**Título: O Cofre da Herança 2.0**

**Contextualização:**

Você foi contratado por uma empresa de segurança digital para testar um sistema de cofres inteligentes implementado com conceitos de Programação Orientada a Objetos (POO). A empresa deseja avaliar se você compreende a estrutura de herança, o uso de atributos privados e protegidos, e a sobrescrita de métodos.

O sistema é composto por três classes Python: Cofre, CofreMestre e CofreFinal. A senha para abrir o cofre final é calculada automaticamente através da interação entre essas classes.

Seu objetivo é entender o funcionamento do código e descobrir qual é a senha secreta do cofre final.

**Objetivos da Atividade:**

* Compreender e aplicar os conceitos de herança e sobrescrita de métodos.
* Identificar a diferença entre atributos privados e protegidos.
* Simular o comportamento de um sistema real utilizando POO em Python.

### Instruções:

1. Analise o código abaixo:
2. Descubra qual é o valor retornado por CofreFinal().abrir().

class Cofre:

def \_\_init\_\_(self):

self.\_\_codigo\_secreto = 19

self.\_chave\_publica = 4

def \_processar\_base(self):

return self.\_\_codigo\_secreto \* self.\_chave\_publica

def abrir(self):

return self.\_processar\_base()

class CofreMestre(Cofre):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.\_bonus = 3

def \_processar\_bonus(self):

return super().abrir() + self.\_bonus \* 2

def abrir(self):

return self.\_processar\_bonus() - self.\_bonus

class CofreFinal(CofreMestre):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.\_fator\_divisor = 5

def abrir(self):

resultado = super().abrir()

return (resultado + self.\_fator\_divisor) // self.\_fator\_divisor