

# Desafio de Código: Simulação Simplificada de FIDC e Distribuição de Caixa

## Contexto

O objetivo deste exercício é implementar um programa de linha de comando (CLI) que simule o fluxo de caixa de um Fundo de Investimento em Direitos Creditórios (FIDC) simplificado e calcule a distribuição desse caixa entre diferentes classes de cotas, seguindo uma lógica de cascata.

## O que é um FIDC?

Um **FIDC (Fundo de Investimento em Direitos Creditórios)** é um tipo de fundo de investimento que tem como objetivo adquirir e gerir direitos creditórios, como duplicatas, parcelas de financiamentos, recebíveis de cartão de crédito, entre outros. Esses fundos são amplamente utilizados por empresas como forma de antecipar receitas futuras e, ao mesmo tempo, oferecer uma oportunidade de investimento estruturado para investidores.

Os FIDCs são regulados pela Comissão de Valores Mobiliários (CVM) e contam com regras específicas de governança, auditoria e transparência. Uma de suas principais características é a **estrutura de cotas por nível de risco**, conhecida como *estrutura de subordinação*.

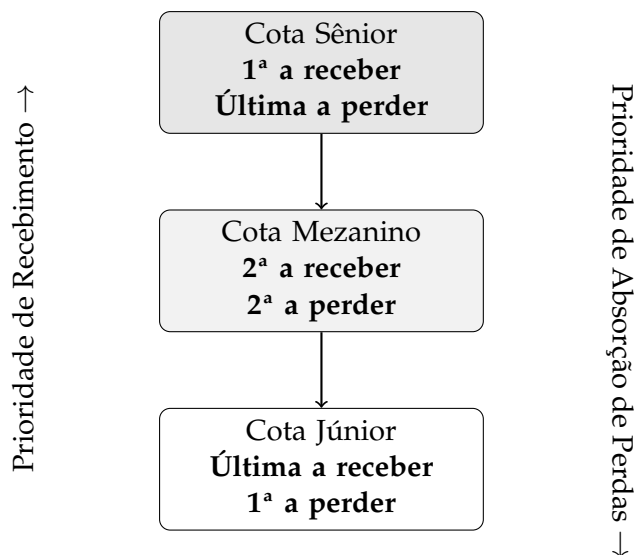
## Estrutura de Subordinação no FIDC

A estrutura de cotas de um FIDC normalmente é dividida em três classes:

- **Cotas Sênior** – com menor risco e prioridade máxima na distribuição de caixa.
- **Cotas Mezanino** – nível intermediário de risco e prioridade.
- **Cotas Júnior** – maior risco e menor prioridade de recebimento.

Essa organização cria uma hierarquia que protege os investidores mais conservadores e permite que os mais arrojados tenham potencial de maior retorno.

A seguir, uma imagem que reflete o mencionado:



### Resumo da lógica:

- Cotas **Sênior** recebem o *principal investido* e o *rendimento* prometido antes de qualquer outra classe.
- Cotas **Mezanino** recebem somente *rendimento*, se houver caixa restante após pagar os Sênior.
- Cotas **Júnior** absorvem as perdas primeiro e só recebem o *resíduo de caixa*, após todas as obrigações com as demais classes.

## Estrutura do FIDC (Simplificada)

Considere um FIDC genérico, que possui os seguintes parâmetros:

- **Principal Sênior Alvo:** R\$ 1.000.000,00
- **Rendimento Sênior Alvo:** R\$ 120.000,00
- **Rendimento Mezanino Alvo:** R\$ 75.000,00

## Regras de Distribuição (Cascata)

- O caixa recebido (`receive_cash`) é acumulado no `fund_cash`.
- As perdas (`register_default`) aumentam o `accumulated_loss_to_absorb`.
- Na operação `distribute_cash`, a ordem de alocação do caixa é:
  - Cobertura de perdas acumuladas
  - Pagamento do Principal Sênior
  - Pagamento do Rendimento Sênior
  - Pagamento do Rendimento Mezanino
  - Caixa restante vai para Júnior
- Após a distribuição, o caixa do fundo volta a zero.

## Formato da Entrada

Cada linha da entrada representa uma simulação independente e contém uma lista de operações em JSON via stdin:

```
[
  {"type": "receive_cash", "amount": 500000.00},
  {"type": "register_default", "amount": 10000.00},
  {"type": "distribute_cash"}
]
```

## Formato da Saída

A saída deve ser um objeto JSON com a distribuição dos valores da operação. Exemplo:

```
{
  "type": "distribute_cash",
  "total_distributed": 1195000.0,
  "distribution": {
    "loss_absorbed": 0.0,
    "senior_principal": 1000000.0,
    "senior_yield": 120000.0,
    "mezanino_yield": 75000.0,
    "junior": 5000.0
  }
}
```

## Estado Interno da Simulação

Cada linha de entrada reinicia o estado interno do fundo. Ou seja, cada simulação é isolada e não compartilha valores com as demais. O estado inclui:

- loss\_absorbed
- senior\_principal
- senior\_yield
- mezanino\_yield
- junior\_total\_value

## Casos de Teste (Exemplos)

### Exemplo 1 — Distribuição parcial do principal sênior

Entrada:

```
[
  {"type": "receive_cash", "amount": 500000.00},
  {"type": "distribute_cash"}
]
```

**Saída esperada:**

```
{
  "type": "distribute_cash",
  "total_distributed": 500000.0,
  "distribution": {
    "loss_absorbed": 0.0,
    "senior_principal": 500000.0,
    "senior_yield": 0.0,
    "mezanino_yield": 0.0,
    "junior_total_value": 0.0
  }
}
```

## **Exemplo 2 — Perda absorvida pela Júnior e distribuição parcial do yield sênior**

**Entrada:**

```
[
  {"type": "receive_cash", "amount": 1100000.00},
  {"type": "register_default", "amount": 10000.00},
  {"type": "distribute_cash"}
]
```

**Saída esperada:**

```
{
  "type": "distribute_cash",
  "total_distributed": 1090000.0,
  "distribution": {
    "loss_absorbed": 10000.0,
    "senior_principal": 1000000.0,
    "senior_yield": 90000.0,
    "mezanino_yield": 0.0,
    "junior_total_value": 0.0
  }
}
```

## **Exemplo 3 — Distribuição completa até Júnior**

**Entrada:**

```
[
  {"type": "receive_cash", "amount": 1200000.00},
  {"type": "distribute_cash"}
]
```

### Saída esperada:

```
{
  "type": "distribute_cash",
  "total_distributed": 1195000.0,
  "distribution": {
    "loss_absorbed": 0.0,
    "senior_principal": 1000000.0,
    "senior_yield": 120000.0,
    "mezanino_yield": 75000.0,
    "junior_total_value": 5000.0
  }
}
```

### Entrega

Entregar da forma que preferir, algumas sugestões:

- git - lembrar de deixar o repositório aberto
- .zip