

AD1: Desconto Racional por Dentro

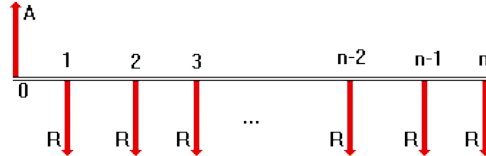
1 Descrição do Problema

Suponha que:

- o preço de uma mercadoria a prazo é x reais e que o preço à vista é y reais.
- o valor é pago em p prestações mensais iguais a $R = x/p$ com uma de entrada $(1 + (p-1))$
- que o mercado esteja adotando uma remuneração bancária fixa de $t\%$ ao mês (taxa).

1.1 Descobrindo o Valor Presente

Para determinar se vale a pena comprar a prazo ou não, é necessário obter o valor presente da compra, e descobrir o quanto o comerciante ou a financeira estão cobrando a mais.



O preço atualizado A , voltando cada parcela para o tempo inicial 0, é a soma de uma P.G.¹ de razão $q = \frac{1}{(1+t)}$ e cujo primeiro termo é q : $R q \frac{(1-q^n)}{1-q}$, onde $R = \frac{x}{p}$ é o valor de cada parcela.

Como, neste exercício, a primeira parcela é paga no ato da compra, na realidade, $n = p - 1$ e deve-se somar $\frac{x}{p}$ (a entrada):

$$x_{atualizado} = A = \frac{x}{p} \left(1 + q \frac{1 - q^{(p-1)}}{1 - q} \right).$$

Com as substituições necessárias, chega-se a seguinte fórmula:

$$A = \frac{x}{p} * \frac{(1+t)^p - 1}{t(1+t)^{(p-1)}} = x * \frac{1+t}{p * CF}, \quad CF = \frac{t}{1 - (1+t)^{-p}}.$$

¹<https://www.todamateria.com.br/progressao-geometrica/>

Para calcular ² o valor das parcelas fixas, no sistema de amortização francês (Crédito Direto ao Consumidor - CDC ³ no Brasil), dados o valor do empréstimo, v , o número de parcelas, p , e a taxa de juros, t , basta usar o Coeficiente de Financiamento, $CF(p, t)$:

$$R = \frac{x}{p} = v * \frac{CF(p, t)}{f(e)} \text{ (valor das prestações em função de } v, t \text{ e } p), \quad (1)$$

Para calcular ⁴ o valor presente no tempo 0, dados o valor final do empréstimo, x , o número de parcelas, p , e a taxa de juros, t , basta usar o Coeficiente de Financiamento, $CF(p, t)$:

$$A = \frac{x}{p} * \frac{f(e)}{CF(p, t)} \text{ (valor presente em função de } x, t \text{ e } p), \quad (2)$$

onde $f(e) = 1 + t$ ou 1, caso haja uma entrada, ou não, respectivamente.

É comum, o comerciante não informar a taxa de juros ou o preço à vista, dizendo que os juros valem zero. Nesse caso, só nos resta usar a taxa Selic como uma base para a taxa de juros empregada. Portanto, o seu programa deve aceitar três opções:

- Opção 1: Imprimir o preço atualizado e o percentual pago a mais ($\frac{A-y}{A} * 100$), ou seja, o custo estimado do financiamento ⁵.

Em geral, a taxa t é muito maior do que a taxa Selic, e às vezes vale mais a pena aplicar o dinheiro no mercado financeiro e pagar à vista mais tarde.

Se o preço à vista oferecido for maior do que o preço corrigido, vale mais a pena comprar a prazo, nesse caso.

- Opção 2: Classificar a adequação do parcelamento da seguinte forma.

Se o percentual pago a mais estiver entre 0% e 1%, imprima VALOR BAIXO.

Se o percentual pago a mais estiver entre 1% e 3%, imprima VALOR ACEITÁVEL.

Se estiver entre 3% e 5%, imprima ESTÁ CARO.

Se estiver acima de 5%, imprima VOCÊ ESTÁ SENDO ROUBADO.

- Opção 3: Imprimir qual deveria ser o preço total a prazo para ninguém sair ganhando (ou seja, que iguale o preço à vista, quando atualizado com a taxa t) e o fator, $k = \frac{f(e)}{p*CF}$, a ser aplicado (caso você quisesse levar para as lojas, por exemplo, para ter uma base de comparação).

1.2 Descobrimos a Taxa Real

Para o caso em que se tenha o valor à vista, o número de prestações e o valor final a prazo, mas não se conheça a taxa de juros empregada, é necessário obtê-la por um método numérico,

²Neste caso, $A = y = v$ e não se conhece x .

³<http://orion.lcg.ufrj.br/python/html/cdc.html>

⁴Neste caso, $A = y$ e não se conhece y .

⁵Embora o preço a prazo corrigido devesse ser igual ao preço à vista ($A=y$), isso vai depender da taxa.

pois não existe uma fórmula explícita para isso. Encontrar a taxa t , que produz o preço à vista y , requer o método de Newton⁶ ⁷, ou o uso de uma planilha Excel⁸:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

com entrada:

$$y = \frac{x}{p} \frac{(c-1)}{tb}, \text{ onde } a = (1+t)^{(p-2)}, b = (1+t)^{(p-1)}, c = (1+t)^p$$

$$f(t) = ytb - \frac{x}{p}(c-1), \quad f'(t) = y(b + t(p-1)a) - xb \quad (3)$$

sem entrada:

$$y = \frac{x}{p} \frac{(1-a)}{t}, \text{ onde } a = (1+t)^{-p}, b = (1+t)^{(-p-1)}$$

$$f(t) = yt - \frac{x}{p}(1-a), \quad f'(t) = y - xb \quad (4)$$

$$t_{n+1} = t_n - \frac{f(t)}{f'(t)}, \quad t_o = \frac{x}{y} \text{ (preço a prazo / preço à vista)}. \quad (5)$$

A função (5) é decrescente e converge para $t : \lim_{n \rightarrow \infty} t_n = t$. A condição de parada pode ser: $|t_{n+1} - t_n| \leq 1.0 \text{ E-04}$.

⁶https://en.wikipedia.org/wiki/Newton's_method

⁷<https://www.youtube.com/watch?v=WuaI5G04Rcw>

⁸<http://www.real-statistics.com/matrices-and-iterative-procedures/newtons-method/>

```

## Acha a taxa que produz o preço à vista pelo método de Newton.
#
# Nota: se não houve entrada, retorna getInterest2(x, y, p)
#
# Taxa Poupança:
# - 70% da taxa Selic (2% ao ano) ou
# - 1.40% ao ano ou
# - 0.1159% ao mês.
#
# @param x preço a prazo.
# @param y preço à vista.
# @param p número de parcelas.
# @return taxa, número de iterações.
#
def getInterest(x, y, p):

```

Exemplo (com entrada): Se $p = 10$, $t = 1\%$ ao mês, $x = \$500$ e $y = \$450$, então:

- O preço à vista é menor do que o preço total corrigido \rightarrow compre à vista.
- Taxa Real = 2.4227% ao mês (15 iterações, fator aplicado = 0.9566)
- Preço a prazo - juros de 1% ao mês = \$478.30
- Preço à vista + juros de 1% ao mês = \$470.42
- Juros Embutidos = $(\$500.00 - \$450.00) / \$450.00 * 100 = 11.11\%$
- Desconto = $(\$500.00 - \$450.00) / \$500.00 * 100 = 10.00\%$
- Excesso = $\$478.30 - \$450.00 = \$28.30$
- Excesso = $(\$500 - \$470.42) * 0.9566 = \$28.30$
- Percentual pago a mais = 5.92%
- Valor financiado = $\$450.00 - \$47.04 = \$402.96$

Tabela Price

Mês	Prestação	Juros	Amortização	Saldo Devedor
1	47.04	4.03	43.01	359.95
2	47.04	3.60	43.44	316.50
3	47.04	3.17	43.88	272.63
4	47.04	2.73	44.32	228.31
5	47.04	2.28	44.76	183.55
6	47.04	1.84	45.21	138.35
7	47.04	1.38	45.66	92.69
8	47.04	0.93	46.11	46.58
9	47.04	0.47	46.58	0.00
Total	423.37	20.42	402.96	0

Note-se que a tabela Price acima foi criada usando a taxa de 1% e não a taxa real de 2.4227%. Com uma taxa de 1%, o valor final não será \$500, mas sim \$470.42. Para que a tabela seja impressa com a taxa real, esta deve ser calculada:

$$t, ni = getInterest(450, 400, 10)$$

- Taxa = 2.4227% - 15 iterações
- O preço à vista é igual ao preço total corrigido.
- Taxa Real = 2.4227%, Iterações = 15, Fator = 0.9000
- Preço à vista + juros de 2.42% ao mês = \$500.00
- Preço a prazo - juros de 2.42% ao mês = \$450.00
- Juros Embutidos = $(\$500.00 - \$450.00) / \$450.00 * 100 = 11.11\%$
- Desconto = $(\$500.00 - \$450.00) / \$500.00 * 100 = 10.00\%$
- Excesso = $\$450.00 - \$450.00 = \$-0.00$
- Excesso = $(\$500.00 - \$500.00) * 0.9000 = \$-0.00$
- Percentual pago a mais = -0.00%
- Coeficiente de Financiamento: 0.113803
- Valor financiado = $450.00 - 50.00 = 400.00$
- Prestação: \$50.00

Tabela Price					
Mês	Prestação	Juros	Amortização	Saldo Devedor	
1	50.00	9.69	40.31	359.69	
2	50.00	8.71	41.29	318.41	
3	50.00	7.71	42.29	276.12	
4	50.00	6.69	43.31	232.81	
5	50.00	5.64	44.36	188.45	
6	50.00	4.57	45.43	143.01	
7	50.00	3.46	46.54	96.48	
8	50.00	2.34	47.66	48.82	
9	50.00	1.18	48.82	-0.00	
Total	450.00	50.00	400.00	0	

1.3 Imprimindo a Tabela Price

A tabela Price⁹ indica o quanto se pagou de juros no período e o quanto foi amortizado do principal. Por exemplo, para um empréstimo de \$23.000 reais, a ser pago em 12 vezes, com uma taxa de juros de 4,55% ao mês, temos:

Mês	Prestação	Juros	Amortização	Saldo Devedor
n	R = pmt	J	U = pmt - J	PV - U
1	2529.54	1046.50	1483.04	21516.96
2	2529.54	979.02	1550.52	19966.44
3	2529.54	908.47	1621.07	18345.37
4	2529.54	834.71	1694.83	16650.54
5	2529.54	757.60	1771.94	14878.60
6	2529.54	676.98	1852.57	13026.03
7	2529.54	592.68	1936.86	11089.18
8	2529.54	504.56	2024.98	9064.19
9	2529.54	412.42	2117.12	6947.07
10	2529.54	316.09	2213.45	4733.62
11	2529.54	215.38	2314.16	2419.46
12	2529.54	110.09	2419.46	0.00
Total	30354.50	7354.50	23000.00	0

```
## Retorna a Tabela Price, também chamada de sistema francês
# de amortização.
# É um método usado em amortização de empréstimos cuja principal
# característica é apresentar prestações (ou parcelas) iguais.
# O método foi apresentado em 1771 por Richard Price em sua obra
# "Observações sobre Pagamentos Remissivos".
#
# @param np número de prestações.
# @param pv valor do empréstimo.
# @param t taxa de juros.
# @param pmt pagamento mensal.
# @return uma matriz cujas linhas são listas com:
# (mês, prestação, juros, amortização, saldo devedor).
#
def priceTable(np, pv, t, pmt):
```

Embora as prestações sejam fixas, ou seja, sabe-se exatamente o quanto vai ser pago a cada mês, se a duração do empréstimo for muito longa e com juros altos, como os cobrados pelo sistema bancário brasileiro, a amortização será muito lenta, e, por conseguinte, se pagará uma quantidade de juros exorbitante.

⁹https://pt.wikipedia.org/wiki/Tabela_Price

1.4 Disponibilizando Tudo na Internet

CGI é uma forma padrão de gerar conteúdo dinâmico para páginas Web. CGI significa *Common Gateway Interface* e provê uma interface padrão entre o servidor HTTP e programas que geram conteúdo Web. Esses programas são mais conhecidos como scripts CGI, e são escritos com uma linguagem script (em geral interpretada) qualquer, como Python ou Perl, por exemplo. O servidor HTTP processa esses arquivos script fonte, linha por linha, e chama as funções do CGI quando necessário. A saída de uma função CGI é enviada ao cliente Web (navegador) como parte de uma página Web.

Dessa forma, se utilizarmos o CGI para executar scripts Python no servidor¹⁰, tudo o que calculamos pode ser acessado por qualquer navegador e ser muito útil na avaliação de um empréstimo por uma pessoa com acesso à Internet¹¹.

Portanto, um script CGI como o do código 1, pode ser executado num servidor HTTP, para processar entradas fornecidas através de um formulário HTML.

```
#!/usr/bin/env python3
# coding: UTF-8
#
## CGI interface for CDC calculations.
# Should be copied to:
# - /Library/WebServer/CGI-Executables/ (MacOS)
# - /usr/lib/cgi-bin/ (Ubuntu)
# - /var/www/cgi-bin (Fedora)
#
# @author Paulo Roma
# @date 29/06/2020
#
# @see https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code
# @see http://lcg.ufrj.br/python/html/cdc.cgi?np=10&tax=0&pv=450&pp=500&e=0

# Import modules for CGI handling
import cgi
import cgitb

import sys
import os
import codecs
sys.path.append(os.environ.get("DOCUMENT_ROOT") + '/python/labs/hidden')
from _02f_rational import getInterest, CF, priceTable, printTable, \
    setDownPayment, getDownPayment

# When Python prints Unicode strings to the console it usually detects
# the console encoding and automatically encodes the Unicode strings
# using that encoding.
# For CGI there is no terminal so the default is ascii.
# One can change stdout to another encoding by using:
sys.stdout = codecs.getwriter('utf8')(sys.stdout.buffer)

# we want some error messages
```

¹⁰https://www.tutorialspoint.com/python/python_cgi_programming.htm

¹¹<http://orion.lcg.ufrj.br/python/html/cdc.cgi?np=12&tax=4.55&pv=23000&pp=30500>

```

cgitb.enable()

# Create instance of FieldStorage
form = cgi.FieldStorage()

errmsg = ""

# Get data from fields
try:
    np = int(form.getvalue('np'))
    t = float(form.getvalue('tax')) / 100
    pp = float(form.getvalue('pp'))
    pv = float(form.getvalue('pv'))
except Exception as err:
    errmsg = "Invalid Parameters: {}".format(err)

print("""Content-type:text/html;charset=utf-8\r\n\r\n
    <html>
    <h6>{}</h6>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>CDC - Crédito Direto ao Consumidor CGI Program</title>
    </head>
    <body>
        """.format("Python {0[0]}.{0[1]}.{0[2]} - Encoding: {1}"
            .format(sys.version_info[:3], sys.getdefaultencoding()))

if errmsg:
    print(errmsg)
    sys.exit(1)

dp = len(form.value) <= 4

print("""
    <h4>Parcelas: {}{:d}</h4>
    <h4>Taxa: {:.2f}%</h4>
    <h4>Preço a Prazo: ${:~.2f}</h4>
    <h4>Preço à Vista: ${:~.2f}</h4>
    """.format("1+" if dp else "", np - 1 if dp else np, 100 * t, pp, pv))

setDownPayment(dp) # com ou sem entrada

i = getInterest(pp, pv, np)
if t <= 0:
    t = i[0] * 0.01
cf = CF(t, np)
pmt = pv * cf
if getDownPayment():
    pmt /= (1 + t)
    np -= 1 # uma prestação a menos
    pv -= pmt # preço à vista menos a entrada
    print("<h4>Valor financiado = ${:~.2f} - ${:~.2f} = ${:~.2f}</h4>".
        format(pv + pmt, pmt, pv))

```



```

print("""
    <h4>Taxa Real ({} iterações): {:.4f}%</h4>
    <h4>Coeficiente de Financiamento: {:.6f}</h4>
    <h4>Prestação: ${:.2f}</h4>
    <h2>Tabela Price</h2>
    """.format(i[1], i[0], cf, pmt))

ptb = priceTable(np, pv, t, pmt)

table = ""
for i in range(len(ptb)):
    table += "<tr>"
    for j in range(len(ptb[0])):
        col = "{:.2f}".format(ptb[i][j]) if i > 0 and \
            j > 0 else str(ptb[i][j])
        table += ("<td style='text-align:center'>" if i > 0 else "<th>") + \
            col + ("</td>" if i > 0 else "</th>")
    table += "</tr>"

print("""
<table border=1>
{content}
</table>
""".format(content=table))

if False:
    # html suppresses spaces
    print("<pre>")
    # monospaced font
    print("<code>")

    printTable(ptb)

    print("</code>")
    print("</pre>")

print("</body>")
print("</html>")

```

Código 1: cdc.cgi

1.5 Criando o Seu Próprio Servidor HTTP

É relativamente simples criar um servidor HTTP (*Hypertext Transfer Protocol*) local em MacOS¹², ¹³ ou Linux¹⁴. Isso vai facilitar muito os testes do seu código Javascript ou Python.

Ambos os sistemas possuem um servidor Apache¹⁵ pré-instalado ¹⁶ que pode ser ativado. Aconselho atribuir um nome ao servidor, por exemplo, o nome do seu computador, e em Linux, redirecionar o *Document Root* para fora da raiz (/) do sistema, colocando-o por exemplo em `/home/html`.

Em MacOS, os arquivos de configuração estão em `/etc/apache2`. Basicamente, é necessário editar o arquivo `httpd.conf` e no diretório `extra`, os arquivos `httpd-ssl.conf` e `httpd-vhosts.conf` para configurar o módulo `ssl`. Opcionalmente, criar arquivos para cada usuário no diretório `users`, se for necessário permitir o acesso ao diretório `Sites` (equivalente ao `public_html` em Linux) no home de cada usuário, que poderá ser acessado como: `http://localhost/~user`. O *Document Root* do Apache2 está em `/Library/WebServer/Documents`. Finalmente, para reiniciar o servidor Apache, basta executar:

```
sudo apachectl restart
```

Em Ubuntu, é possível criar um host virtual em `/etc/apache2/sites-available`, usando o arquivo `000-default.conf` como ponto de partida. Desta forma, não é necessário alterar o arquivo de configuração global em `/etc/apache2/apache2.conf`. Algo como `cederj.conf`:

```
<VirtualHost *:8088>                                # porta para o host virtual
    ServerName cederj                                # nome do host
    ServerAlias cederj.local

    ServerAdmin roma@localhost                        # email do administrador
    DocumentRoot /home/html                          # onde ficam os arquivos html
    <Directory "/home/html">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
        AddDefaultCharset utf-8                      # queremos acentos
        # usar CGI scripts fora dos diretórios ScriptAliased
        AddHandler cgi-script .cgi
    </Directory>

    .....
</VirtualHost>
```

Para ativar este novo arquivo, deve-se fazer:

```
sudo a2ensite cederj.conf
```

¹²<https://discussions.apple.com/docs/DOC-3083>

¹³<https://jasonmccreary.me/articles/install-apache-php-mysql-mac-os-x-catalina/>

¹⁴<https://phoenixnap.com/kb/how-to-install-apache-web-server-on-ubuntu-18-04>

¹⁵<https://apache.org>

¹⁶Em Ubuntu: `sudo apt-get install apache2`.

Para ativar o servidor SSL, deve-se fazer:

```
sudo a2ensite cederj-ssl.conf
```

Depois, deve-se ativar os módulos CGI e SSL:

```
sudo a2enmod cgi
```

e

```
sudo a2enmod ssl
```

Para reler os arquivos de configuração do servidor Apache no Ubuntu 18.04 basta usar¹⁷:

```
sudo apache2ctl graceful
```

Em caso de erro, pode-se verificar a validade dos arquivos de configuração:

```
sudo apachectl configtest
```

O Diretório padrão para colocar os *scripts* é:

- /Library/WebServer/CGI-Executables, para macOS;
- /usr/lib/cgi-bin, para Ubuntu e
- /var/www/cgi-bin, para Fedora.

Para poder colocar os *scripts* num diretório qualquer, basta adicionar um arquivo `.htaccess`, com o seguinte conteúdo:

```
Options +ExecCGI
AddHandler cgi-script .py
```

Se a sua distribuição Linux usar SELinux¹⁸ (*Security-Enhanced Linux*), deve-se alterar o contexto dos arquivos python, para que possam ser executados:

```
sudo chcon -t httpd_user_script_exec_t *.py
```

É possível também ativar o servidor seguro que utiliza o módulo `ssl` para criptografar o tráfego. A configuração está em `/etc/apache2/mods-available/ssl.conf`. No entanto, isso dá um pouco mais de trabalho, porque será necessário criar chaves RSA para identificar o servidor. Neste curso, isso é absolutamente dispensável. O script pode ser acessado como `http://localhost:porta/cgi-bin/cdc.cgi` ou `http://cederj:8088/cgi-bin/cdc.cgi`. A porta padrão 80 pode ser omitida e os argumentos podem ser passados da seguinte forma:

```
http://localhost/cgi-bin/cdc.cgi?np=12&tax=4.55&pv=23000&pp=30500
```

É possível acessar o seu servidor de fora da Intranet (rede local), desde que se saiba o IP externo da conexão, e que o seu provedor de Internet não utilize CGNAT¹⁹. Para ter acesso ao IP externo, pode-se utilizar um serviço de redirecionamento qualquer, como o do **no-ip**²⁰.

¹⁷<https://www.configserverfirewall.com/ubuntu-linux/restart-apache-ubuntu-18/>

¹⁸SELinux pode controlar que atividades um sistema permite para cada usuário, processo e daemon, com especificações muito precisas.

¹⁹<https://www.copeltelecom.com/site/blog/conheca-o-cgnat/>

²⁰<https://www.noip.com/>