

Exercício 04

Sistema de transações bancárias

Nessa sequência de exercícios vamos simular as operações básicas de um sistema bancário. Para elaboração dessa atividade vamos utilizar o conceito de métodos para os objetos criados.

a) Crie um objeto `contaBancaria`. Uma conta bancária será sempre um objeto com o seguinte formato:

```
const contaBancaria = {  
  nome: "Maria",  
  saldo: 0,  
  historicos: []  
}
```

Sabendo que as transações básicas de uma conta bancária são depositar, sacar e emitir extrato, faça o seguinte:

b) Declare um método `depositar` no objeto `contaBancaria` que tenha um argumento `valor`.

c) Implemente o método `depositar` de modo que o valor passado como argumento seja um número inteiro que representará o valor a ser depositado em centávos. Esse método precisa realizar as seguintes operações:

- Somar o valor informado como argumento ao saldo atual da conta;
- Adicionar um objeto ao array `historicos`. Esse objeto precisa ter o seguinte formato:

```
{  
  tipo: "Depósito",  
  valor: 10000, //aqui será o valor informado no argumento do método depositar  
}
```

- Retornar a seguinte mensagem: `Deposito de R$100 realizado para o cliente: Maria`.

d) Declare um método `sacar` no objeto `contaBancaria` que tenha um argumento `valor`.

e) Implemente o método `sacar` de modo que o valor passado como argumento seja um número inteiro que representará o valor a ser sacado em centávos. Esse método precisa realizar as seguintes operações:

- Verificar se o valor é maior que o saldo atual da conta e caso a condição for verdadeira retornar a seguinte mensagem: `Saldo insuficiente para o saque de: Maria`. Lembrando que, essa condição sendo verdadeira, as demais operações não poderão ser executadas;
- Subtrair o valor informado como argumento ao saldo atual da conta caso o valor seja menor que o saldo atual da conta;
- Adicionar um objeto ao array `historicos`. Esse objeto precisa ter o seguinte formato:

```
{
  tipo: "Saque",
  valor: 5000, //aqui será o valor informado no argumento do método sacar
}
```

- Retornar a seguinte mensagem: `Saque de R$50 realizado para o cliente: Maria`.

f) Declare um método `extrato` no objeto `contaBancaria`. Esse método não receberá argumento.

g) Implemente o método `extrato` de modo que o retorno desse método seja o seguinte:

```
Extrato de Maria - Saldo: R$50
Histórico:
Depósito de $100
Saque de $50
```

h) Para testar sua implementação faça as seguintes chamadas:

```
console.log(contaBancaria.depositar(10000));
console.log(contaBancaria.sacar(50000));
console.log(contaBancaria.sacar(5000));
console.log(contaBancaria.extrato());
```

Isso deve imprimir o seguinte resultado:

```
Deposito de R$100 realizado para o cliente: Maria
Saldo insuficiente para o saque de: Maria
Saque de R$50 realizado para o cliente: Maria
Extrato de Maria - Saldo: R$50
Histórico:
Depósito de $100
Saque de $50
```

Faça commit do resultado.

Preencha a checklist para finalizar o exercício:

- ☐ Resolver o exercício revendo a aula se necessário
- ☐ Adicionar as mudanças aos commits (`git add .` para adicionar todos os arquivos, ou `git add nome_do_arquivo` para adicionar um arquivo específico)

- ☐ Commitar a cada mudança significativa ou na finalização do exercício (`git commit -m "Mensagem do commit"`)
- ☐ Pushar os commits na sua branch na origem (`git push origin nome-da-branch`)

TAGS: **LÓGICA** **MÓDULO 1** **EXERCÍCIO DE CLASSE** **NODEJS** **FUNCAO** **OBJETOS**