

SSC0800 - Introdução à Ciência de Computação I

Conceitos de Computação: Algoritmos e Programação

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

Baseado no material dos profs Fernando S. Osório e Claudio F.M. Toledo

O que vamos aprender hoje?



Objetivos

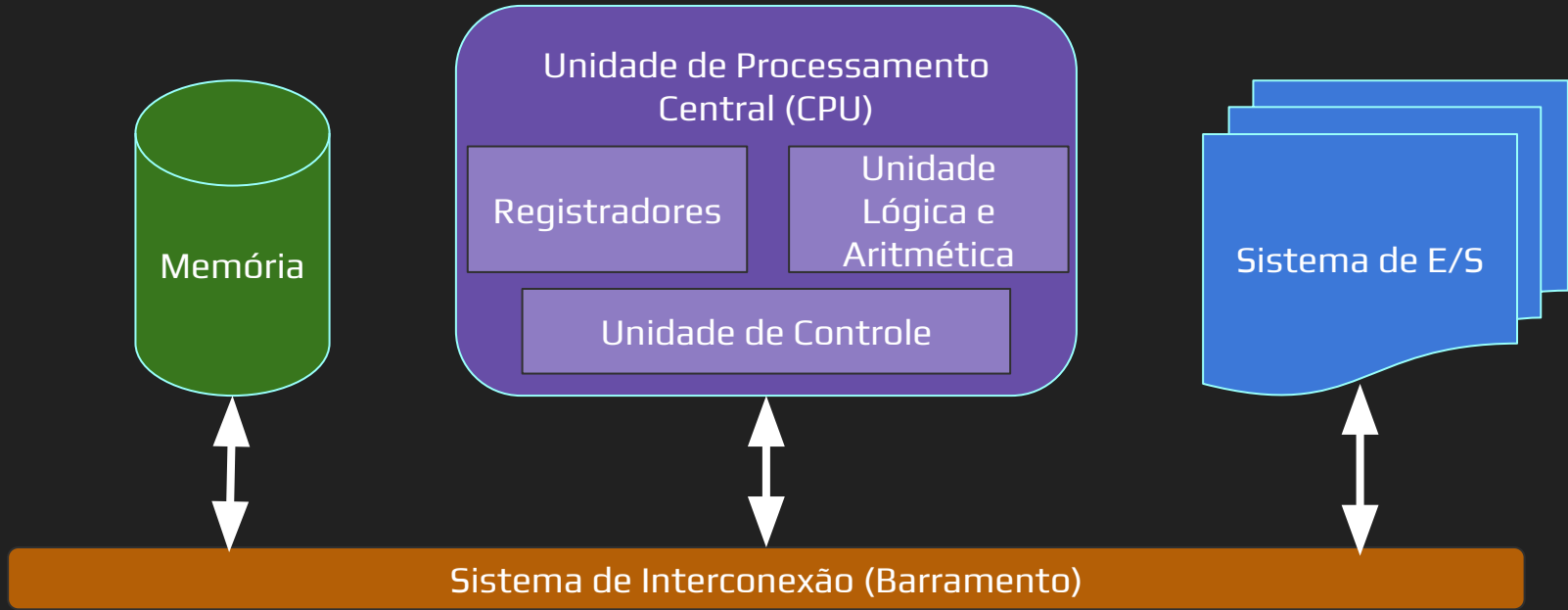
- Conhecer computadores e como programá-los
- Entender pensamento computacional e como usá-lo para resolver problemas
- Demonstrar o poder do Python para Ciência de Dados e Estatística

Tópicos da Aula

- Como funciona um computador
- Pensamento computacional e solução de problemas
- Um pouco do que se pode fazer com Python



Como funciona um computador



Arquitetura de von Neumann. Fonte: [1, 2]

A Arquitetura de von Neumann

- A mais conhecida arquitetura de computador
- Unidade de processamento central (CPU)
 - ◆ Busca, interpreta e executa instruções
 - ◆ Controla outros componentes
- Memória
 - ◆ Armazena dados e instruções

A Arquitetura de von Neumann

- Sistemas de Entrada e Saída (E/S) ou (I/O)
 - ◆ Comunicação externa (ambiente operacional)
- Sistema de Interconexão
 - ◆ Comunicação interna (entre os componentes internos)

Componentes de um computador

→ Processadores

- ◆ CPU, controladores, co-processadores
- ◆ Possuem conjunto de instruções
 - Operam sobre instruções e dados
- ◆ CPU
 - Propósito geral
- ◆ Co-processadores
 - e.g.: um co-processador aritmético para floats

Componentes de um computador

→ Memória

- ◆ Interna e Externa

- ◆ Custo relacionado à capacidade de armazenamento e velocidade de operação

→ Dispositivos de E/S

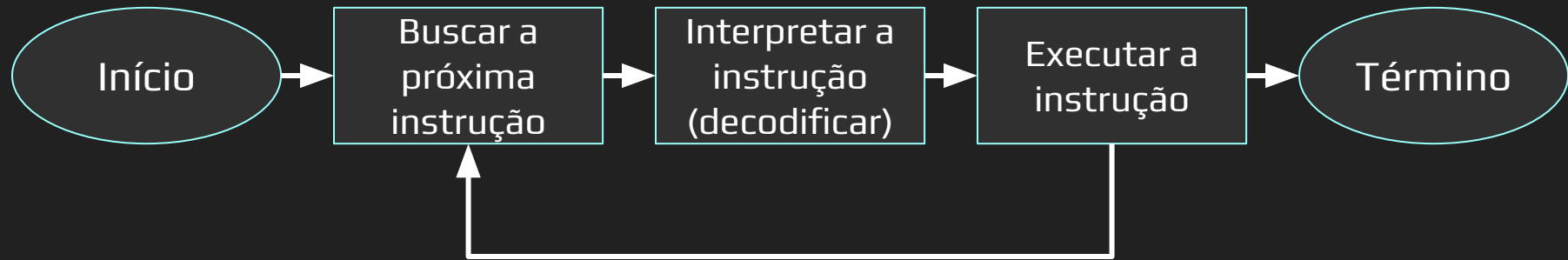
- ◆ Conversores de representação física de dados

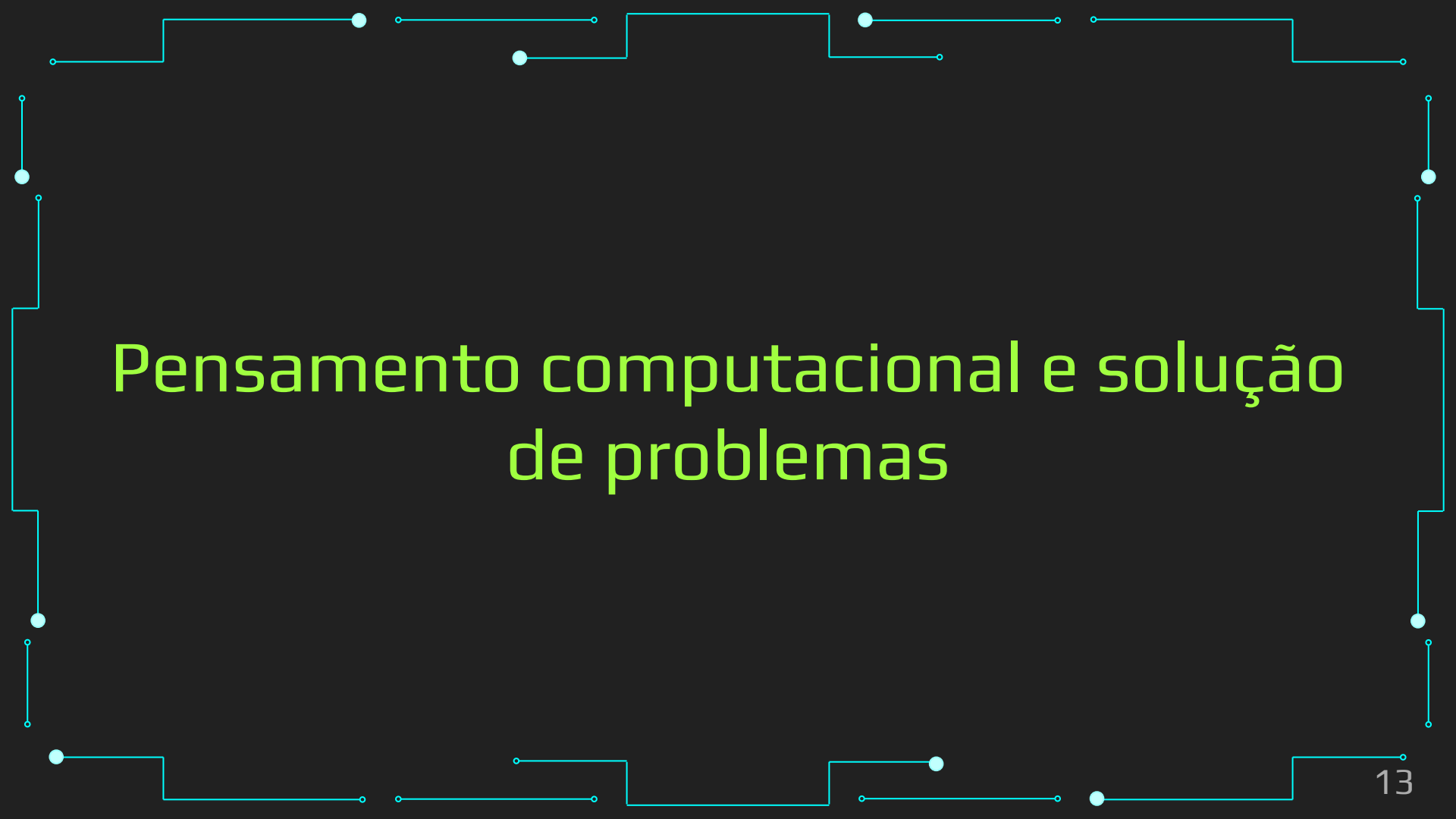
- ◆ Lentos em comparação aos processadores

Componentes de um computador

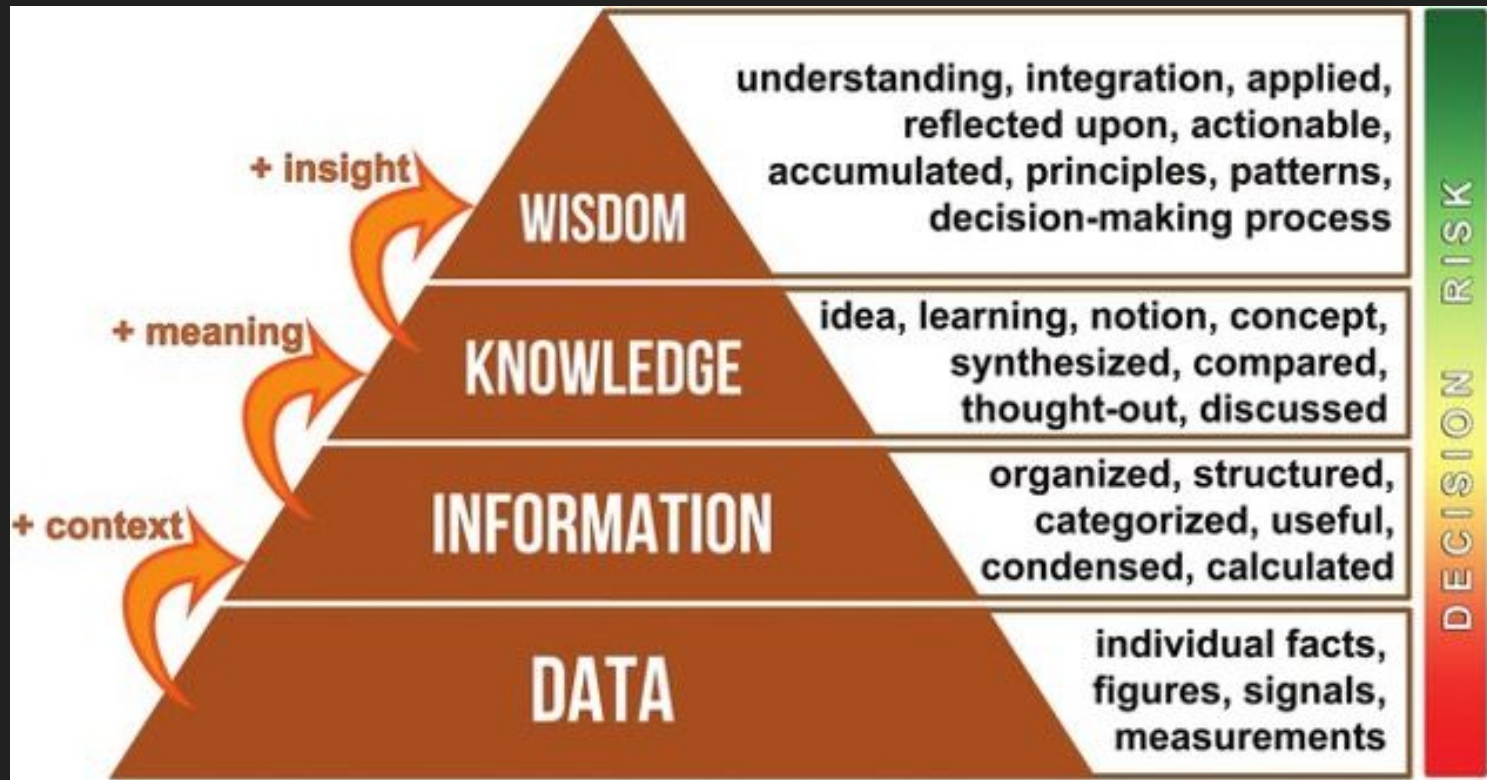
- Meios de interconexão
 - ◆ Comunicação entre componentes por meio de barramentos e slots
 - ◆ Disputa pelos recursos compartilhados

Ilustração básica de um programa





Pensamento computacional e solução de problemas



Pirâmide DIKW. Fonte: [3]

Dados, informação, conhecimento, e sabedoria

→ Dados

◆ Valores brutos armazenados (ex: 8)

→ Informação

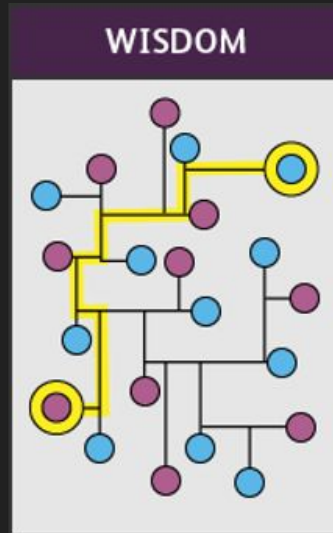
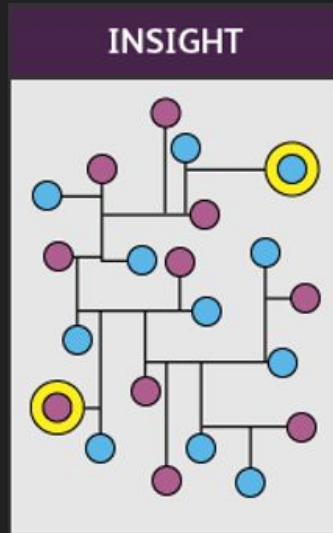
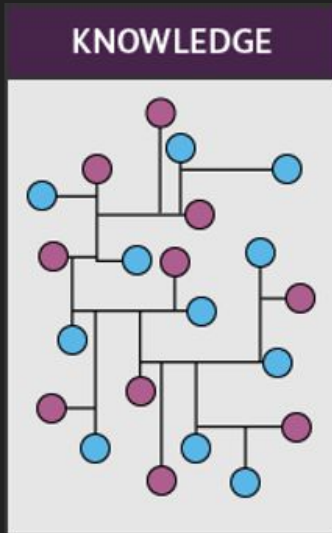
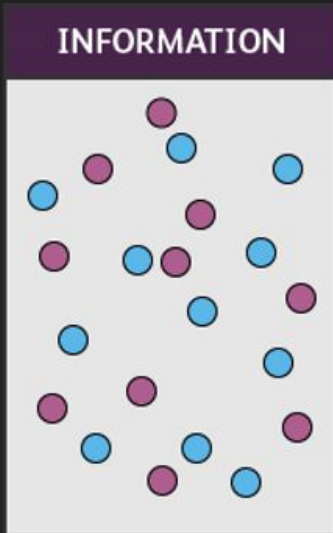
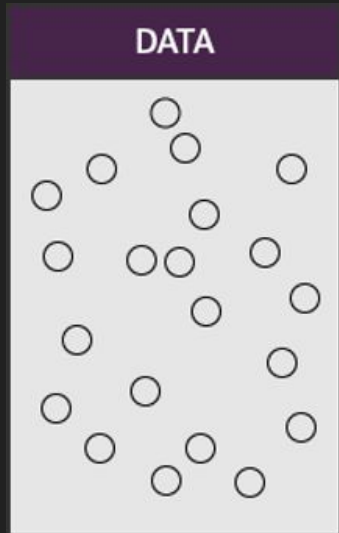
◆ Quando atribuímos um sentido ao dado

● Ex: nota da prova = 8

→ Conhecimento

◆ Regras, políticas, manipulação da informação

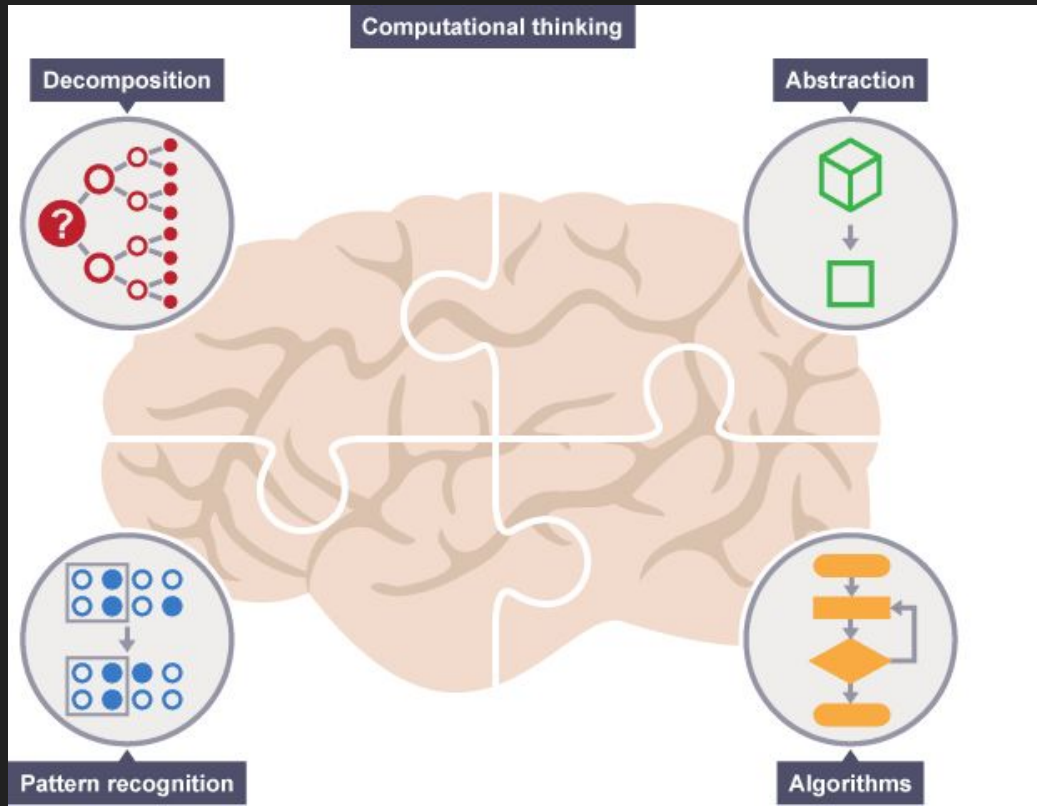
● Ex: notas devem estar entre 0 e 10



DIKIW. Fonte: [4]

Dados, informação, conhecimento, e sabedoria

- Computador manipula informações em sua memória
- Instruções:
 - ◆ Comandam o funcionamento da máquina e determinam a maneira como os dados devem ser tratados
- Dados:
 - ◆ Informação que deve ser manipulada



Pensamento computacional. Fonte: [5]

Pensamento Computacional [5]

- Decomposição
 - ◆ Quebrar um exemplo complexo ou sistema em pedaços menores, mais fáceis de manipular
- Reconhecimento de Padrões
 - ◆ Procurar por similaridades entre e dentro dos problemas

Pensamento Computacional [5]

→ Abstração

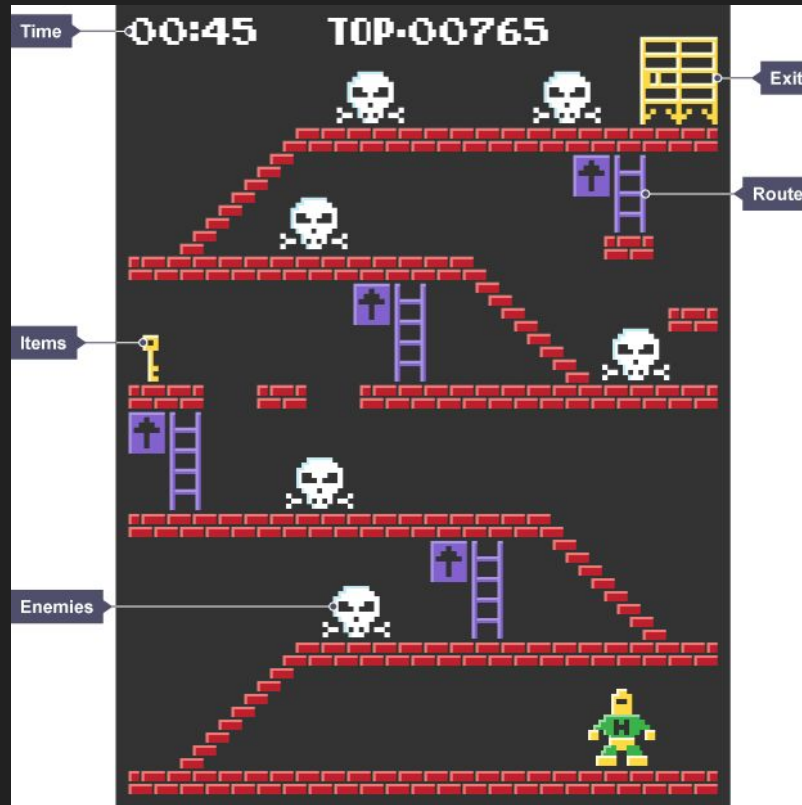
- ◆ Focar apenas nas informações importantes
- ◆ Ignorando detalhes irrelevantes

→ Algoritmos

- ◆ Desenvolver uma solução passo-a-passo para o problema
- ◆ Ou as regras a seguir para solucionar o problema

Pensamento Computacional [5]

- Ex: Um jogo
- Que itens é preciso coletar?
- Como coletá-los?
- Quanto tempo tenho para coletar?
- Onde está a saída?
- Qual a melhor rota para alcançá-la no menor tempo?
- Quais inimigos existem?
- Quais suas fraquezas?



Como ganhar? Fonte: [5]

Pensamento Computacional [5]

→ Decomposição:

◆ Como completar o nível?

→ Abstração:

◆ Focar no local da saída

→ Reconhecimento de padrões:

◆ Conhecimento de outros jogos

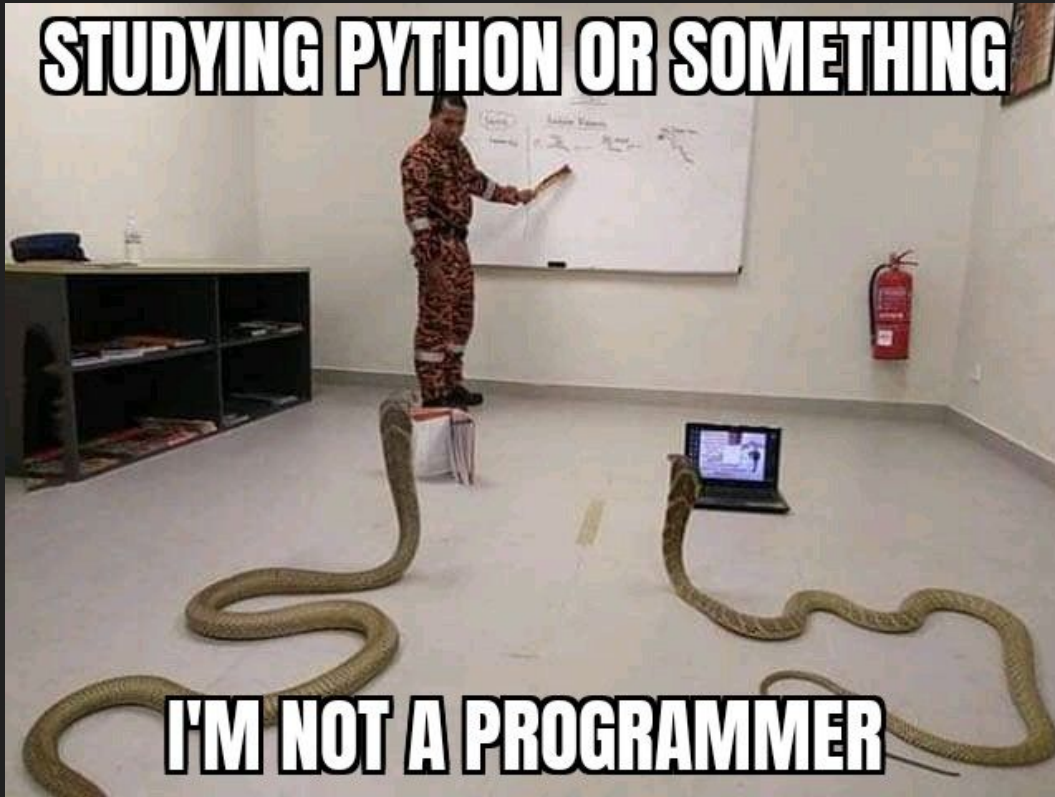
→ Algoritmos:

◆ Plano passo a passo



O que podemos fazer com Python?

STUDYING PYTHON OR SOMETHING



I'M NOT A PROGRAMMER

Aprendendo Python. Fonte:

https://www.reddit.com/r/memes/comments/imwpg9/learning_python/

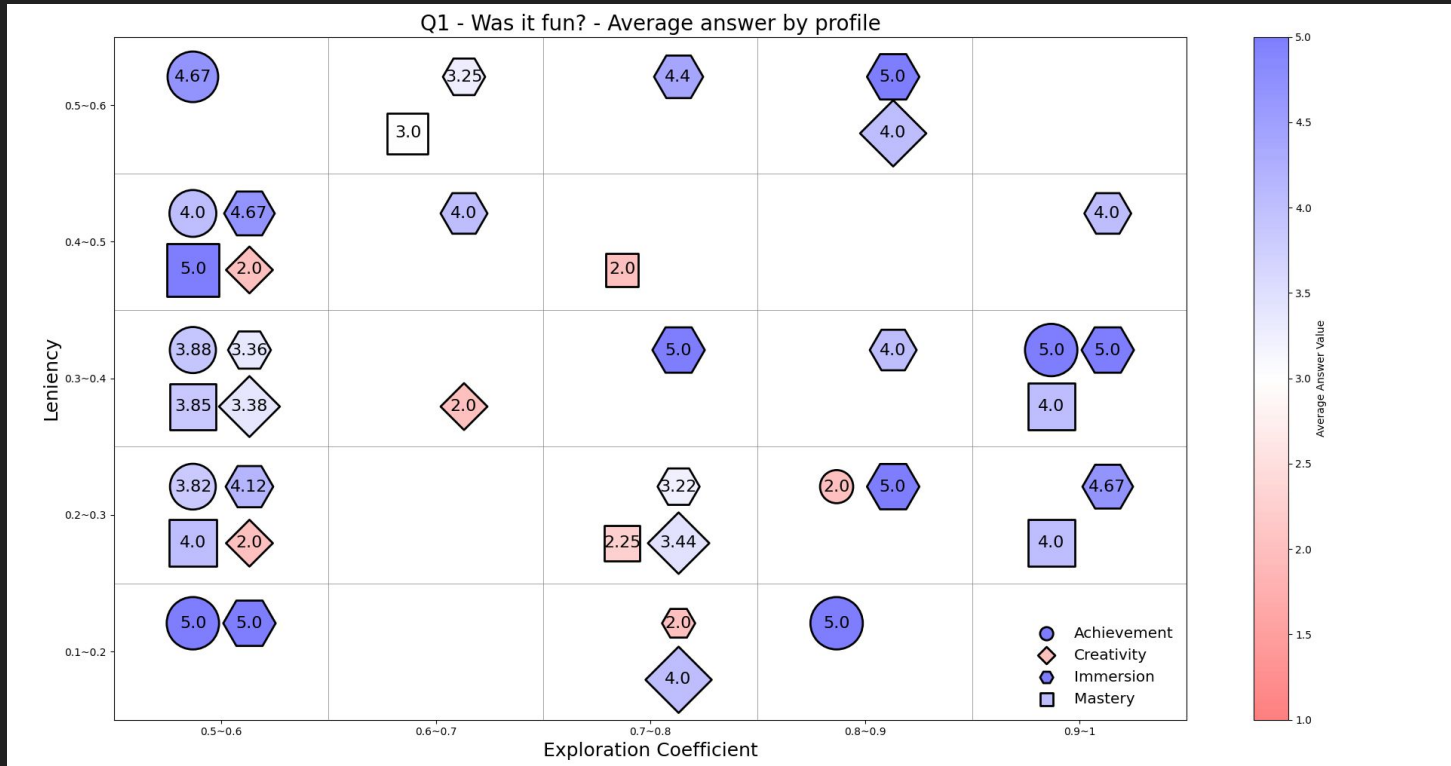
Testes Estatísticos - [Link Colab](#)

```
def statistical_test_on_column(_different_profile, _matching_profile, _column, _column_name, _alternative):
    stat, p = stats.shapiro(_column)
    print('Results for Question:' + _column_name)
    print('Statistics=%.3f, p=%.3f' % (stat, p))
    alpha = 0.05
    if p > alpha:
        print('Sample looks Gaussian (fail to reject H0)')
        stat, p = stats.ttest_ind(_matching_profile[_column_name], _different_profile[_column_name], alternative=_alternative)
    else:
        print('Sample does not look Gaussian (reject H0)')
        stat, p = stats.mannwhitneyu(_matching_profile[_column_name], _different_profile[_column_name], alternative=_alternative)
    print('Statistics=%.3f, p=%.3f' % (stat, p))
    alpha = 0.05
    if p > alpha:
        print('Samples come from same distribution (fail to reject H0)')
    else:
        if _alternative == CurveAlternativeHypothesis.GREATER:
            print('Matching profile is greater than different profile (reject H0)')
        elif _alternative == CurveAlternativeHypothesis.LESS:
            print('Matching profile is lesser than different profile (reject H0)')
        else:
            print('Matching profile is different than different profile (reject H0)')
```

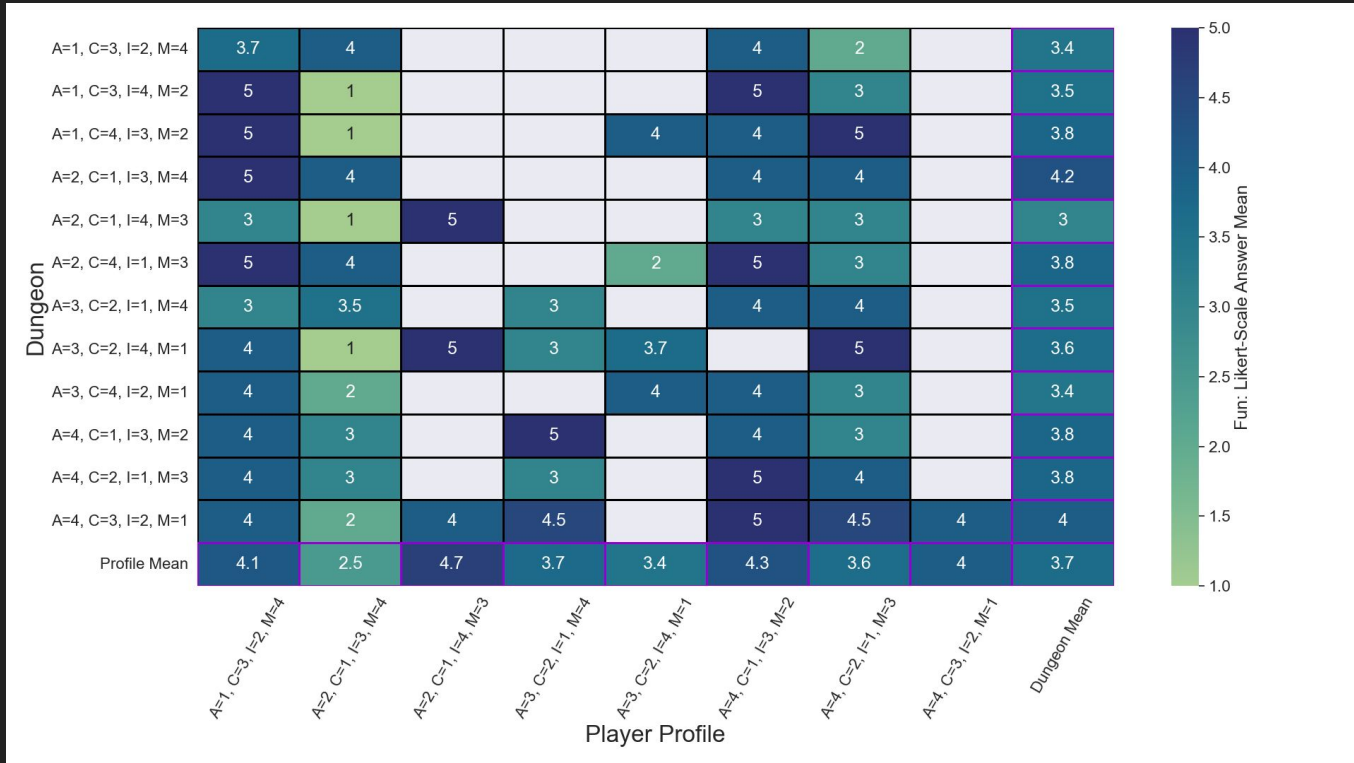
Testes Estatísticos

- Power: 0.902
- Results for Question:Q1-Was it fun?
- Statistics=0.848, $p=0.000$
- Sample does not look Gaussian (reject H_0)
- Statistics=5596.500, $p=0.000$
- Matching profile is greater than different profile (reject H_0)

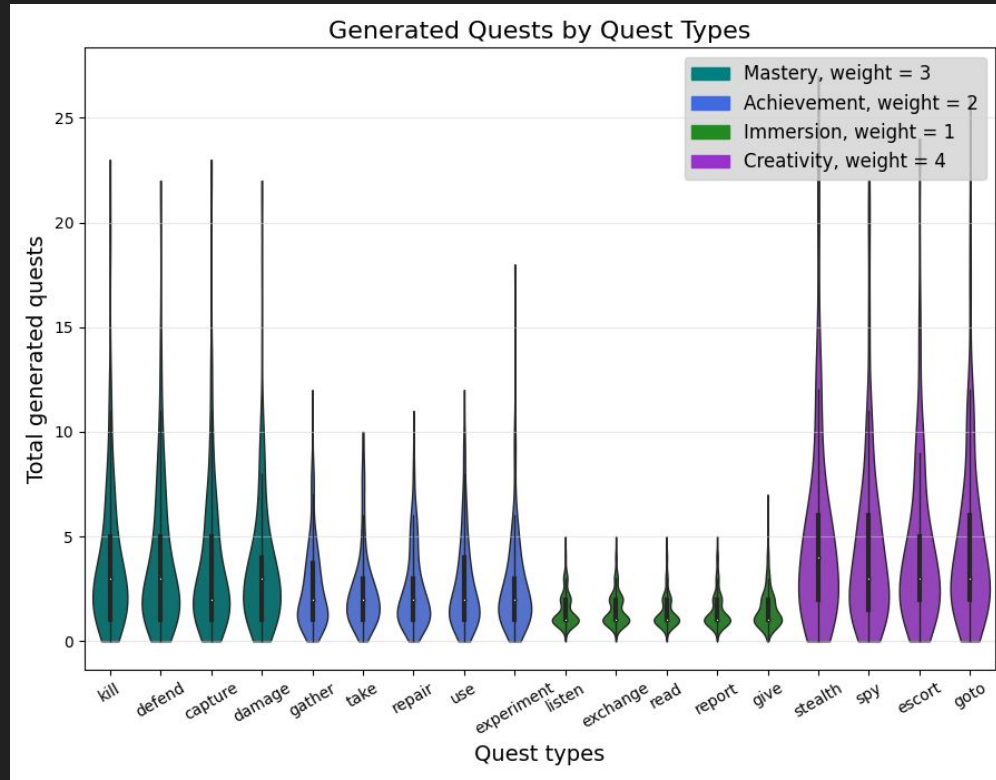
Visualizações - Link Colab



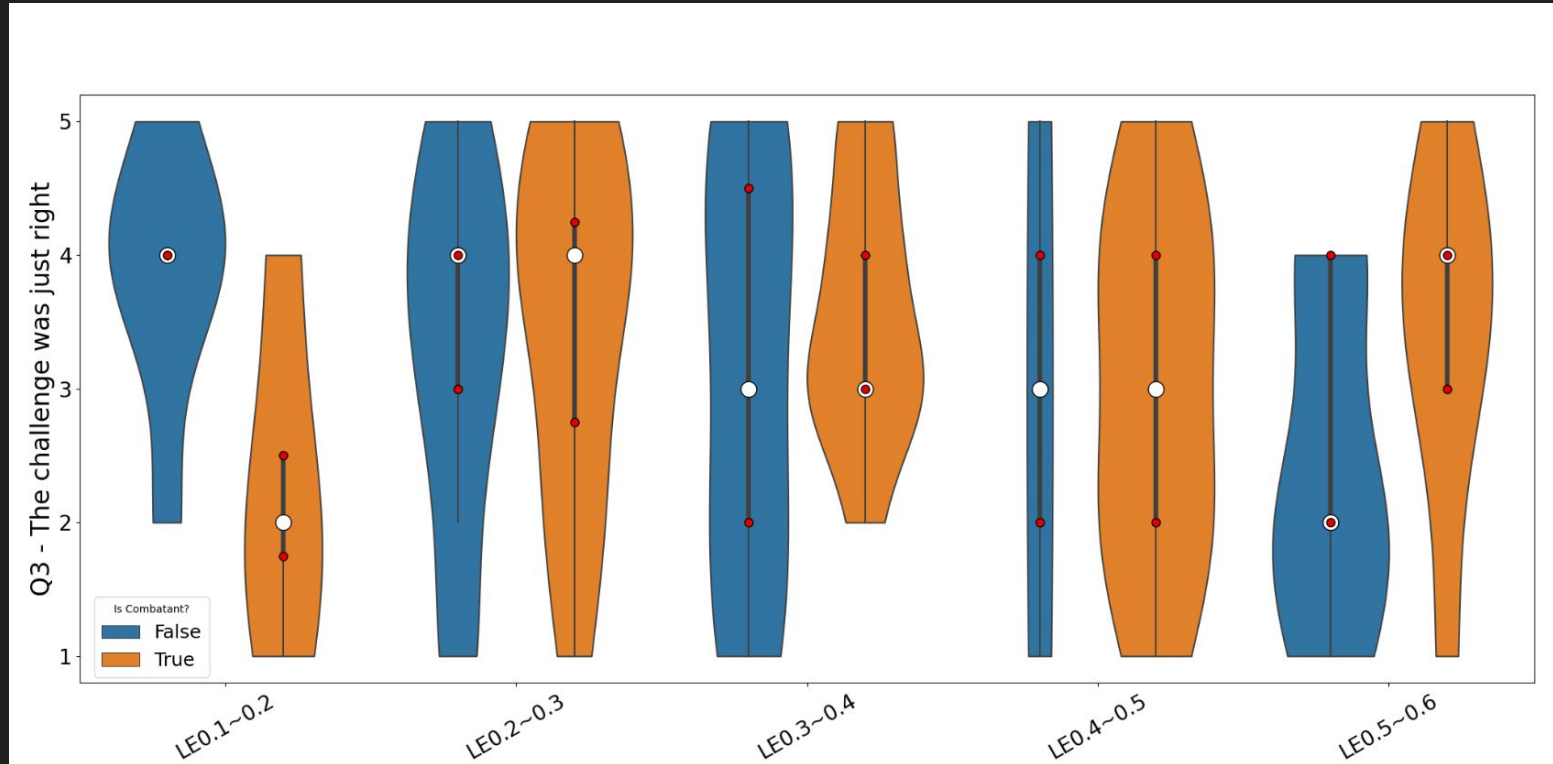
Visualizações



Visualizações



Visualizações





E muito mais!

Referências

Referências

1. https://en.wikipedia.org/wiki/Von_Neumann_architecture
2. <https://edisciplinas.usp.br/pluginfile.php/5979364/course/section/6114666/VonNeumann.pdf>
3. <https://www.curryproject.net/starter-kit-dikw-pyramid/>
4. <https://www.evalueserve.com/blog/making-the-leap-from-insights-to-wisdom-a-collaborative-approach/>
5. <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>