

Nome: Pedro Henrique Padilha Portella da Cruz

Telefone: (11) 947012902

E-mail: pedro.kadjin.sg@gmail.com

<https://github.com/PedroPadilhaPortella/2toMobility>

1) Escreva um algoritmo (em qualquer linguagem) que resolva a seguinte questão:

Um professor citou as seguintes definições:

1. Número esotérico é todo número inteiro divisível por 3 e 5
2. Número cético é todo número inteiro que não é esotérico
3. Número primo é todo número inteiro divisível somente por 1 e ele mesmo

Faça um algoritmo que dado um número inteiro descubra se ele é Esotérico, Cético e/ou Primo

```
/**
 * Checa se um numero inteiro é esotérico
 * @param {number} number
 * @returns `boolean` se o numero é isotérico ou não
 */
const isEsoteric = (number) => number % 3 == 0 && number % 5 == 0;

/**
 * Checa se um numero inteiro é primo
 * @param {number} number
 * @returns `boolean` se o numero é primo ou não
 */
function isPrimo(number) {
  for (let i = 2; i < number; i++) {
    if (number % i === 0) return false;
  }
  return true;
}

/**
 * Checa se um numero inteiro não é nem esotérico nem primo
 * @param {number} number
 * @returns `boolean` se o numero não é nem isotérico nem primo
 */
const isCetic = (number) => !isEsoteric(number) && !isPrimo(number);

async function main() {
  console.log(`Digite um número inteiro: `);
  process.stdin.on('readable', () => {
    const number = process.stdin.read().toString().replace(/\n/, "").replace(/\r/, "");

    if (isEsoteric(number))
      console.log(`${number} é um número esotérico`);

    if (isPrimo(number))
      console.log(`${number} é um número primo`);

    if (isCetic(number))
      console.log(`${number} é um número cético`);
  });
}

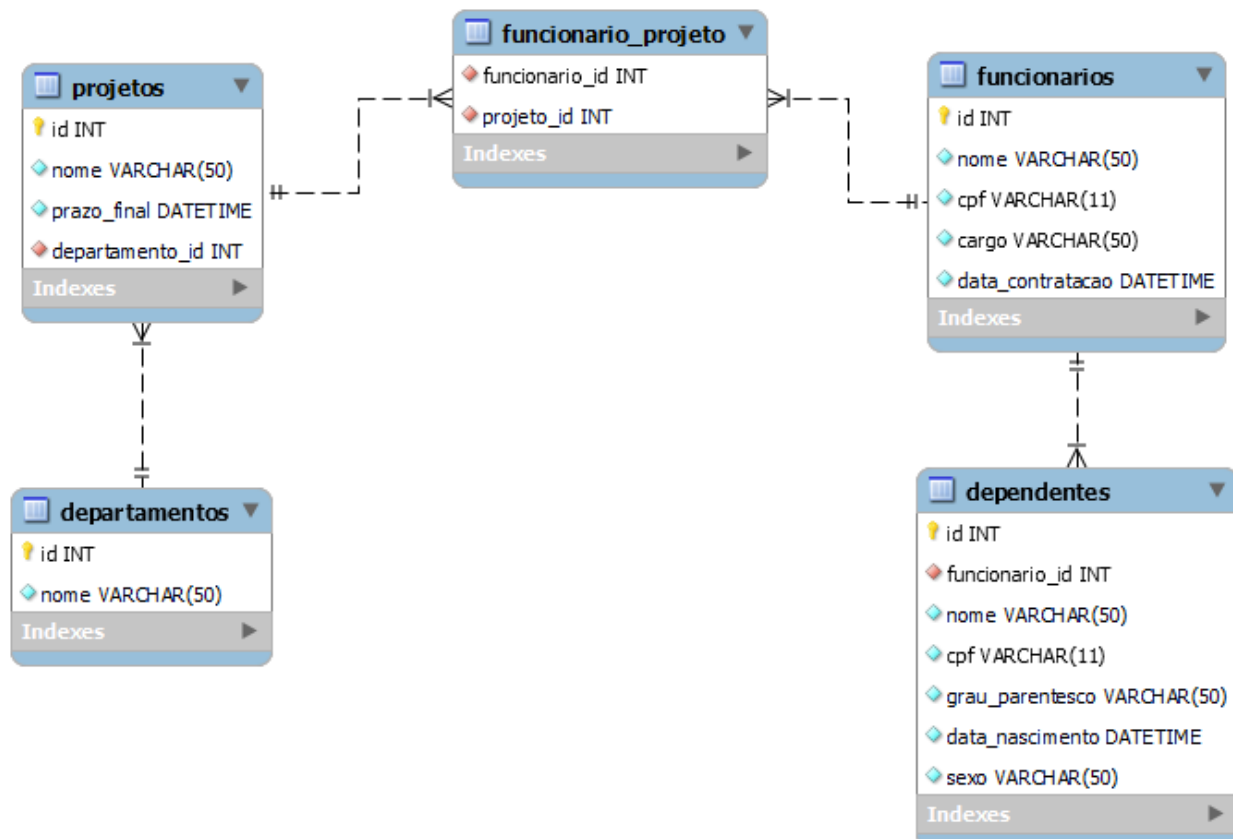
main()
```

Texto para as questões 2,3 e 4

A 2go é organizada em departamentos. Cada departamento possui um nome e controla um ou mais projetos. Cada projeto, por sua vez, possui um nome, prazo final, e um ou mais funcionários envolvidos. Os funcionários da 2go possuem um nome, CPF, cargo, e a data do contrato. Eles sempre estão alocados em um departamento e podem participar de um ou mais projetos. Os funcionários podem - ou não - possuir dependentes. Dependentes de um funcionário devem ser identificados por nome, CPF, grau de parentesco com o funcionário, data de nascimento, e sexo.

Uma tabela guarda conjuntos de informações de um mesmo tipo, podendo ter relações com outras tabelas. Com base no texto acima defina as tabelas, seus atributos e relacionamentos, de forma conceitual.

2) Defina as tabelas, campos e relacionamentos que possam representar o problema levantado no texto



3)Escreva as consultas SQLs para:

a) A lista de funcionários ordenado pelo nome do departamento e pelo nome:

```
select departamentos.nome as Departamento, projetos.nome as Projeto, funcionarios.nome as Funcionario
from funcionarios, funcionario_projeto, projetos, departamentos
where funcionarios.id = funcionario_projeto.funcionario_id
and funcionario_projeto.projeto_id = projetos.id
and projetos.departamento_id = departamentos.id
order by departamentos.nome ASC, funcionarios.nome ASC;
```

b) O número de funcionários em cada departamento:

```
select count(funcionarios.id) as 'Funcionarios por Departamento'
from funcionarios, funcionario_projeto, projetos, departamentos
where funcionarios.id = funcionario_projeto.funcionario_id
and funcionario_projeto.projeto_id = projetos.id
and projetos.departamento_id = departamentos.id
group by departamentos.id;
```

c) A lista de departamentos, com o nome do projeto mais recente (maior prazo final) de cada um:

```
select departamentos.nome as Departamento, projetos.nome as 'Projeto mais recente' ,  
max(projetos.prazo_final) as 'Prazo'  
  from projetos, departamentos  
 where projetos.departamento_id = departamentos.id  
 group by departamentos.id;
```

6)Desenhe o resultado da renderização das seguintes páginas HTML:

a)

```
<html>
  <body>
    <span style="border: 1px solid">Eu trabalho na </span>
    <span style="border: 1px solid">2go</span>
  </body>
</html>
```

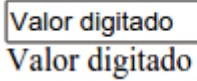
Eu trabalho na	2go
----------------	-----

b)

```
<html>
  <body>
    <div style="border: 1px solid">Eu trabalho na </div>
    <div style="border: 1px solid">2go</div>
  </body>
</html>
```

Eu trabalho na
2go

7) Faça o código HTML e o código javascript de uma página que tenha um campo de texto e conforme o usuário digite algo nesse campo, esse valor digitado apareça logo abaixo do campo:



a) Usando apenas HTML e javascript nativo (sem framework):
HTML:

JAVASCRIPT:

```
<input type="text" id="campo">
<p id="paragrafo"></p>

<script>
  const campo = document.querySelector('#campo');
  const paragrafo = document.querySelector('#paragrafo');

  campo.addEventListener('keyup', function () {
    paragrafo.innerHTML = campo.value;
  });
</script>
```

b) Usando apenas HTML e algum framework javascript:
HTML:

```
<form>
  <input type="text" [(ngModel)]="texto" name="texto">
  <p>{{ texto }}</p>
</form>
```

JAVASCRIPT:

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  texto: string = '';

  constructor() {}

  ngOnInit() {}
}
```

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, FormsModule],
  bootstrap: [AppComponent]
})
export class AppModule { }
```


8)Defina com poucas palavras:

a)Webservice Restful:

Serviços Web são aplicações desenvolvidas para a Web, que utilizam padrões com Http e Soap para se comunicar entre si, Serviços Web Restful são serviços que implementam o padrão Rest em suas aplicações. Restful significa que a aplicação possui uma série de princípios de Arquitetura recomendados para a construção de uma API, como por exemplo, o uso dos verbos do Http (Get, Put, Post, Delete), os códigos de status (200, 400, 404, 500), e HATEOAS.

b)HTTP:

Http é um protocolo de comunicação utilizado para a transferência de dados na web, é o protocolo que os web services utilizam ao se comunicar com a arquitetura Rest.

c)Microserviço:

A Arquitetura de Microserviços é, diferente das da Arquitetura Monolíticas, uma maneira de desenvolver serviços de modo mais desacoplado, a partir de pequenos serviços independentes que se comunicam entre si.