

# Análise Financeira Simplificada: Descubra o Poder do Python para Decisões Estratégicas



## O Desafio da Gestão Financeira

Às vezes as empresas se deparam com a decisão de tomar novos recursos onerosos junto a bancos para sanar suas necessidades de caixa.

Dependendo da decisão tomada, a nova dívida poderá provocar reflexos negativos na rentabilidade da empresa o que é evidenciado pela métrica do retorno sobre o capital próprio ou do total emprestado.

Isso porque, quando se trata de gestão financeira, tomar decisões com base em dados é essencial e decisivo. Imagine a seguinte situação: sua empresa precisa decidir se uma nova captação de dívida é viável. Como calcular o impacto dessa dívida nos lucros e na rentabilidade?

Para ajudar nesse desafio, recorreremos à tecnologia, em especial à linguagem Python e criamos um simulador financeiro que faz todo o trabalho pesado para você.

## A Solução: Python para Administradores

O Python é uma linguagem de programação incrivelmente versátil. Com ele, criamos um script capaz de:

- Calcular lucros ajustados após custos de dívidas.
- Medir o impacto financeiro de novas captações.
- Gerar um relatório detalhado em PDF com todas as análises necessárias.

Tudo isso de forma automatizada, rápida e precisa.

## Como Funciona o Simulador

O simulador começa pedindo algumas informações básicas:

- **Capital Próprio:** Quanto os acionistas investiram na empresa.
- **Capital de Terceiros:** Dívidas já existentes.
- **Nova Dívida:** O valor que a empresa pretende captar.
- **Taxas de Juros:** Percentuais das dívidas existentes e da nova captação.
- **Receita e Custos Operacionais:** Valores estimados para o período.
- **Retorno do Investimento:** O impacto positivo esperado pela aplicação da nova dívida.

Com essas informações, o script realiza os cálculos e exibe os resultados em uma tabela no terminal. Além disso, ele gera um **relatório em PDF**, pronto para ser usado em reuniões ou apresentações.

```
(venv)
Izairton@DESKTOP-09EPSMI MINGW64 /1/VSCODE/PYTHON/ESTUDOS/captacao_bancaria_rentabilidade
$ python main.py
Bem-vindo ao simulador de rentabilidade!
Informe o valor do Capital Próprio (em R$): 1000000
Informe o valor do Capital de Terceiros (em R$): 500000
Informe a taxa de juros da dívida existente (%): 10
Informe o valor da nova captação (em R$): 300000
Informe a taxa de juros da nova dívida (%): 15
Informe a Receita Operacional esperada (em R$): 1500000
Informe os Custos Operacionais esperados (em R$): 1000000
Informe o retorno esperado do investimento da nova dívida (em R$): 100000
```

## Entendendo os Indicadores

Os resultados incluem uma análise detalhada de sete indicadores principais. Aqui está o que cada um deles significa:

1. **Lucro Operacional:** O lucro gerado pela empresa antes de considerar os custos financeiros e impostos. Ele mede a eficiência operacional.
2. **Custo da Dívida Total:** O custo combinado das dívidas existentes e da nova captação. Esse valor mostra quanto do lucro operacional será consumido para pagar juros.
3. **Lucro Ajustado:** O lucro restante após considerar os custos das dívidas. Ele revela se a empresa ainda é rentável após os encargos financeiros.
4. **Rentabilidade Inicial (%):** Mede a eficiência operacional antes de incluir os custos das dívidas. É calculado em relação à receita operacional.
5. **Rentabilidade Ajustada (%):** Avalia a eficiência após considerar os custos das dívidas e o retorno esperado do investimento.
6. **ROE Inicial (%):** Mostra o retorno sobre o capital próprio antes de incluir novas dívidas. Um ROE alto indica boa eficiência no uso do capital dos acionistas.
7. **ROE Ajustado (%):** Mede o retorno sobre o capital próprio após o impacto da nova dívida. Esse indicador é essencial para avaliar se a captação foi vantajosa.

```
PROBLEMAS  SAÍDA  CONSOLE DE DEPUÇÃO  TERMINAL  PORTAS  SQL CONSOLE  SPELL CHECKER  500

--- Informações Iniciais ---
+-----+-----+
| Indicador | Valor |
+-----+-----+
| Capital Próprio | R$ 1.000.000,00 |
+-----+-----+
| Capital de Terceiros | R$ 500.000,00 |
+-----+-----+
| Nova Dívida | R$ 300.000,00 |
+-----+-----+
| Taxa de Juros da Dívida Existente (%) | 10.00% |
+-----+-----+
| Taxa de Juros da Nova Dívida (%) | 15.00% |
+-----+-----+

--- Resultados da Simulação ---
+-----+-----+
| | Indicador | Valor |
+-----+-----+
| 0 | Lucro Operacional | R$ 500.000,00 |
+-----+-----+
| 1 | Custo da Dívida Total | R$ 95.000,00 |
+-----+-----+
| 2 | Lucro Ajustado | R$ 505.000,00 |
+-----+-----+
| 3 | Rentabilidade Inicial (%) | 30.00% |
+-----+-----+
| 4 | Rentabilidade Ajustada (%) | 33.67% |
+-----+-----+
| 5 | ROE Inicial (%) | 180.00% |
+-----+-----+
| 6 | ROE Ajustado (%) | 126.25% |
+-----+-----+

Relatório gerado com sucesso: relatorio_rentabilidade.pdf
(venv)
Izairton@DESKTOP-09EPSMI MINGW64 /1/VSCODE/PYTHON/ESTUDOS/captacao_bancaria_rentabilidade
$
```

## Relatório Automatizado

Para facilitar ainda mais o trabalho do administrador, o script gera um **relatório em PDF** que inclui:

- Todas as informações iniciais fornecidas.
- Os resultados calculados pelo simulador.
- Uma análise detalhada item por item, explicando o significado e impacto de cada indicador.

Com isso, o administrador pode tomar decisões informadas e apresentar os dados de forma clara para os stakeholders.

## Por Que Isso É Importante?

No contexto atual, num cenário de negócios onde os dados são cruciais na tomada de decisão, contar com ferramentas que automatizam cálculos e análises financeiras pode ser um divisor de águas. A ideia do simulador é oferecer:

- **Precisão:** Reduz o risco de erros em cálculos complexos.
- **Agilidade:** Economiza tempo, permitindo que o administrador se concentre em estratégias.
- **Clareza:** Gera relatórios que qualquer pessoa pode entender.

## Códigos do Projeto e Suas Funcionalidades

Para garantir o funcionamento de forma organizada, dividimos o projeto em três arquivos principais, cada um com responsabilidades específicas:

### 1. `simulacao.py`: Lógica de Cálculos Financeiros

Trata-se do arquivo que contém a lógica principal para os cálculos financeiros. Ele recebe os dados fornecidos pelo administrador (como receitas, custos, dívidas e taxas de juros) e calcula os indicadores financeiros relevantes.

Funções Principais:

- `simular_estrutura_capital()`: Realiza os cálculos dos indicadores financeiros, como:
  - Lucro Operacional
  - Custo da Dívida Total
  - Lucro Ajustado
  - Rentabilidade Inicial e Ajustada
  - ROE Inicial e Ajustado
- Retorna uma tabela com os resultados prontos para análise.

## 2. gerar\_relatorio.py: Geração do Relatório em PDF

Este arquivo é responsável por criar um relatório profissional em PDF, contendo:

- As **informações iniciais** fornecidas pelo administrador.
- Os **resultados da simulação**, exibidos em formato de tabela.
- Uma **análise detalhada**, explicando cada indicador.

Funções Principais:

- **PDF**: Classe personalizada para criar o PDF, incluindo cabeçalhos e rodapés.
- **gerar\_relatorio\_pdf()**: Função que formata e organiza os dados no relatório, gerando um arquivo **relatorio\_rentabilidade.pdf**.

## 3. main.py: Script Principal para Execução

Este é o ponto de entrada do projeto. Ele solicita ao administrador as informações necessárias para a simulação e coordena as outras partes do sistema.

Etapas do Script:

1. **Entrada de Dados**: Solicita os valores iniciais, como capital próprio, taxa de juros e receita operacional.
2. **Execução da Simulação**: Chama a função **simular\_estrutura\_capital()** do arquivo **simulacao.py** para realizar os cálculos.
3. **Exibição dos Resultados**: Mostra os resultados da simulação no terminal, formatados em tabelas.
4. **Geração do Relatório**: Usa a função **gerar\_relatorio\_pdf()** para criar o relatório em PDF com os resultados e análises.

Fluxo de Execução:

- O usuário interage com o **main.py** para fornecer os dados necessários.
- Os cálculos são feitos em **simulacao.py**, e o relatório final é gerado por **gerar\_relatorio.py**.

## Exemplo de Estrutura de Código

Aqui está um resumo do que cada arquivo faz e como eles se conectam:

**main.py**

- Solicita os dados do administrador
- Executa a simulação (chama funções de **simulacao.py**)
- Gera o relatório em PDF (chama funções de **gerar\_relatorio.py**)

**simulacao.py**

- └─ Contém a lógica para calcular os indicadores financeiros

gerar\_relatorio.py

- └─ Organiza os dados e gera o relatório em PDF

Com esses três arquivos bem-organizados, o sistema é modular, fácil de manter e escalável, podendo ser modificado, aprimorado ou expandido sem comprometer o restante do projeto.

## **Conclusão**

Como podemos verificar, um script desenvolvido em Python quando bem idealizado pode ser usado para resolver desafios reais. Quando aliado à teoria, torna-se uma ferramenta essencial ao administrador financeiro na tomada e no acerto de decisões, reafirmando a gestão financeira como a combinação de dados e tecnologia.

Anexos:

a) simulacao.py

```
import pandas as pd

def simular_estrutura_capital(
    receita_operacional,
    custos_operacionais,
    divida_atual,
    taxa_juros_atual,
    nova_divida,
    taxa_juros_nova,
    retorno_investimento
):
    """
    Realiza a simulação dos indicadores financeiros.
    """

    # Cálculo do lucro operacional antes das dívidas
    lucro_operacional = receita_operacional - custos_operacionais

    # Cálculo dos custos financeiros
    custo_divida_existente = divida_atual * (taxa_juros_atual / 100)
    custo_nova_divida = nova_divida * (taxa_juros_nova / 100)

    # Lucro operacional ajustado após as dívidas
    lucro_ajustado = lucro_operacional - custo_divida_existente -
    custo_nova_divida + retorno_investimento

    # Rentabilidade sobre a receita
    rentabilidade_inicial = (lucro_operacional - custo_divida_existente)
    / receita_operacional * 100
    rentabilidade_ajustada = lucro_ajustado / receita_operacional * 100

    # Retorno sobre o capital próprio (ROE)
    capital_total = divida_atual + nova_divida
    roe_inicial = (lucro_operacional - custo_divida_existente) /
    (divida_atual / 2) * 100
    roe_ajustado = lucro_ajustado / (capital_total / 2) * 100

    # Montar o DataFrame dos resultados
    resultado = {
        "Indicador": [
            "Lucro Operacional",
            "Custo da Dívida Total",
            "Lucro Ajustado",
            "Rentabilidade Inicial (%)",
            "Rentabilidade Ajustada (%)",
            "ROE Inicial (%)",
        ]
    }
```

```

        "ROE Ajustado (%)",
    ],
    "Valor": [
        lucro_operacional,
        custo_divida_existente + custo_nova_divida,
        lucro_ajustado,
        rentabilidade_inicial,
        rentabilidade_ajustada,
        roe_inicial,
        roe_ajustado,
    ],
}

return pd.DataFrame(resultado)

```

b) gerar\_relatorio.py

```

from fpdf import FPDF
from datetime import datetime

# Classe para criar o PDF
class PDF(FPDF):
    def header(self):
        self.set_font('Arial', 'B', 12)
        self.cell(0, 10, 'Relatório de Rentabilidade - Simulação Financeira', border=0, ln=1, align='C')
        self.ln(10)

    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', 'I', 8)
        self.cell(0, 10, f'Página {self.page_no()}', align='C')

# Função para criar o relatório
def gerar_relatorio_pdf(informacoes_iniciais, resultados, analise):
    pdf = PDF()
    pdf.add_page()

    # Cabeçalho
    pdf.set_font('Arial', '', 12)
    pdf.cell(0, 10, f'Data: {datetime.today().strftime("%d/%m/%Y")}',
ln=1, align='R')
    pdf.ln(5)

    # Endereçamento
    pdf.set_font('Arial', '', 12)

```



```

pdf.cell(0, 10, 'Para: Administrador da Empresa XYZ', ln=1)
pdf.cell(0, 10, 'De: Departamento Financeiro', ln=1)
pdf.ln(10)

# Informações iniciais
pdf.set_font('Arial', 'B', 12)
pdf.cell(0, 10, 'Informações Iniciais', ln=1)
pdf.set_font('Arial', '', 12)
for linha in informacoes_iniciais:
    pdf.cell(0, 10, f'{linha[0]}: {linha[1]}', ln=1)
pdf.ln(10)

# Resultados da simulação
pdf.set_font('Arial', 'B', 12)
pdf.cell(0, 10, 'Resultados da Simulação', ln=1)
pdf.set_font('Arial', '', 12)
for index, row in resultados.iterrows():
    indicador = row["Indicador"]
    valor = row["Valor"]

    # Remove a duplicidade do "%" no momento de adicionar ao PDF
    if "(%)" in indicador:
        valor = valor.replace("%%", "%")

    pdf.cell(0, 10, f'{indicador}: {valor}', ln=1)
pdf.ln(10)

# Análise detalhada
pdf.set_font('Arial', 'B', 12)
pdf.cell(0, 10, 'Análise Detalhada', ln=1)
pdf.set_font('Arial', '', 12)

# Adicionando análises item por item
pdf.multi_cell(0, 10, "1. Lucro Operacional:\n"
                  "O lucro operacional da empresa foi de R$ 500.000,00, indicando uma "
                  "boa capacidade de geração de lucros antes de custos financeiros e impostos.")
pdf.ln(5)

pdf.multi_cell(0, 10, "2. Custo da Dívida Total:\n"
                  "O custo total da dívida foi de R$ 95.000,00, composto por R$ 50.000,00 (juros da dívida existente a 10%) "
                  "e R$ 45.000,00 (juros da nova dívida a 15%). Esse custo representa 19% do lucro operacional.")
pdf.ln(5)

pdf.multi_cell(0, 10, "3. Lucro Ajustado:\n"

```

```

        "O lucro ajustado após o custo da dívida foi de
R$ 505.000,00. Esse valor mostra que, "
        "apesar dos custos financeiros, a empresa ainda
mantém uma rentabilidade saudável.")
    pdf.ln(5)

    pdf.multi_cell(0, 10, "4. Rentabilidade Inicial (%):\n"
        "A rentabilidade inicial foi de 30,00%,
calculada antes de considerar os custos das dívidas adicionais. "
        "Isso reflete uma operação eficiente com boa
margem sobre a receita operacional.")
    pdf.ln(5)

    pdf.multi_cell(0, 10, "5. Rentabilidade Ajustada (%):\n"
        "Após considerar o custo da dívida e o retorno
esperado do investimento, a rentabilidade ajustada foi de 33,67%. "
        "O aumento em relação à rentabilidade inicial
demonstra que o retorno do investimento foi positivo.")
    pdf.ln(5)

    pdf.multi_cell(0, 10, "6. ROE Inicial (%):\n"
        "O ROE inicial, ou retorno sobre o capital
próprio antes de incluir novas dívidas, foi de 180,00%. "
        "Esse alto valor indica uma excelente
eficiência no uso do capital dos acionistas.")
    pdf.ln(5)

    pdf.multi_cell(0, 10, "7. ROE Ajustado (%):\n"
        "Após considerar o impacto da nova dívida, o
ROE ajustado foi de 126,25%. Embora tenha diminuído em relação ao ROE
inicial, "
        "esse valor ainda reflete um retorno muito
atrativo, validando a decisão de captação.")
    pdf.ln(10)

# Outra opção do PARECER do relatório
# for linha in analise:
#     pdf.multi_cell(0, 10, linha)
# pdf.ln(10)

# Rodapé
pdf.set_font('Arial', 'I', 10)
pdf.cell(0, 10, 'Este relatório foi gerado automaticamente pelo
simulador de rentabilidade.', ln=1, align='C')

# Salvar o PDF
pdf.output('relatorio_rentabilidade.pdf')
print("Relatório gerado com sucesso: relatorio_rentabilidade.pdf")

```

c) main.py

```
import locale
from tabulate import tabulate
from gerar_relatorio import gerar_relatorio_pdf
from src.simulacao import simular_estrutura_capital

# Configurar o locale para moeda brasileira (R$)
locale.setlocale(locale.LC_ALL, 'pt_BR.UTF-8')

# Função para formatar valores como moeda ou percentual
def formatar_valor(indicador, valor):
    if "(" in indicador: # Verifica se o indicador é percentual
        return f"{valor:.2f}%" # Formata como percentual com 2 casas
    else:
        return locale.currency(valor, grouping=True) # Formata como
    moeda (R$)

# Solicitar valores do administrador para simulação
print("Bem-vindo ao simulador de rentabilidade!")
capital_proprio = float(input("Informe o valor do Capital Próprio (em
R$): "))
capital_terceiros = float(input("Informe o valor do Capital de Terceiros
(em R$): "))
taxa_juros_atual = float(input("Informe a taxa de juros da dívida
existente (%): "))
nova_divida = float(input("Informe o valor da nova captação (em R$): "))
taxa_juros_nova = float(input("Informe a taxa de juros da nova dívida
(%): "))
receita_operacional = float(input("Informe a Receita Operacional esperada
(em R$): "))
custos_operacionais = float(input("Informe os Custos Operacionais
esperados (em R$): "))
retorno_investimento = float(input("Informe o retorno esperado do
investimento da nova dívida (em R$): "))

# Executar a simulação
resultado = simular_estrutura_capital(
    receita_operacional,
    custos_operacionais,
    capital_terceiros,
    taxa_juros_atual,
    nova_divida,
    taxa_juros_nova,
    retorno_investimento
)
```

```

# Adicionar informações iniciais à tabela
informacoes_iniciais = [
    ["Capital Próprio", locale.currency(capital_proprio, grouping=True)],
    ["Capital de Terceiros", locale.currency(capital_terceiros,
grouping=True)],
    ["Nova Dívida", locale.currency(nova_divida, grouping=True)],
    ["Taxa de Juros da Dívida Existente (%)",
f"{taxa_juros_atual:.2f}%"],
    ["Taxa de Juros da Nova Dívida (%)", f"{taxa_juros_nova:.2f}%"],
]

# Formatando os resultados
resultado["Valor"] = resultado.apply(lambda row:
formatar_valor(row["Indicador"], row["Valor"]), axis=1)

# Exibir as informações iniciais no terminal
print("\n--- Informações Iniciais ---")
print(tabulate(informacoes_iniciais, headers=["Indicador", "Valor"],
tablefmt="grid"))
print("\n--- Resultados da Simulação ---")
print(tabulate(resultado, headers="keys", tablefmt="grid"))

# Adicionar uma análise detalhada para o relatório
analise_detalhada = [
    "1. Lucro Operacional: Representa o lucro gerado antes de considerar
os custos financeiros.",
    "2. Custo da Dívida Total: Reflete o custo das dívidas existentes e
da nova captação.",
    "3. Lucro Ajustado: Lucro restante após deduzir todos os custos
financeiros.",
    "4. Rentabilidade Inicial (%): Mede a eficiência operacional antes de
incluir novas dívidas.",
    "5. Rentabilidade Ajustada (%): Avalia a eficiência operacional após
considerar a nova captação.",
    "6. ROE Inicial (%): Mostra o retorno sobre o capital próprio antes
das novas dívidas.",
    "7. ROE Ajustado (%): Mostra o retorno sobre o capital considerando o
impacto da nova captação.",
]

# Gerar o relatório em PDF
gerar_relatorio_pdf(informacoes_iniciais, resultado, analise_detalhada)

```

Siga-me no LinkedIn: [www.linkedin.com/comm/mynetwork/discovery-see-all?usecase=PEOPLE\\_FOLLOWS&followMember=izairton-oliveira-de-vasconcelos-a1916351](https://www.linkedin.com/comm/mynetwork/discovery-see-all?usecase=PEOPLE_FOLLOWS&followMember=izairton-oliveira-de-vasconcelos-a1916351)