. Faça uma query em SQL que traga em cada linha o nome de jogadores que se enfrentaram mais de duas vezes, onde em cada partida a soma dos pontos foi maior que 30 e a duração do jogo foi maior que 90 minutos. Não podem haver resultados repetidos.

Exemplos de retorno válido:

Nadal	Murray
Nadal	Federer

Neste exemplo, Nadal enfrentou Murray pelo menos duas vezes, onde em cada partida a soma dos pontos foi maior que 30 e a duração do jogo foi maior que 90 minutos. O mesmo aconteceu com Nadal e Federer. Não há repetições.

Exemplo de retorno inválido:

Nadal	Murray
Murray	Nadal

A segunda linha invalida a primeira, pois é uma repetição. Mesmo que a ordem dos jogadores mudou, a partida é a mesma.

```
SELECT DISTINCT
    LEAST(j1.nome, j2.nome) AS jogador1,
    GREATEST(j1.nome, j2.nome) AS jogador2
FROM
    Partidas p
JOIN
    Jogador j1 ON p.jogador1_id = j1.id
JOIN
    Jogador j2 ON p.jogador2_id = j2.id
WHERE
     (p.pontos_jogador1 + p.pontos_jogador2) > 30
AND
    p.duracao > 90
GROUP BY
    jogador1, jogador2
HAVING COUNT(*) > 2;
```

7. Dado o array no arquivo data.json, crie um interface em React.js, CSS e Bootstrap mostre os itens como se fosse um marketplace de roupas.

Aplicação entregue no repositório.