

Exercício Prático: O Cofre Digital Seguro

Programação IV

6 de agosto de 2025

1 Objetivo

Construir um módulo de "Cofre Digital" usando o Padrão de Módulo (com uma IIFE) e closures. O objetivo é criar um estado interno (o saldo e a senha) que seja completamente privado e só possa ser modificado através de uma API pública e segura.

Este exercício testará sua compreensão de escopo, closures e encapsulamento.

2 Requisitos do Cofre

O seu módulo `Cofre` deve ter as seguintes características:

1. Estado Privado:

- Deve conter uma senha (`senha`) privada.
- Deve conter um saldo (`saldo`) privado.

2. API Pública:

- O módulo deve expor um objeto com métodos para interagir com o cofre.

3. Funcionalidades:

- `depositar(valor)`: Uma função pública que permite adicionar um valor ao saldo. Esta operação não requer senha.
- `sacar(senhaFornecida, valor)`: Uma função pública que permite sacar um valor do saldo.
 - Só deve funcionar se a `senhaFornecida` for igual à senha privada.
 - Não deve permitir o saque de um valor maior que o saldo atual.
 - Deve retornar uma mensagem de sucesso ou de erro.
- `consultarSaldo(senhaFornecida)`: Retorna o saldo atual, mas apenas se a senha estiver correta.
- `alterarSenha(senhaAntiga, senhaNova)`: Permite alterar a senha, mas apenas se a senha antiga estiver correta.

3 Passo a Passo da Implementação

3.1 Crie a Estrutura do Módulo

Comece criando uma IIFE (Immediately Invoked Function Expression) e atribua seu resultado a uma constante chamada `meuCofre`.

```
1 const meuCofre = (function() {  
2   // Todo o seu código vira aqui dentro...  
3 })();
```

3.2 Defina o Estado Privado

Dentro da IIFE, declare as variáveis privadas **senha** e **saldo** com valores iniciais à sua escolha (ex: senha "1234" e saldo 0).

3.3 Crie a API Pública e Implemente os Métodos

Dentro da IIFE, crie e retorne um objeto que conterá seus métodos públicos (**depositar**, **sacar**, etc.). Escreva a lógica para cada um dos métodos, garantindo que eles acessem e modifiquem as variáveis privadas corretamente.

4 Testando seu Cofre

Depois de implementar o módulo, use o seguinte código para testar sua funcionalidade.

```
1 // Testando o deposito
2 meuCofre.depositar(100);
3 console.log("Saldo apos deposito: " + meuCofre.consultarSaldo("senha-errada"));
4 console.log("Saldo apos deposito: " + meuCofre.consultarSaldo("1234"));
5
6 console.log("---");
7
8 // Testando o saque
9 console.log(meuCofre.sacar("senha-errada", 20));
10 console.log(meuCofre.sacar("1234", 150));
11 console.log(meuCofre.sacar("1234", 50));
12 console.log("Saldo apos saque: " + meuCofre.consultarSaldo("1234"));
13
14 console.log("---");
15
16 // Testando a alteracao de senha
17 console.log(meuCofre.alterarSenha("senha-errada", "5678"));
18 console.log(meuCofre.alterarSenha("1234", "5678"));
19 console.log("Saldo com a nova senha: " + meuCofre.consultarSaldo("5678"));
```

5 Funcionalidades

Adicione uma nova funcionalidade ao cofre:

- Crie uma variável privada chamada **tentativasFalhas** e um estado de **bloqueado** (booleano).
- Se a senha for inserida incorretamente 3 vezes seguidas, o cofre deve ser bloqueado.
- Enquanto estiver bloqueado, nenhuma operação pode ser realizada.