

Você sabia que...

* A RUSSA MARIA BOCHKAREVA organizou o Batalhão Feminino da Morte?

* UM TENENTE BRASILEIRO LIDOU com o pelotão de soldados coloniais franceses de origem muçulmana?



HarperCollins
@HarperCollinsBR
@harpercollinsbrasil
/HarperCollinsBR

1) [9,0] Considere:

- (i) O banco de dados "acme", localizado em "https://mysql.acme.com", com charset "utf8", usuário "gerente" e senha "g3ReNT&", que contenha a tabela "fornecedor", cuja estrutura é indicada abaixo:

```
CREATE TABLE fornecedor (
  id          INT          NOT NULL AUTO_INCREMENT PRIMARY KEY,
  codigo      CHAR(8)      NOT NULL UNIQUE,
  nome        VARCHAR(100) NOT NULL,
  cnpj        CHAR(14)     NOT NULL UNIQUE,
  email       VARCHAR(60)  NOT NULL UNIQUE,
  telefone    CHAR(11)     NOT NULL
) ENGINE=INNODB;
```

- (ii) A classe Fornecedor, do arquivo "fornecedor.php":

```
<?php
namespace acme;
class Fornecedor {
  public int $id          = 0;
  public string $codigo   = '';
  public string $nome     = '';
  public string $email    = '';
  public string $cnpj     = '';
  public string $telefone = '';
  public function __construct( array $dados = [] ) {
    $this->id          = valor( $dados, 'id', 0 );
    $this->codigo      = valor( $dados, 'codigo', '' );
    $this->nome        = valor( $dados, 'nome', '' );
    $this->cnpj        = valor( $dados, 'cnpj', '' );
    $this->email       = valor( $dados, 'email', '' );
    $this->telefone    = valor( $dados, 'telefone', '' );
  }
}

function valor( array $a, $chave, $default ) {
  return htmlspecialchars( isset( $a[ $chave ] ) ? $a[ $chave ] : $default );
}
```

- (iii) A classe e a interface abaixo, ambos do arquivo "repositorio-fornecedor.php":

```
<?php
namespace acme;
require_once 'fornecedor.php';
class RepositorioException extends \RuntimeException {}

/** Todos os métodos devem lançar apenas a exceção do tipo RepositorioException. */
interface RepositorioFornecedor {
  /** Retorna um array de objetos de Fornecedor. */
  function todos( string $filtro = '' );
  /** O fornecedor recebido deve ganhar o ID gerado pelo meio de persistência. */
  function cadastrar( Fornecedor & $fornecedor );
  /** O ID do fornecedor não deve ser atualizado. */
  function atualizar( Fornecedor $fornecedor );
  /** Remove pelo id ou pelo código. */
  function remover( string $idOuCodigo );
}
```


Implemente uma API RESTful, com ponto de entrada em um arquivo "index.php", que disponibilize um recurso "/fornecedores" em que seja possível:

- 7,95
1,95
1,95
- [2,0] Consultar todos os fornecedores que **contenham**, em qualquer um de seus campos, o conteúdo do parâmetro "filtro" fornecido para a URL (ex. "/fornecedores?filtro=Xpto"). Se o parâmetro não for informado ou se seu valor estiver vazio, todos os fornecedores devem ser retornados. Os fornecedores devem ser retornados em formato JSON.
 - [2,0] Cadastrar um fornecedor, recebendo os dados no formato de URL codificada (x-www-form-urlencoded).
 - [2,0] Atualizar um fornecedor específico, identificado na URL pelo id sem usar parâmetros (ex. "/fornecedores/10"), recebendo os dados no formato JSON.
 - [2,0] Remover um fornecedor específico, identificado na URL pelo id ou pelo código sem usar parâmetros (ex. "/fornecedores/10" ou então "/fornecedores/AX-98.kf"). Veja o formato do campo código abaixo.

Para isso, é preciso:

- Retornar os códigos de status HTTP e cabeçalhos mais apropriados para cada caso, inclusive para quando o recurso requisitado não é encontrado.
- Utilizar o banco de dados indicado anteriormente, através de uma implementação da interface RepositorioFornecedor, observando o comportamento descrito em seus comentários;
- [1,0] Em uma função própria, validar os dados de um Fornecedor, tanto para o cadastro quanto para a alteração, retornando de uma só vez todos os problemas encontrados, caso haja, conforme as regras a seguir:
 - Cada atributo de Fornecedor deve respeitar os limites mínimo e máximo indicados nos campos da tabela fornecida;
 - O id deve ser um número inteiro e não pode ser negativo;
 - [0,5] O código deve ter, nessa ordem: duas letras maiúsculas, traço, dois números, ponto, duas letras minúsculas (ex. "AX-98.kf"). Utilize uma expressão regular para validá-lo.
 - O telefone e o CNPJ só devem conter caracteres numéricos.

DICA: Considere criar funções adicionais que agilizem a validação de coisas repetitivas.

0,8
0,7
0,2
0,3

2) [1,0] Complete as sentenças a seguir, sobre HTTP:

- [0,3] O cabeçalho que identifica o tipo (MIME) de um conteúdo enviado é Content-Type.
- [0,2] O cabeçalho que identifica o tamanho em bytes de um conteúdo é Size.
- [0,2] O cabeçalho que solicita que o navegador redirecione o usuário para outra localização é Location.
- [0,3] No HTML, os métodos suportados por um formulário são: POST, GET.

Boa prova!

Funções úteis (sintaxe simplificada; explicação não será fornecida):

```
is_numeric(mixed $var): bool;
is_int(mixed $var): bool;
isset(mixed $var [, mixed $...]): bool;
unset(mixed $var): void;
count(array $array): int;
array_push(array &$array, mixed $value1 [, mixed $value2], ...): int;
array_key_exists(string|int $key, array $array): bool;
empty(string|array $valor): bool;
explode(string $delimitador, string $texto [, int $limite]): array;
implode(string $delimitador, array $a): string;
dirname( string $arquivo ): string;
file_get_contents( string $arquivo ): string|false;
readline(string|null $prompt = null): string|false;
getallheaders(): array;
header( string $valor ): void;
http_response_code( int $codigo ): void;
htmlspecialchars( string $conteudo [, int $flags, string $encoding]): string;
json_encode( mixed $objetoOuArray ): string;
json_decode( string $conteudo ): mixed;
mb_parse_str( string $texto, &array $destino ): void;
preg_match(string $expressao, string $valor [, array &$casamentos [, int $flags=0 [, int $offset= 0]]]): int;
mb_strlen(string $texto): int;
mb_strpos(string $texto, string $busca [, int $inicio]): int;
mb_substr(string $texto, int $inicio [, int $tamanho]): string;
str_replace(mixed $busca, mixed $substituto, mixed $texto [, int &$modificacoesRealizadas]): mixed;
trim(string $texto [, string $caracteresARetirar]): string;
```

Algumas constantes	Algumas variáveis globais	Alguns status HTTP	
PHP_EOL	\$SERVER['PHP_SELF']	200 (OK)	400 (Bad Request)
PDO::ATTR_ERRMODE	\$SERVER['REQUEST_URI']	201 (Created)	401 (Unauthorized)
PDO::ERRMODE_EXCEPTION	\$SERVER['REQUEST_METHOD']	204 (No Content)	403 (Forbidden)
PDO::FETCH_ASSOC			404 (Not Found)
			405 (Method Not Allowed)
			500 (Server Error)