05

Dois tipos genéricos

Fernando utiliza muito o IndexedDB, um banco de dados que vive no próprio navegador. Com forte influência de padrões de projeto, decidiu criar um **GenericDAO**:

```
class GenericDAO {

    adiciona(objeto: Negociacao): number {

        /* implementação do método omitida */
    }

    apaga(objeto: Negociacao): void {

        /* implementação do método omitida */
    }

    buscaPorId(id: number): Negociacao {

        /* implementação do método omitida */
    }

    atualiza(objeto: Negociacao): void {

        /* implementação do método omitida */
    }

    listaTodos(): Negociacao[] {

        /* implementação do método omitida */
    }
}

// exemplo de uso
let dao = new GenericDao();
let negociacao = new Negociacao(new Date(), 1, 200);
// recebe o ID da negociação gerada
let id = dao.adiciona(negociacao);
let negociacaoBuscada = dao.buscaPorId(id);
```

O código escrito por Fernando não é genérico, pois esta amarrado ao tipo `Negociacao`. Além disso, o ID do elemento no IndexedDB pode ser um número ou uma string, e esse tipo esta fixo na definição da classe.

Marque a opção que torna a classe realmente genérica, permitindo persistir outros tipos, inclusive a definir um outro tipo de ID.

```
class GenericDAO<T, K> {

    adiciona(objeto: T): K {
```

```
        /* implementação do método omitida */
    }

    apaga(objeto: T): void {

        /* implementação do método omitida */
    }

    buscaPorId(id: K): T {

        /* implementação do método omitida */
    }

    atualiza(objeto: T): void {

        /* implementação do método omitida */
    }

    listaTodos(): T[] {

        /* implementação do método omitida */
    }
}
```

Correto! Pode indicar mais de um tipo genérico. No caso T, será o tipo da classe e K, o tipo do ID.

B

```
class GenericDAO<T> {

    adiciona(objeto: T): number {

        /* implementação do método omitida */
    }

    apaga(objeto: T): void {

        /* implementação do método omitida */
    }

    buscaPorId(id: number): T {

        /* implementação do método omitida */
    }

    atualiza(objeto: number): void {

        /* implementação do método omitida */
    }

    listaTodos(): T[] {

        /* implementação do método omitida */
    }
}
```

C

```
class GenericDAO<K> {
```

```
        adiciona(objeto: Negociacao): K {

            /* implementação do método omitida */
        }

        apaga(objeto: Negociacao): void {

            /* implementação do método omitida */
        }

        buscaPorId(id: K): T {

            /* implementação do método omitida */
        }

        atualiza(objeto: Negociacao): void {

            /* implementação do método omitida */
        }

        listaTodos(): Negociacao[] {

            /* implementação do método omitida */
        }
    }
```

Uma classe pode ter mais de um tipo genérico.

PRÓXIMA ATIVIDADE