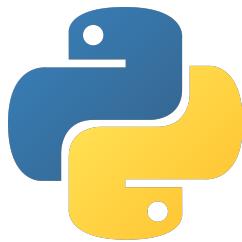


"If you want to teach people a new way of thinking, don't bother trying to teach them. Instead, give them a tool, the use of which will lead to new ways of thinking."
(R. Buckminster Fuller)

Lógica de programação aplicada à criação e análise da forma

Aula 01 - Introdução



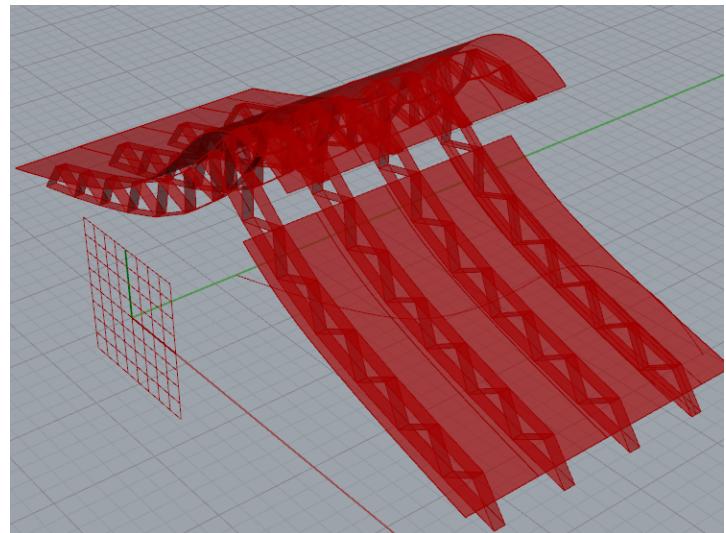
Fernando Ferraz Ribeiro
fernando.ribeiro@ufba.br
19/05/2017

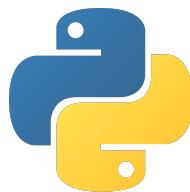
```
from __future__ import division
#importando bibliotecas do Rhinoceros:
#Rhino Common, rhinoscriptsyntax e ghpythonlib
from Rhino.Geometry import Point3d, Line, NurbsCurve
import rhinoscriptsyntax as rs
import ghpythonlib.components as gh

Tr_3D = []
Tr_copias = []
teste=[]
Cargas = []

if not Bz_1:
    Bz_1=Diag_1
if not Peso_esp_Tr:
    Peso_esp_Tr = 7800 #Peso específico do aço Kg/m^3
if not Peso_cobertura:
    Peso_cobertura = 20 #Sobrecarga do telhado Kg/m^2
if not Fator_de_conv:
    Fator_de_conv = 10
if not N_Tr:
    N_Tr = 1

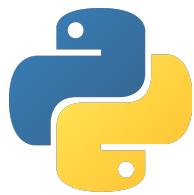
if not Plano:
    Plano = rs.WorldXYPlane()
#decompe o plano de trabalho nos componentes Origem e os eixos xyz
pOr, eX, eY, eZ = Plano
```





REGRAS DO LCAD

- 1. LABORATÓRIO EXCLUSIVO PARA AULAS DA FAUFBA. PERMITIDA A UTILIZAÇÃO COM A PRESENÇA DOS PROFESSORES E/OU MONITORES DAS DISCIPLINAS.**
- 2. ZELAR PELOS EQUIPAMENTOS DO LABORATÓRIO.**
- 3. TERMINANTEMENTE PROIBIDO DESCONECTAR OS EQUIPAMENTOS DA CPU: MOUSE, TECLADO, MONITOR, ETC.**
- 4. EVITAR O USO DE PEN DRIVES. UTILIZAR A INTERNET PARA A TRANSFERÊNCIA DE ARQUIVOS (GOOGLE DRIVE, DROPBOX, ONE DRIVE, GITHUB, ETC.**
- 5. APÓS UTILIZAR, DESLIGAR OS EQUIPAMENTOS. CPU, MONITOR E ESTABILIZADOR**
- 6. É PROIBIDO COMER E BEBER NO LABORATÓRIO. UTILIZA O INTERVALO PARA ISSO**
- 7. NÃO UTILIZAR O CELULAR NO PERÍODO DA AULA.**
- 8. RESPEITAR O HORÁRIO DA AULA E A TURMA EM QUE ESTÁ MATRICULADO.**



Definição de Algoritmos

<

>

Knuth (1997) define o conceito de algoritmo como um conjunto de regras que fornece uma sequência de operações para a solução de um problema específico, e que deve atender cinco requisitos:

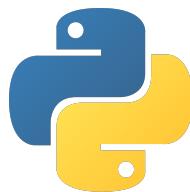
Finitude

Precisão

Entrada de Informação

Saída de Informação

Efetividade



ALGORITMOS

Exemplo de Algoritmo

Divisão de um segmento de reta em "N" partes iguais:

1 - Dado o segmento de reta AB desenhar, a partir do ponto A ,uma semi-reta não paralela a AB.

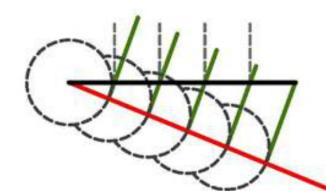
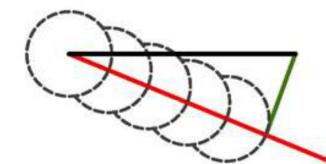
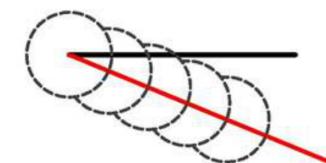
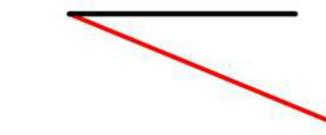
2 - Dado o número "N" de divisões, utilizando o compasso com uma mesma abertura qualquer, marcar "N" pontos equidistantes sobre a semi-reta Traçada no passo anterior.

3 - Traçar um segmento unindo o "N-ésimo" ponto marcado no passo anterior com a extremidade B do segmento AB

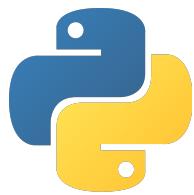
4 - Traçar paralelas ao segmento desenhado no passo 3, passando pelos pontos definidos no passo 2.

5- Fim

N=5 A ————— B

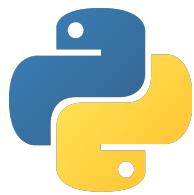


O segmento AB e o número inteiro "N" são as **variáveis de entrada**. Os Passos serão chamados de **operações** ou **funções**.

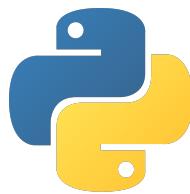


Relying upon software kills rather than mechanical ingenuity, graphics programmers first replicated the functions of traditional drafting instruments, and then went far beyond them. This has made a wider graphic vocabulary available to designers, together with a more elaborate syntax—in all, a richer and potentially more expressive graphic and spatial language.

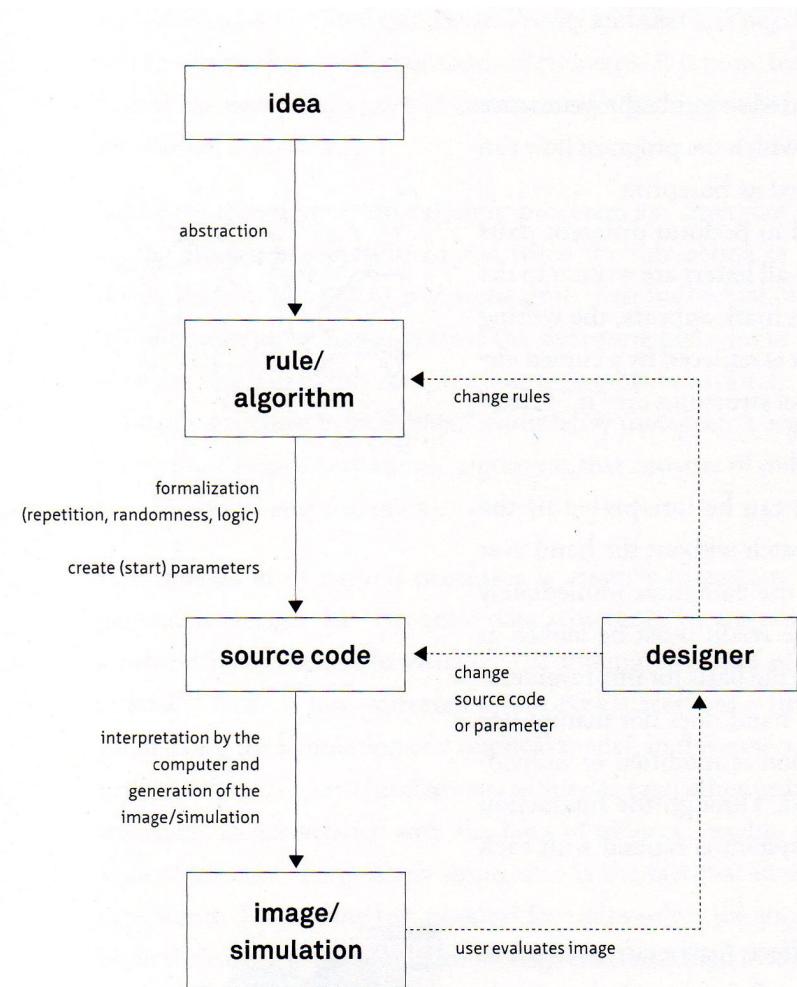
MITCHELL, W. J. World's Greatest Architect MAKING, MEANING, AND NETWORK CULTURE. Cambrige: MIT Press, 2008. v. 40



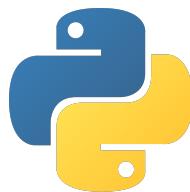
- 1. Software de projeto**
- 2. Ampliação de Software de projeto**
- 3. Automação de Etapas de projeto**
- 4. Auxílio na tomada de decisões em um projeto**
- 5. Geração da Forma**
- 6. Organização, manipulação e extração da informação em
projetos**



ALGORITMOS GENERATIVOS

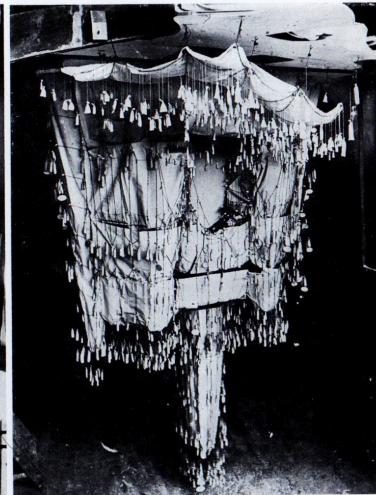
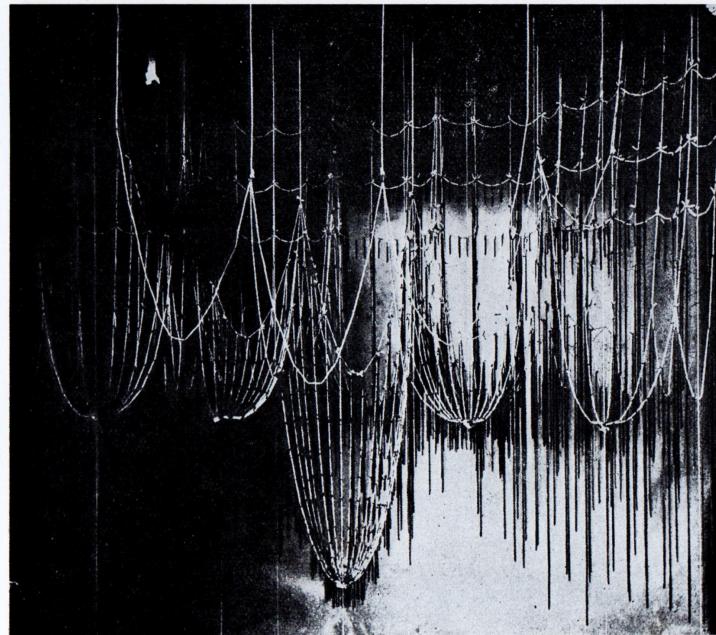


(Bohnacker et al. 2012)



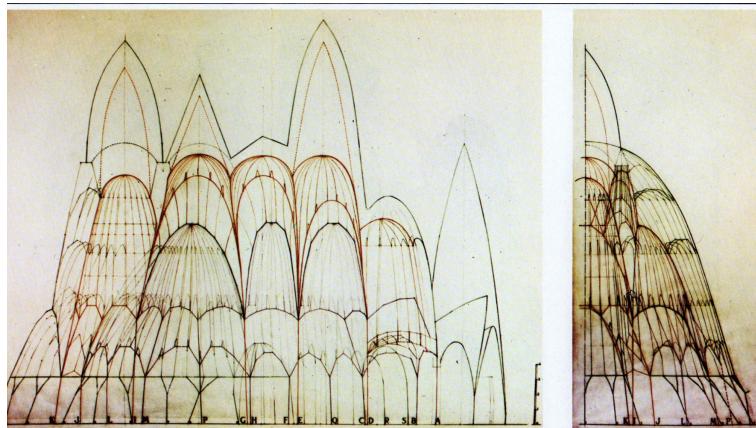
ALGORITMOS GENERATIVOS - ABORDAGEM CORRELATA

< 8 de 22 >

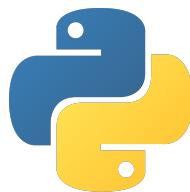


26. Maqueta funicular revestida. / Maqueta funicular revestida. / Draped funicular model.

25. Maqueta funicular revestida amb paper en plà d'estudi. / Maqueta funicular revestida con papel, en plan de estudio. / Funicular model hung with paper, in the course of study.



PUIG-BOADA, I.; ARIBAU, A. L'eglesia de la colonia Güell. Barcelona: Lumen, 1976.



ALGORITMOS GENERATIVOS

ALGORITMO DAS CÚPULAS MÓVEIS

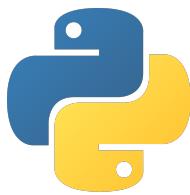
As Cúplas do Hospital Hospital Sarah Kubitschek - RJ e as do TRT-Ba, do Arquiteto João (Lelé) Filgueiras Lima, serviram de inspiração para o algoritmo apresentado a seguir.



TRT - Ba



Hospital Sarah - RJ



ALGORITMOS GENERATIVOS

< 10 de 22>

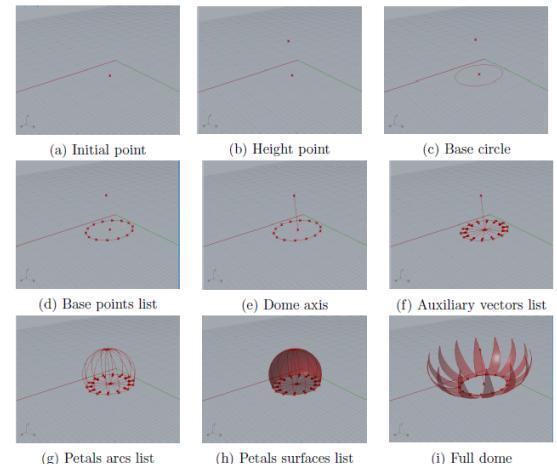
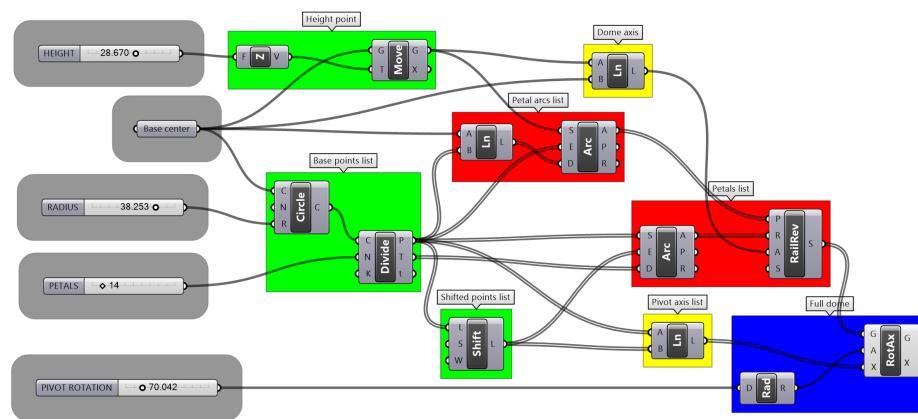
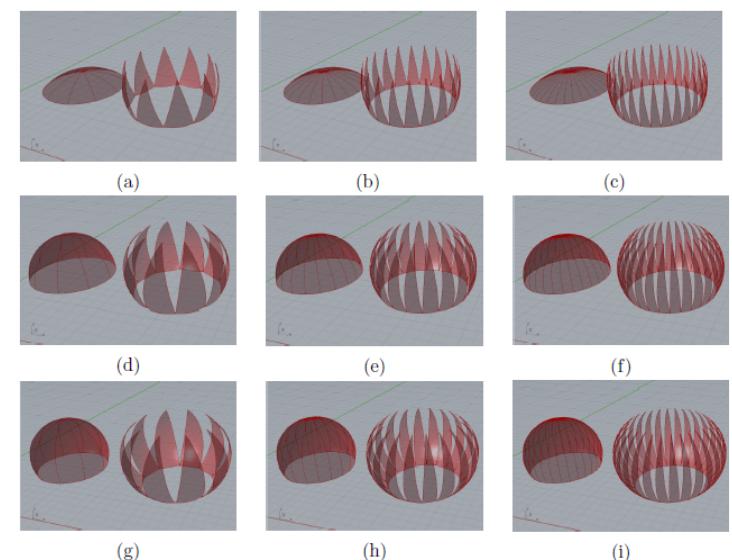
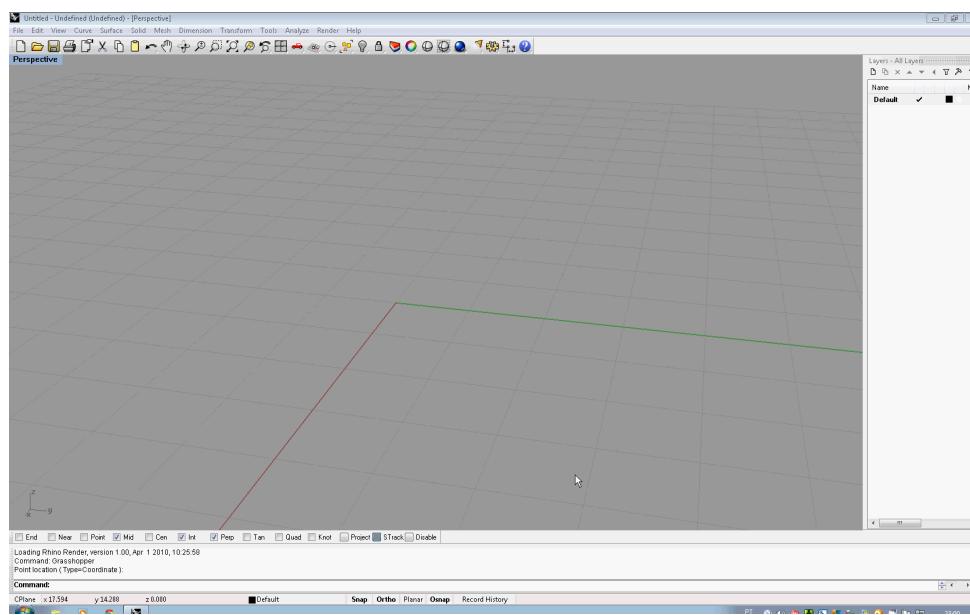
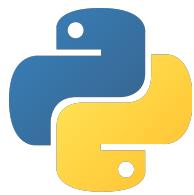


Figure 9: Steps of the list manipulation-based algorithm





História da Programação

< 11 de 22 >

Precursors



Máquina de Anticítera
(87 AC)



Wilhelm Schickard
(1592-1635)



Blaise Pascal
(1623-1662)



Gottfried Wilhelm Leibniz
(1646-1726)

Calculadoras Mecânicas



Joseph Marie Jacquard
(1752-1834)

Tear mecânico com leitura automática de cartões

O Motor Analítico

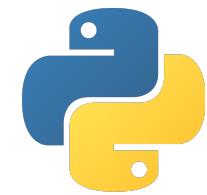


Ada Lovelace
(1815-1852)



Charles Babbage
(1792-1871)

Criadores do primeiro computador programável



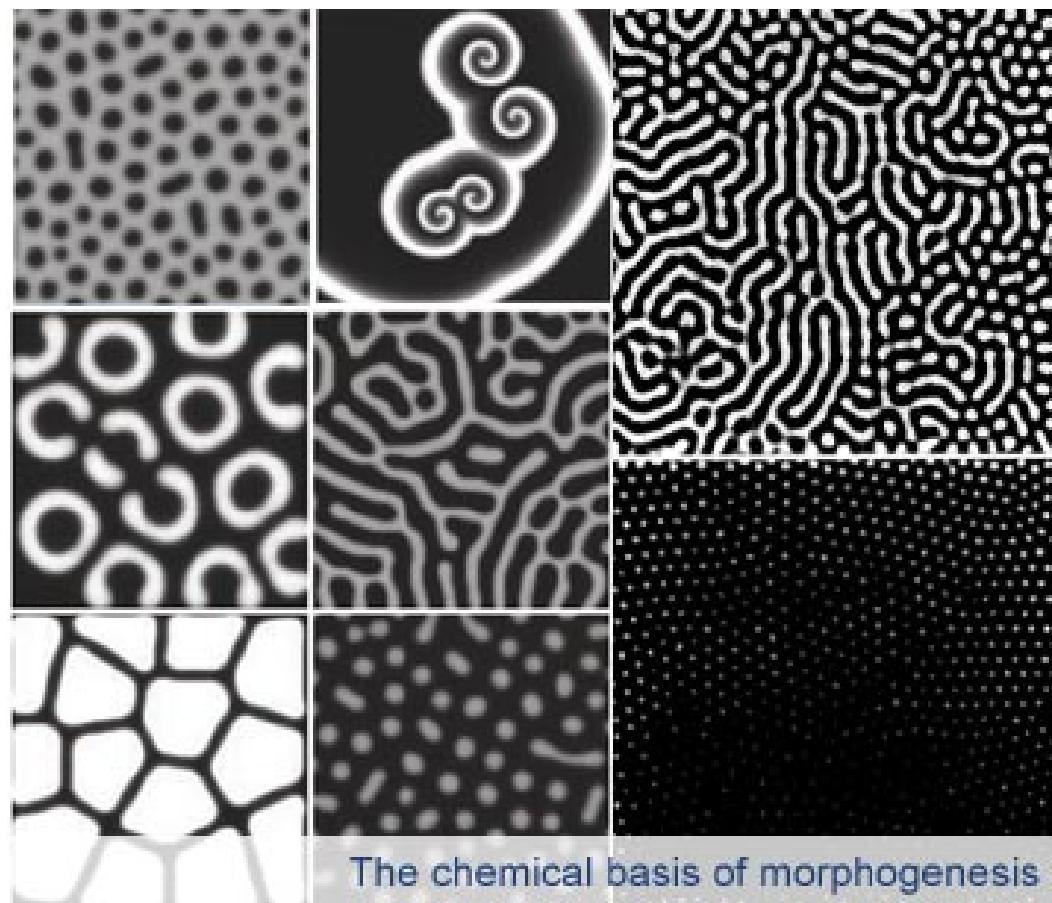
Padrões de Turing - 1952

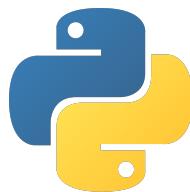


Alan Turing
1912 - 1954



Replica do computador Bombe (Alan Turing - 1939)





História da Programação

< 13 de 22>

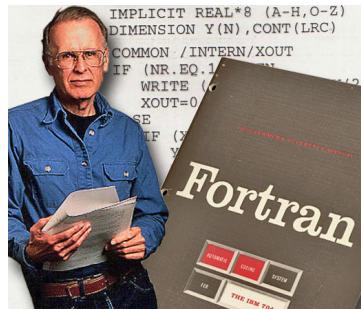
Flow-Matic / COBOL 1955

```
//COBCLGL JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELWORD EXEC COBCLGL,PARM.COB='MAP,LIST,LET'
//COBSYSIN DD *
 001 IDENTIFICATION DIVISION.
 002 PROGRAM-ID. 'HELLO'.
 003 ENVIRONMENT DIVISION.
 004 CONFIGURATION SECTION.
 005 SOURCE-COMPUTER. IBM-360.
 006 OBJECT-COMPUTER. IBM-360.
 0065 SPECIAL-NAMES.
 0066 CONSOLE IS CNSL.
 007 DATA DIVISION.
 008 WORKING-STORAGE SECTION.
 009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
 075 PROCEDURE DIVISION.
 090 000-DISPLAY.
 100      DISPLAY HELLO-CONST UPON CNSL.
 110      STOP RUN.
//LKED.SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR
//           DD DSNAME=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYOUT=A
```



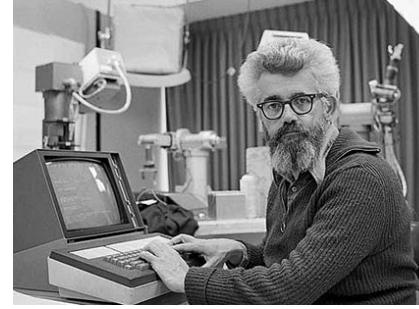
Grace Hopper (1906 -1992)

FORTRAN 1954/1957



John Backus
(1924 -2007)

LISP
1958



John McCarthy (1927 -2011)

C
1972

```
# include <stdio.h>

typedef struct Pessoa
{
    char nome[64]; // vetor de 64 chars para o nome
    unsigned short int idade;
    char rg[13];
} Pessoa;

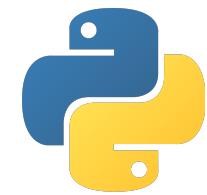
int main()
{
    Pessoa exemplo = {"Felipe", 16, "00.000.000-00"};

    printf("Nome: %s\n", exemplo.nome);
    printf("Idade: %hu\n", exemplo.idade);
    printf("RG: %s\n", exemplo.rg);

    getchar();
    return 0;
}
```

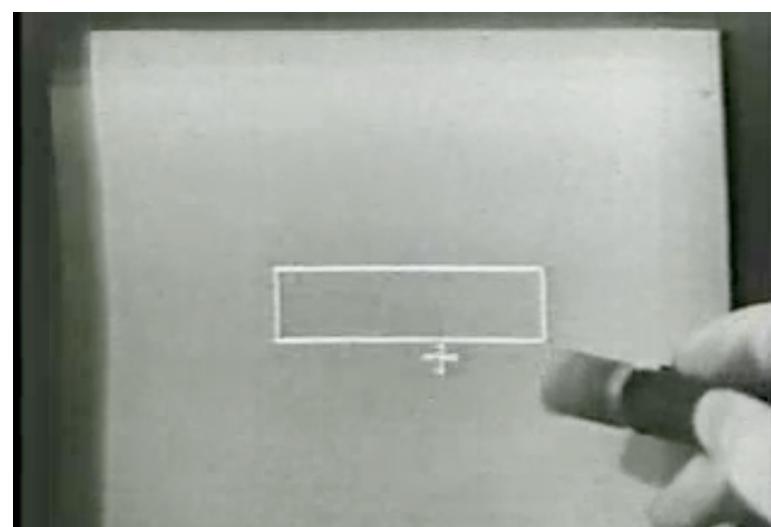
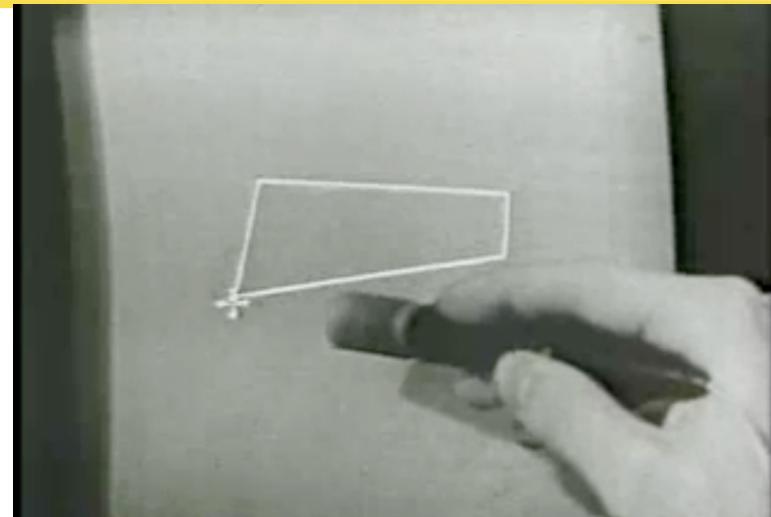


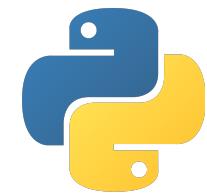
Dennis Ritchie
(1941 -2011)



Ivan Sutherland – Sketchpad protótipo de CAD 1962

< >

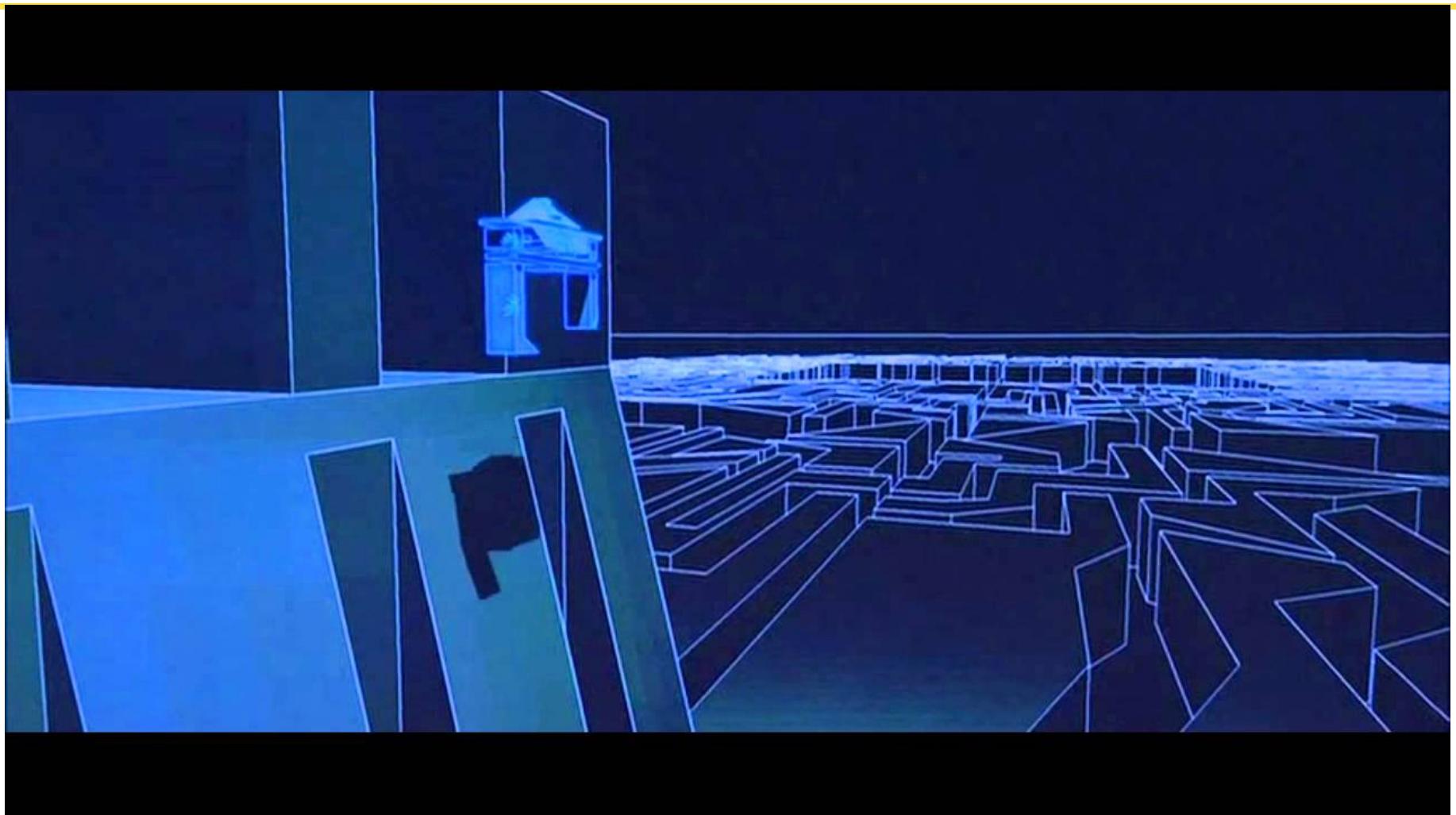


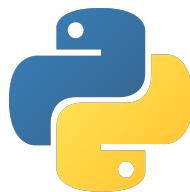


TRON - 1982

<

>

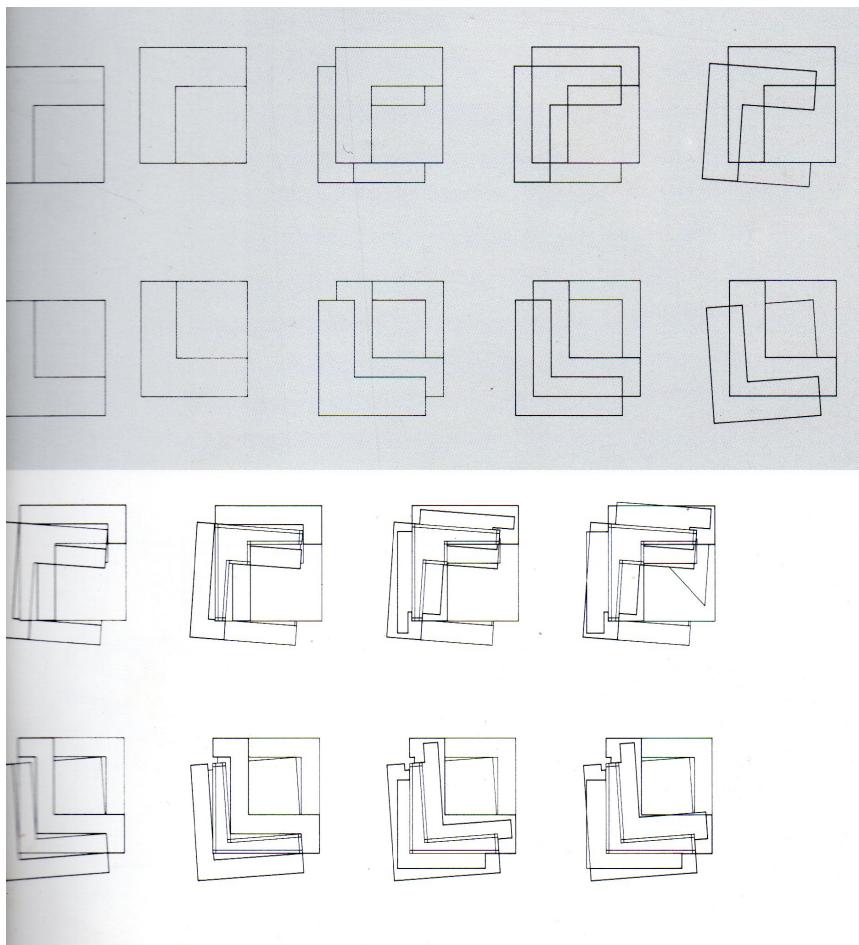




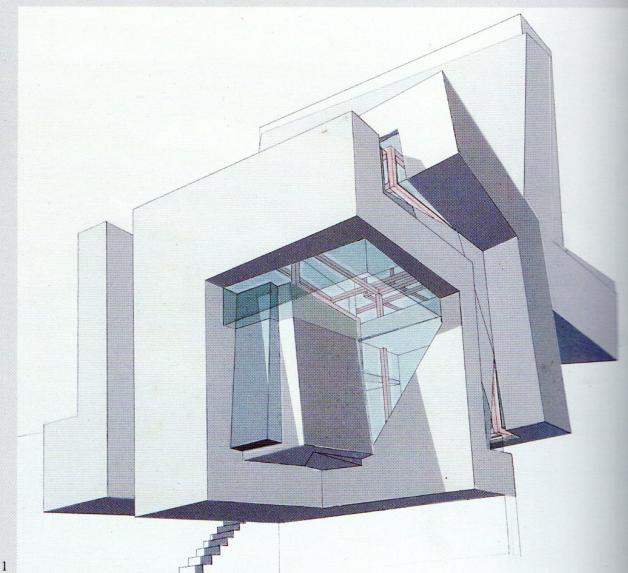
MANIPULAÇÃO DE VOLUMES

<

>

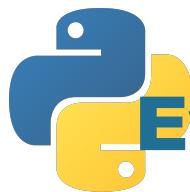


This house can be seen as the manifestation of a receptacle in which traces of logic and irrationality are intrinsic components of the object/place. It exists between the natural and the rational, between logic and chaos; the arabesque. It breaks the notion of figure/frame, because it is figure and frame simultaneously. Its tangential el-shapes penetrate three planes, always interweaving. These fluctuating readings resonate in the material of this house which, unlike a traditional structure of outside and inside, neither contains nor is contained. It is as if it were constructed of a substance which constantly changes shape.



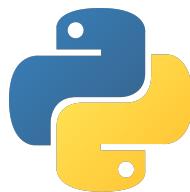
1

GUARDIOLA HOUSE
Peter Eisenman(1988)



Relying upon software skills rather than mechanical ingenuity, graphics programmers first replicated the functions of traditional drafting instruments, and then went far beyond them. This has made a wider graphic vocabulary available to designers, together with a more elaborate syntax—in all, a richer and potentially more expressive graphic and spatial language.

MITCHELL, W. J. World's Greatest Architect MAKING, MEANING, AND NETWORK CULTURE. Cambrige: MIT Press, 2008. v. 40



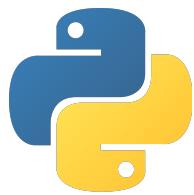
Linguagem de programação interpretada, multi-paradigma, multi-plataforma, orientada a objetos, imperativa, funcional, procedural, reflectiva e de código aberto.

Criada em 1989/1991 por Guido van Rossum

"I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (a government-run research lab in Amsterdam) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus). "

Guido van Rossum





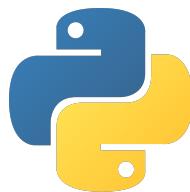
Python



“ O conjunto de peças de xadrez Bauhaus de Josef Hartwig é um exemplo de design excelente: bonito, simples e claro. Guido von Rossum, filho de um arquiteto e irmão de um designer de fontes premiado, criou uma obra-prima do design de linguagens. Adoro ensinar Python porque ela é bonita, simples e clara.”



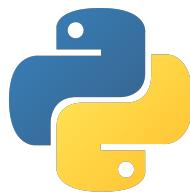
Luciano Ramalho. Python Fluente
2016



Python - IEEE Spectrum 2016

< 20 de 22 >

Language Rank	Types	Spectrum Ranking
1. C	📱💻CHIP	100.0
2. Java	🌐📱💻	98.1
3. Python	🌐💻	98.0
4. C++	📱💻CHIP	95.9
5. R	💻	87.9
6. C#	🌐📱💻	86.7
7. PHP	🌐	82.8
8. JavaScript	🌐📱	82.2
9. Ruby	🌐💻	74.5
10. Go	🌐💻	71.9

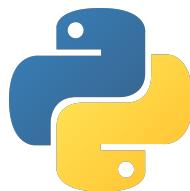


Python - IEEE Spectrum 2017

< 21 de 22 >

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.5
4. C++		97.1
5. C#		87.7
6. R		87.7
7. JavaScript		85.6
8. PHP		81.2
9. Go		75.1
10. Swift		73.7

<https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

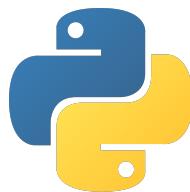


Python - IEEE Spectrum 2018

< 22 de 22 >

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4
6. PHP		84.9
7. R		82.9
8. JavaScript		82.6
9. Go		76.4
10. Assembly		74.1

<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>



Filosofia do Python

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.

Principais influencias:

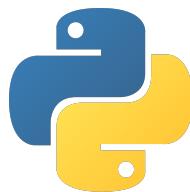
ABC
ALGOL 68
C
C++
Java
Lisp
Modula-3
Perl

Inspirou:
Boo
Cobra
D
F#
Falcon
Go
Groovy
JavaScript,
Julia
Nim
Ruby
Swift

Dialeto oficial: CPython

Desenvolvido e mantido por:

The Python Fondation - www.python.org



Vantagens e Desvantagens

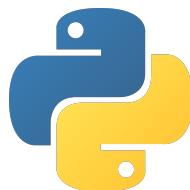
< 24 de 22>

Vantagens:

- é uma linguagem interpretada
- fácil de aprender
- fácil de usar interativamente
- sintaxe clara, elegante e legível
- fácil testar e modificar componentes
- mais eficiente do que um interpretador clássico
- programa 2x - 10x menores que implementados em C / C++ ou Java
- pode ser facilmente integrada/expandida com C/C++

Desvantagens:

- é uma linguagem interpretada
- gera programas mais lentos que implementados em C / C++ ou Java



Python em programas Cad, GIS e 3D

< 25 de 22>



Blender



Maya



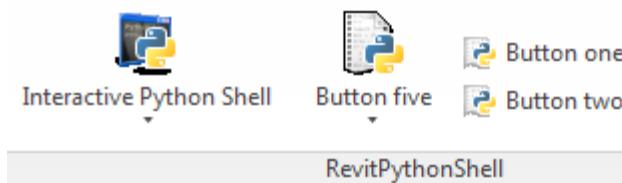
Rhino 3D



Grasshopper



FreeCad



Revit



ArcGIS



QGIS

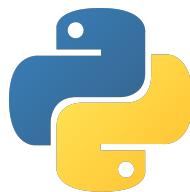


Unit 7: Python

Dynamo



FUSION



Outras Aplicações

< 26 de 22>



INDUSTRIAL
LIGHT & MAGIC
A LUCASFILM COMPANY

