

Esboço mínimo

Aprendemos a lidar com o IndexedDB no capítulo anterior. Contudo, precisamos organizar todo aquele código que escrevemos em algo mais legível e fácil de manter. O primeiro passo é quebrar o problema em dois, ou seja, termos uma classe especializada na criação da conexão e outra especializada nas operações de persistência (inclusão, remoção, etc.). Vamos atacar o primeiro problema, o da conexão.

Qual das opções abaixo possui o esboço da classe `ConnectionFactory`, que não nos permite criar instâncias a partir dela e que possui o método estático `getConnection`, cujo retorno é uma *promise*? Ah, e não esqueça que precisamos usar o operador `new` quando formos criar uma *promise*!

Selecione uma alternativa

A

```
class ConnectionFactory {  
  
  constructor() {  
  
    throw new Error("ConnectionFactory não pode ser instanciada");  
  }  
  
  static getConnection() {  
  
    return Promise((resolve, reject) => {  
  
      });  
  }  
}
```

B

```
class ConnectionFactory {  
  
  constructor() {  
  
    throw new Error("ConnectionFactory não pode ser instanciada");  
  }  
  
  static getConnection() {  
  
    return new Promise((resolve, reject) => {  
  
      });  
  }  
}
```

C

```
class ConnectionFactory {  
  
  static getConnection() {  
  
    return new Promise((resolve, reject) => {  
  
      });  
  }  
}
```

```
}  
}
```

D

```
class ConnectionFactory {  
  
  constructor() {  
  
    throw new Error("ConnectionFactory não pode ser instanciada");  
  }  
  
  getConnection() {  
  
    return new Promise((resolve, reject) => {  
  
      });  
    }  
  }  
}
```