



Python

Paixão à primeira edição

Para aqueles que ainda não me conhecem...



Vamos falar de quê ?

- Mandamentos do Desenvolvedor
- Comparação entre linguagens de programação
- Python: Como e Por quê ?
- Aplicações Práticas (c/ - d 10 linhas)
- Casos de Uso no Mundo e no Brasil

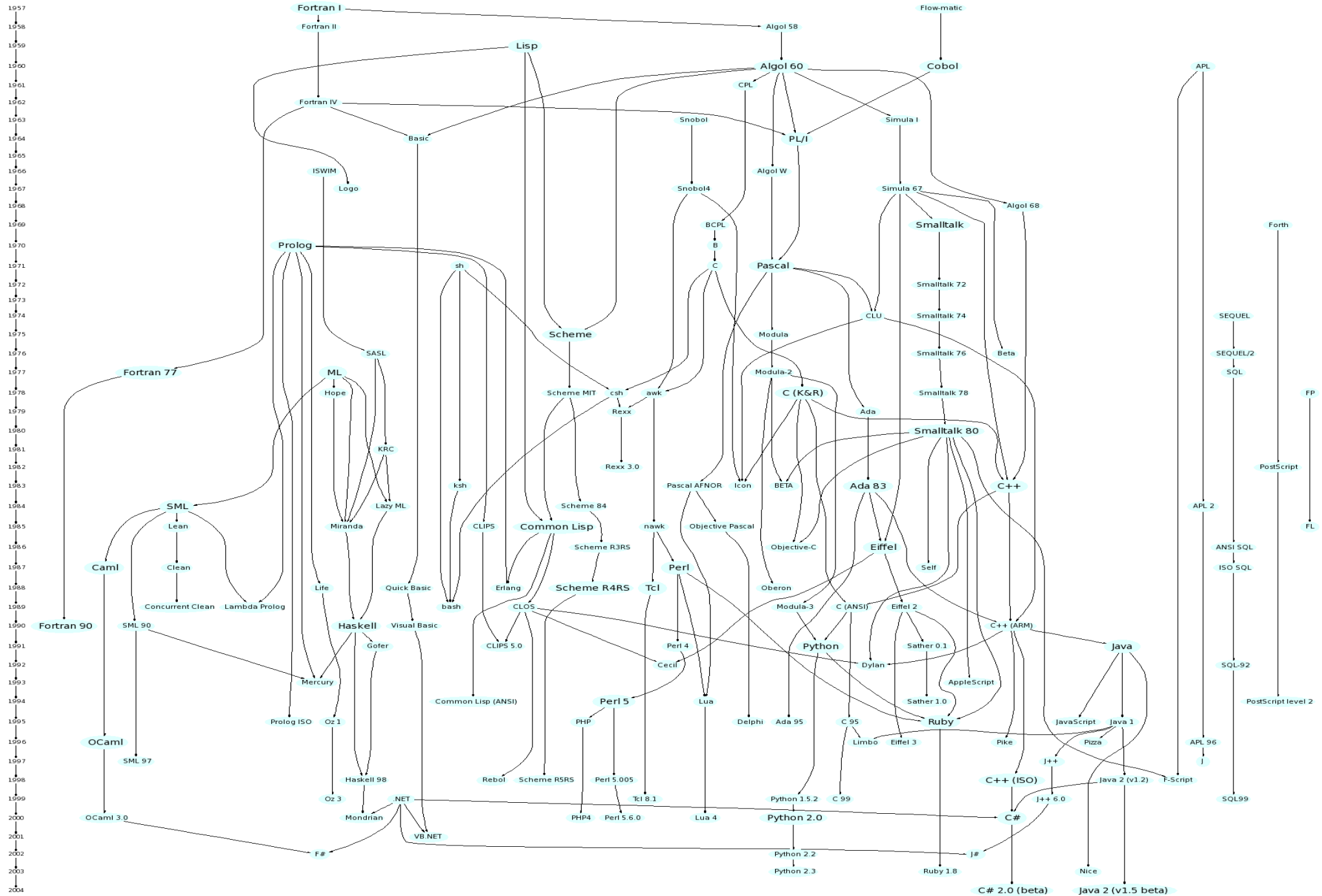


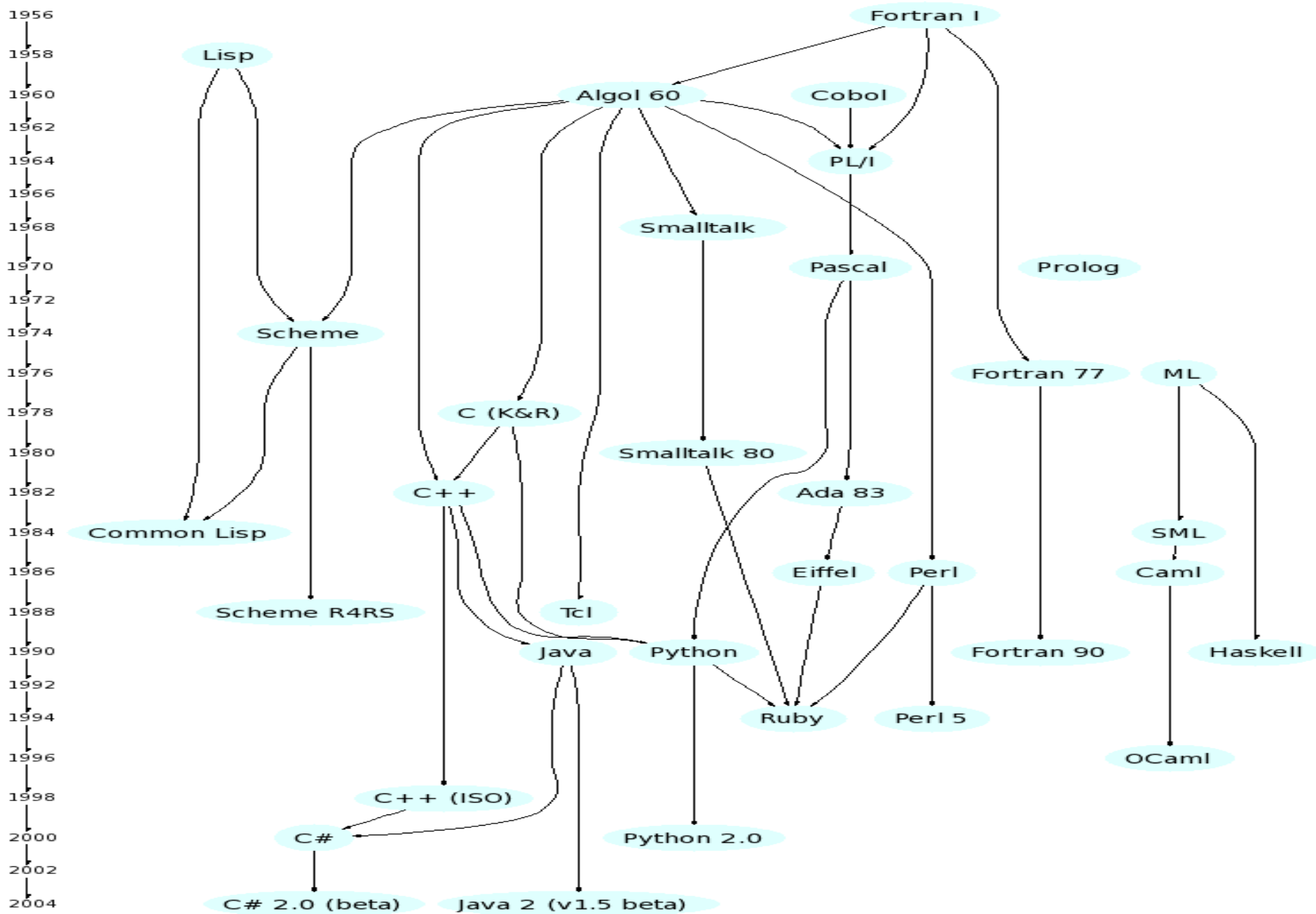
Mandamentos do Desenvolvedor

- Rapidez no ciclo de desenvolvimento
- Qualidade
- Portabilidade
- Customização
- Extensibilidade
- Interoperabilidade
- Escalabilidade



5º Fórum Internacional do Software Livre





Fonte: <http://people.mandrakesoft.com/~prigaux/language-study>

Duelos entre linguagens



The Good, the Bad and the Ugly

Duelos entre linguagens

Tcl/Tk

```
proc main {} {  
    set count 0  
    while {[gets stdin line]> 0} {  
        incr count $line  
    }  
    puts $count  
}
```

Perl

```
use integer;  
sub main {  
    shift;  
    while(<>) { $count += $_ }  
    print "$count\n";  
}
```

Python

```
import sys  
def main():  
    count = 0  
    for line in sys.stdin.readlines():  
        count += int(line)  
    print count
```


Duelos entre linguagens

Java

```
public static void main(String[] args) {
    int count = 0;
    String line;
    try {
        BufferedReader in = new BufferedReader(
            new InputStreamReader(System.in));
        while ((line = in.readLine()) != null) {
            count = count + Integer.parseInt(line);
        }
    } catch (IOException e) { }
    System.out.println(Integer.toString(count));
}
```

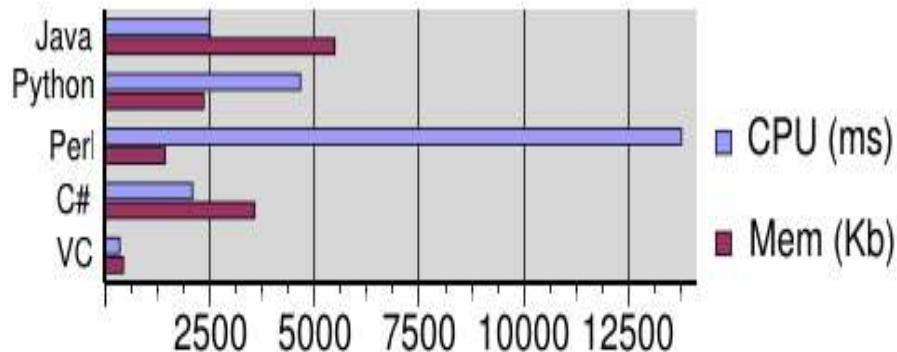
Duelos entre linguagens

C#

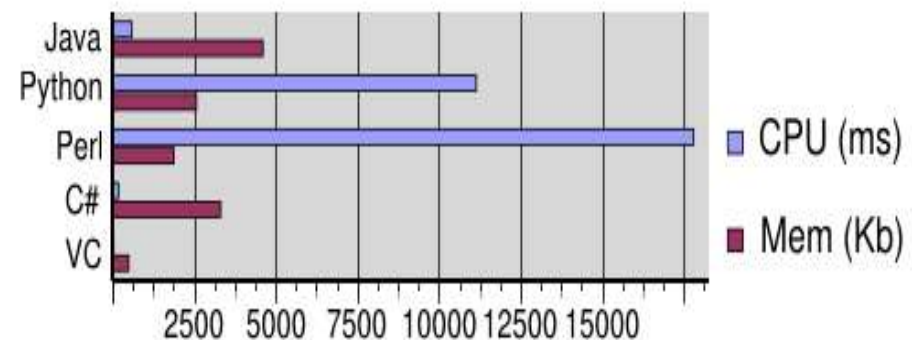
```
public static int Main(String[] args) {  
    int sum = 0;  
    for (String line = Console.In.ReadLine();  
        line != null;  
        line = Console.In.ReadLine())  
    { sum += System.Convert.ToInt32(line); }  
    Console.WriteLine(sum.ToString() + "\n");  
    return(0);  
}
```

Duelos entre linguagens: Desempenho

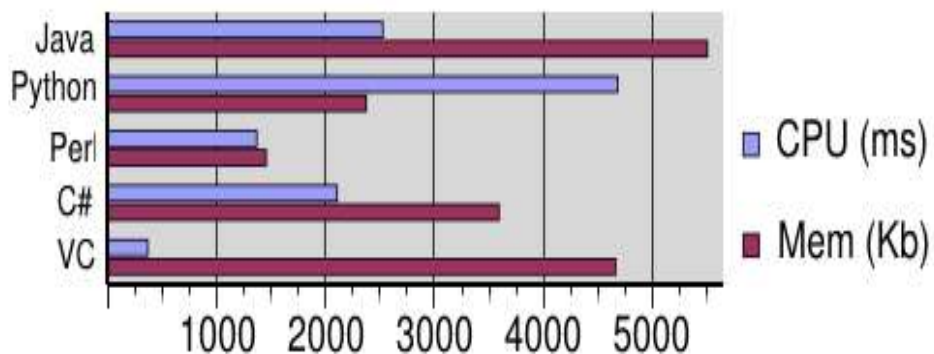
Soma de Inteiros N=1000



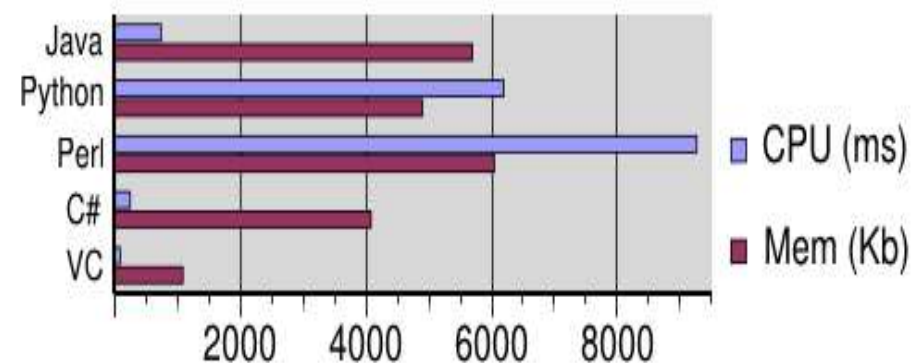
Acesso em Vetor N=7000



Espalhamento N=80000

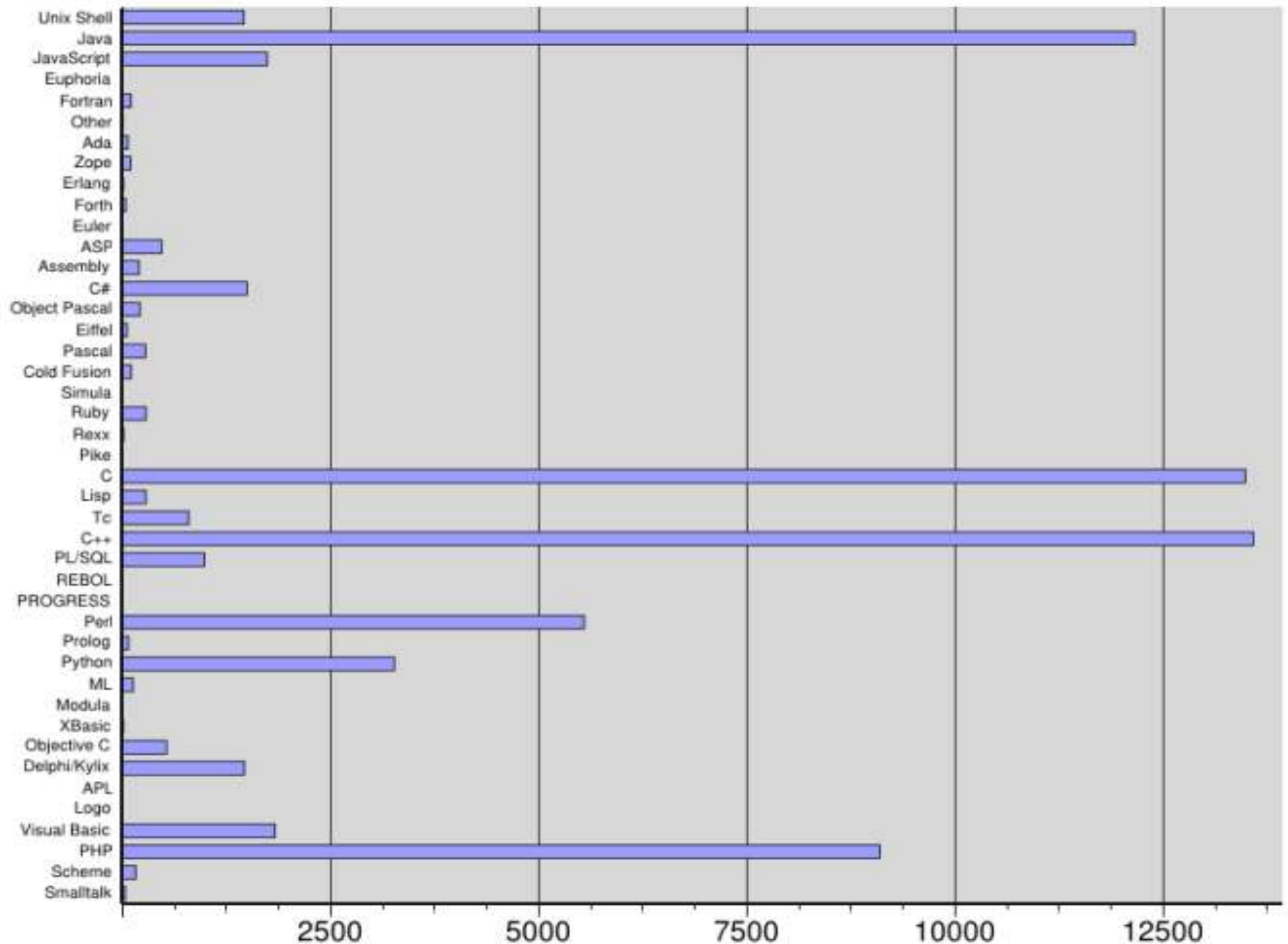


Heapsort N=80000



Fonte: <http://www.bagley.org/~doug/shootout/>

5º Fórum Internacional do Software Livre



Python é uma linguagem . . .

- Interpretada
- Orientada a objetos
- Tipagem e **semântica dinâmicas**
- Poderosas estruturas de dados **nativas**
- **Extensa** biblioteca padrão
- **Fácil** aprendizagem e manutenção
- Disponível livremente **sem custo**
- Multi-plataforma
- **Interativa**
- Divertida



Banco de Dados

psycopg (Postgres)

```
from psycopg import *  
  
o = connect("dbname=teste user=fulano")  
c = o.cursor()  
c.execute("SELECT * FROM nome_da_tabela")  
row = c.fetchone()  
print row  
  
l = [ ('valor_1', 'fulano', 0), ('valor_2', 'beltrano', 5), ('valor_3', 'ciclano', 15)]  
c.executemany("INSERT INTO outra_tabela VALUES (%s, %s, %s)", l)  
  
c.commit()
```

Networking

Twisted (time server)

```
from twisted.internet.protocol import Protocol, Factory
from twisted.internet import reactor
from time import asctime

class TimeServer(Protocol):
    def dataReceived(self, data):
        self.transport.write(asctime())

def main():
    f = Factory()
    f.protocol = TimeServer
    reactor.listenTCP(8000, f)
    reactor.run()

main()
```

WebServices

SOAPpy

```
from SOAPpy import SOAPProxy
from SOAPpy import Types

google = SOAPProxy( 'http://api.google.com/search/beta2',
                    'urn:GoogleSearch')

results = google.doGetCachedPage('Pd6ke...ArQoX' ,
                                 'www.owls.org')

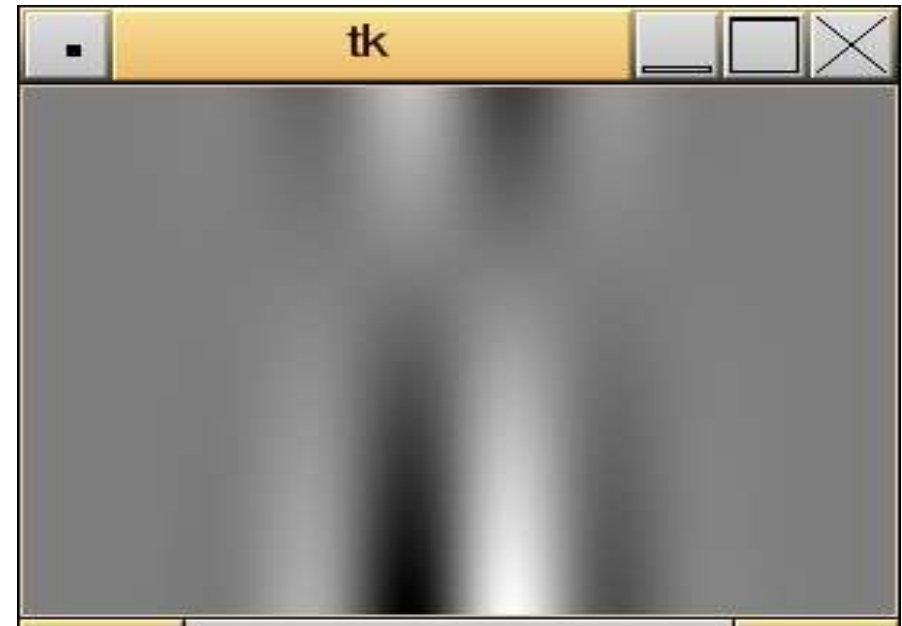
of = open('cached_page_response.html', 'w')
of.write(results)
of.close()
```


Scientific Python

Numeric-23.1 (testview.py)

```
Import view
from Numeric import *

x = arange(-3, 6, .04)
y = arange(-12, 12, .08)
y = sin(y)*exp(-y*y/18.0)
z = x * y[:,NewAxis]
view.view(z)
```



Graphical User Interface

Tkinter



```
from Tkinter import *

class Test(Frame):
    def __init__(self, master=None):
        Frame.__init__(self, master)
        Pack.config(self)
        self.Gpanel = Frame(self, width='1i', height='1i', background='green')
        self.Gpanel.pack(side=LEFT)
        self.Gpanel.QUIT = Button(self.Gpanel, text='QUIT',
                                   foreground='red', command=self.quit)
        self.Gpanel.QUIT.pack(side=LEFT)

test = Test()
test.master.title('tkinter demo')
test.mainloop()
```

Processamento de Imagens

PIL – Python Image Library

```
from PIL import Image, ImageDraw, ImageFont, ImageFilter, ImageChops
```

```
def make_vertical_label(label):  
    font = ImageFont.load("courB08.pil")  
    fontMask=Image.new("RGBA", (20,49), (10,144,216))  
    draw=ImageDraw.Draw(fontMask)  
    sizex,sizey=draw.textsize(label,font=font)  
    fontMask = fontMask.resize((sizex,sizey))  
    draw=ImageDraw.Draw(fontMask)  
    bimsize=(sizex, sizey); width,height=bimsize; xoff=yoff=0  
    draw.text((xoff,yoff), label,fill=(255,255,255),font=font)  
    fontMask = fontMask.rotate(90)  
    fontMask.save('output.png','PNG')
```

```
make_vertical_label("PIL é bacana")
```

Tecno-diversidade

- **Bancos de Dados** (ODBC,mySQL,Postgres,DCOracle,SQLServer,sqlite,gadfly,ZODB,PyDO,Metakit)
- **Networking** (asyncore,htmlib,SimpleHTTPServer,urllib,ftplib,poplib,smtplib,telnetlib,Twisted)
- **GUI's** (Tkinter,wxPython,PyGTK,PyQt,PyKDE,Pythonwin)
- **Ciência**(Numarray,SciPy,BioPython,AstroPy,Py2R/SPlus)
- **Proc. de Imagens** (PIL,PythonMagick,Gimp-python)
- **XML** (PyXML, 4Suite, ElementTree, RDFLib, Cwm)
- **Web** (ZOPe,Webware,Quixote,PSP,mod_python,Nevow)
- **IDE** (Emacs,vi, idle,SPE,Pythonwin,Komodo,BlackAdder)
- **Interoperabilidade** (Jython,ctypes,win32all,IronPython)

Fonte <http://www.python-eggs.org/links.html>.

Histórias de Sucesso (Mundo)

- **Philips** – controle da manufatura de semi-condutores
- **Industrial Ligth & Magic** – (desde 1996)
- United Space Alliance, **Nasa**
- **Nokia, Nortel**
- **Disney** – produção de animações
- **RedHat, Gentoo** - instalação, config, pacotes
- **Google, Yahoo**
- **Los Alamos National Laboratory** - controle de cluster
- **Apple** – pré-instalado no Mac OSX

Fonte <http://www.pythonology.org/success>

Histórias de Sucesso (Software Livre)

- **Blender** – animação e criação de jogos tempo-real
- **OpenOffice** – interfaceamento e automação PyUNO
- **Zope/Plone** – gerenciamento de conteúdo na Web
- **MoinMoin** – criação de sites Wiki
- **Mailman** – Gerenciador de listas de discussão GNU
- **BitTorrent** – P2P compartilhamento de arquivos
- **Chandler** (OSAF) – Gerenciamento de Info.Pessoais
- **Gimp** – Gimp-python alternativo ao Script-fu

Histórias de Sucesso (Brasil)

- **Serpro** – portais Web
- **Conectiva** – gerenciamento de pacotes e ferramentas
- **Embratel** – monitoramento do backbone
- Async, GPr, Haxent, Hiperlógica, Paradigma, Propus
- **Receita Federal** – IA aplicada a aduana (pesquisa)
- **Unicamp** – Susy : Sistema Submissão de Trabalhos

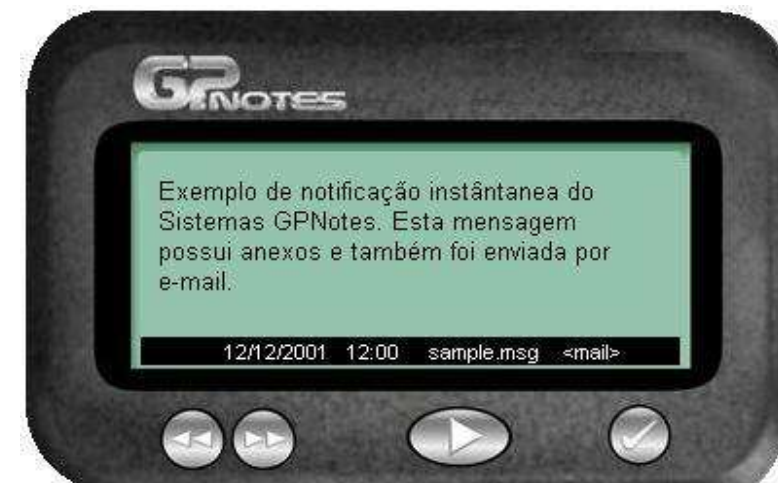
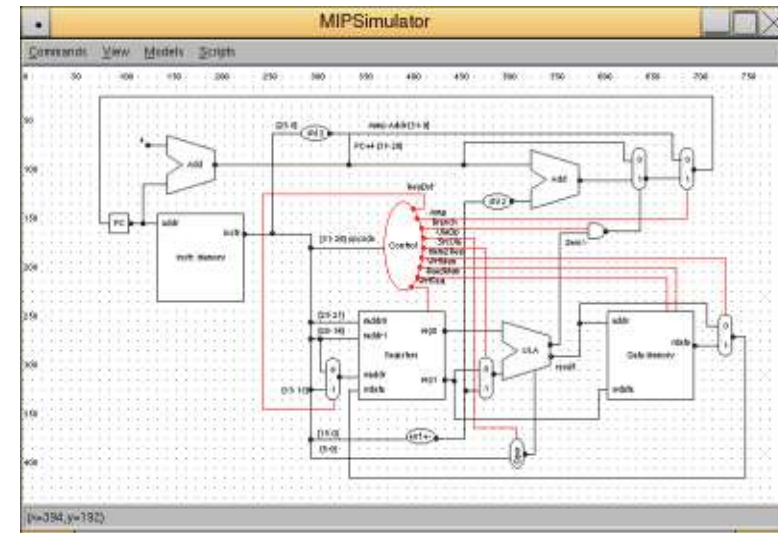
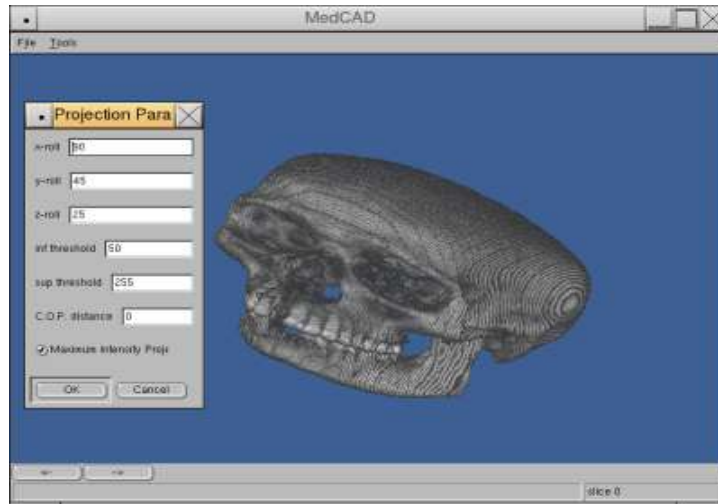
Fonte <http://www.pythonbrasil.com.br/moin.cgi/EmpresasPython>

Histórias de Sucesso (Pessoal)

- Instant Messenger (HP-UX/Win) – GPNotes
- **Caçador de Pornografia em Intranet** – GPView
- Transferidor de Arquivos scriptizável em XML - TRF3K
- Monitor de disponibilidade de servidores – Gpinger
- **Proxy TCP transparente** – PyProxy (em 40 min)
- Controle de tarefas - GPonto
- CRM – WebSAC
- Troubleshoot de tráfego – Excalibur
- Vizualizador 3D de Imagens Médicas – MedCAD
- Ensino de Arq.Computadores - **MIPSimulator**

Fonte <http://www.gpr.com.br>

Histórias de Sucesso (Pessoal)



Fonte <http://www.gpr.com.br>

Referências

- <http://www.python.org>
- <http://www.pythonbrasil.com.br/>
- <http://www.python-eggs.org/links.html>

>>> import this

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

Considerações Finais

- Python é uma linguagem moderna
- Python é um “canivete suíço”, poder e versatilidade
- Python é muito utilizada (Negócios/Pesquisa)
- Python continua crescendo e evoluindo
- Python oferece alta produtividade (baixo custo)
- Python é divertida

Python merece uma avaliação por qualquer desenvolvedor!

Obrigado a todos pela atenção!