



Desafio proposto:

Criar uma api em **.NET** utilizando as melhores práticas de desenvolvimento, para resolver o problema a seguir.

Problema:

Nossos usuários têm custódia em vários tipos de investimentos, que vem de serviços distintos, para isso devemos consolidar estes dados e devolver para canais diversos. A seguir temos 3 endpoints para consulta de valores:

Tesouro Direto

<http://www.mocky.io/v2/5e3428203000006b00d9632a>

Renda Fixa

<http://www.mocky.io/v2/5e3429a33000008c00d96336>

Fundos

<http://www.mocky.io/v2/5e342ab33000008c00d96342>

Esperado:

Criar um endpoint que retorne o valor total do investimento do cliente e lista dos seus investimentos. Cada item da lista deverá conter seu valor unitário, cálculo de IR conforme regra abaixo e valor calculado caso o cliente queira resgatar seu investimento na data. O contrato esperado para o retorno é o seguinte:

```
{
  "valorTotal": 829.68,
  //Aqui deverão ser listados todos os investimentos retornados pelos 3 serviços
  "investimentos": [{
    "nome": "Tesouro Selic 2025",
    "valorInvestido": 799.4720,
    "valorTotal": 829.68,
    "vencimento": "2025-03-01T00:00:00",
    "Ir": 3.0208,
    "valorResgate": 705.228
  }]
}
```

Regras para cálculos:

IR:

	Taxa sobre a rentabilidade*
Tesouro Direto	10%
LCI	5%
Fundos	15%

***Rentabilidade = Valor Total – Valor Investido (Pode ser negativo)**

Cálculo para resgate:

Investimento com mais da metade do tempo em custódia: Perde 15% do valor investido

Investimento com até 3 meses para vencer: Perde 6% do valor investido

Outros: Perde 30% do valor investido

Resultado esperado:

Criar um repositório público no github, contendo a solução desenvolvida.

Readme explicando todas as decisões de projeto.

Caso tenha realizado o nível 3, fornecer o endpoint para consumo da Api

Critérios de Avaliação:

1. Organização do projeto
2. Boas práticas
3. Testes
4. Monitoria de Aplicação

Também usaremos os níveis abaixo para avaliar sua entrega:

Nível 1

Criar uma api consolidando os dados descritos no problema acima.

Nível 2

Uma boa prática para aplicações de alto desempenho é cachear as informações. O resultado pode ser cacheado até o dia seguinte (00:00).

Nível 3

Hospedar a aplicação em uma nuvem pública de sua escolha (heroku, next, aws, google cloud, azure, etc). Pode ser utilizado o free-tier, não será realizado nenhum teste de carga intensa.

Tempo estimado para entrega do Desafio: 1 semana