

# Elementos de Sistemas

## Apresentação da disciplina

### Organização de computadores

Renan Trevisoli

Engenharia da Computação

16/08/2023

*A verdadeira viagem de descoberta não consiste em buscar novas paisagens, mas em ter novos olhos.  
Le véritable voyage de découverte ne consiste pas à chercher de nouveaux paysages, mais à avoir de nouveaux yeux.*

*Marcel Proust (1871–1922) escritor francês*

# Apresentação do curso

# Objetivos

- Essenciais
  - Implementar um computador a partir de componentes eletrônicos.
  - Integrar as camadas de programação e execução de um computador.
  - Trabalhar de forma colaborativa no desenvolvimento do sistema.

# Objetivos

- Essenciais

- Implementar um computador a partir de componentes eletrônicos.
- Integrar as camadas de programação e execução de um computador.
- Trabalhar de forma colaborativa no desenvolvimento do sistema.

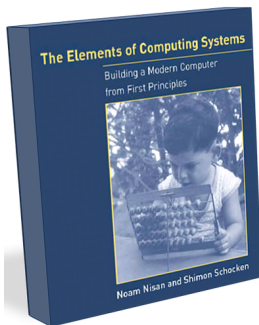
- Complementares

- Compreender a evolução da informática.
- Descrever como dados e instruções são tratados em computadores.
- Entender questões relacionadas a desempenho e operação.

# Ideia inicial

- The Elements of Computing Systems  
Noam Nisan e Shimon Schocken.

Os primeiros capítulos estão disponíveis no [site](#)



# Histórico/Equipe atual

- Curso idealizado e desenvolvido pelo [Prof. Luciano Soares](#) (2016-17).
- Posteriormente ministrado pelo [Prof. Rafael Corsi](#) (2017-21).
- A partir do 2º semestre de 2021, [Prof. Renan Trevisoli](#).
- Equipe atual:



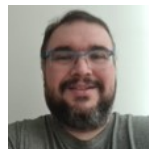
[Renan Trevisoli](#)



Alex Fukunaga  
Prof. Auxiliar



Lícia Sales



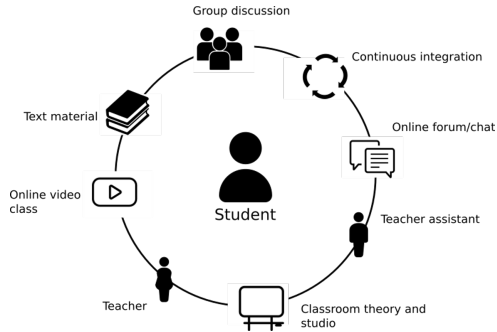
Rogério Cuenca

Insper

# Informações gerais

- Horários das aulas:
  - Quartas: 13:30 às 15:30
  - Sextas: 7:30 às 9:30
  - Atendimento: sextas das 9:30 às 11:00.
- Plano de Ensino
  - Blackboard
- Site da disciplina
  - <https://insper.github.io/Z01.1/>

# Formato do curso



- Estudo prévio
  - Vocês deverão estudar a teoria no [site da disciplina](#) antes das aulas!



# Projetos/APS

- Ao final de todas as APS, vocês terão desenvolvido um computador.
- Serão 9 APS no total.
- Serão desenvolvidas em grupo de 5/6 integrantes (exceto APS A).
- O grupo permanecerá o mesmo ao longo do semestre.
- Desenvolvimento colaborativo/ágil. Cada um terá seu papel (facilitador ou desenvolvedor) em cada APS.
- Cada APS será avaliada pelo trabalho do grupo (rubricas são apresentadas nas páginas de cada projeto) e pela atuação individual (através de Forms).

# Avaliações/Nota final

- 4 avaliações (2 em aula + AI + AF)
- Haverá um projeto opcional valendo 15 pontos no total além de atividades a serem realizadas em aula no total de 10 pontos extras.
- A média final na disciplina, será a média das notas das avaliações e das APS, caso as condições de aprovação sejam satisfeitas.
- Para ser aprovado(a), é necessário:
  - Ao final das avaliações, acumular:
    - 50 pontos em HW
    - 50 pontos em SW
  - Ter no máximo 2 notas individuais em APS menor que C.
  - Ter no máximo 1 nota de grupo em APS menor que C.
  - Média final das APS maior ou igual à C.

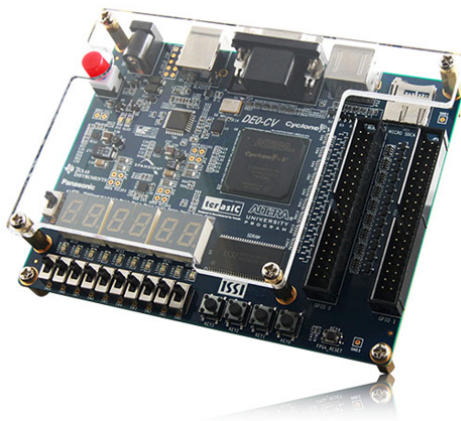
# Cronograma

Elementos de Sistemas			
	Quarta	Sexta	
Introdução	16/08	18/08	Álgebra Booleana
Álgebra Booleana	23/08	25/08	Álgebra Booleana
Feira de Carreiras	30/08	01/09	Álgebra Booleana
AV1	06/09	-	
B - Lógica booleana	13/09	15/09	B - Lógica booleana
C - ULA	20/09	22/09	C - ULA
D - Seq	27/09	29/09	D - Seq
D - Seq	04/10	06/10	AI-AV2
AI-AV2	11/10	-	
E - CPU	18/10	20/10	E - CPU
F - ASM	25/10	27/10	F - ASM
F - ASM	01/11	-	
AV3	08/11	10/11	G - Assembler
	-	17/11	G - Assembler
G - Assembler	22/11	24/11	H - VM\Translator
H - VM\Translator	29/11	01/12	AF - AV4
AF - AV4	06/12	-	

# Ferramental

- Todas as APS (exceto a APS A) serão feitas e entregue pelo Github/Github Classroom.
- A Av1 será feita pelo Blackboard. As demais também serão realizadas pelo Github Classroom.
- Toda a teoria, laboratórios e projetos estarão no [site da disciplina](#).
- Serão usados os softwares GHDL, GTKWave, Quartus, VSCode no SSD que vocês receberão.

# Hardware



# Organização Básica de Computadores

- Os usuários veem as aplicações finais.
- Mas, como uma aplicação é executada fisicamente a nível de hardware?



Tetris



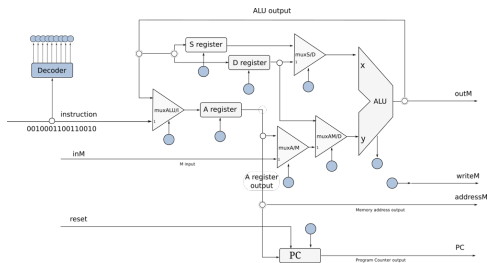
Pong

<sup>1</sup><http://vadim.oversigma.com/Tetris.htm>

<sup>2</sup><https://en.wikipedia.org/wiki/Pong>

## • Hardware

### • Desenvolvimento da CPU

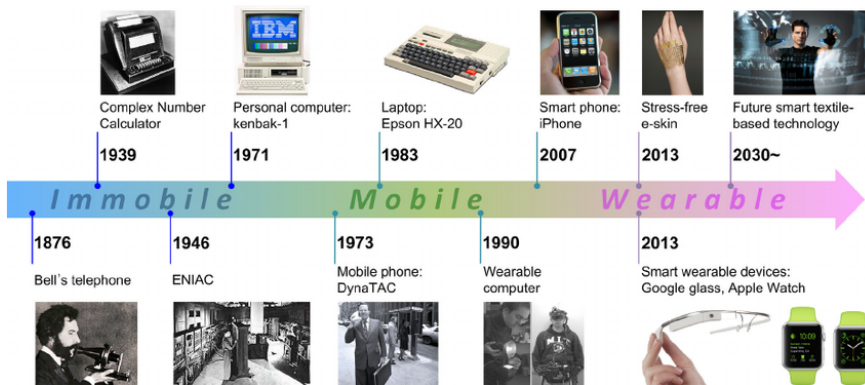


## • Software

### • Ferramentas para programar a CPU



# Evolução dos eletrônicos



<sup>1</sup> [https://www.researchgate.net/figure/A-chronological-illustration-presenting-the-journey-of-electronic-components-from\\_fig1\\_282502585](https://www.researchgate.net/figure/A-chronological-illustration-presenting-the-journey-of-electronic-components-from_fig1_282502585)

# Quiz

Quando foi depositada a primeira patente de um transistor?

- ① 1926
- ② 1933
- ③ 1949
- ④ 1952
- ⑤ 1961

**KLA+**  
Keep Looking Ahead



# Quiz

Em 1958, Jack Kilby, ganhador do prêmio Nobel de 2000, projetou o primeiro circuito integrado. Quantos transistores possuía esse CI?

- 1 1
- 2 8
- 3 12
- 4 32
- 5 2300

# Lei de Moore

## Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World  
in Data

### Transistor count

50,000,000,000

10,000,000,000

5,000,000,000

1,000,000,000

500,000,000

100,000,000

50,000,000

10,000,000

5,000,000

1,000,000

500,000

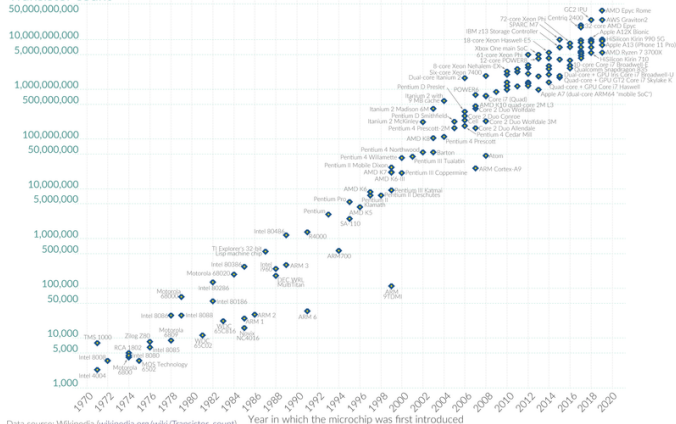
100,000

50,000

10,000

5,000

1,000



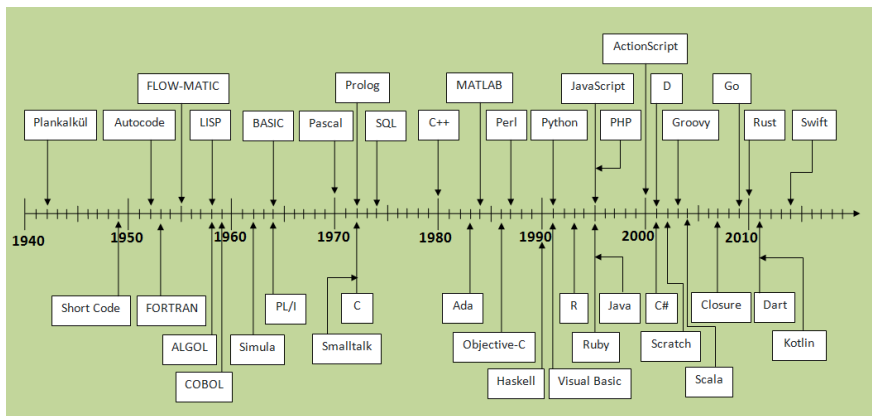
Data source: Wikipedia (wikipedia.org/wiki/Transistor\_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

<sup>1</sup>[https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count)

# Linguagens de programação



<sup>1</sup><https://javaconceptsoftheday.com/history-of-programming-languages/>

# Atividade

## Problema:

Queremos reduzir o tempo de uma viagem de avião.

Proponham, **uma e apenas uma** mudança no processo de uma viagem com potencial para redução de tempo e respondam às perguntas:

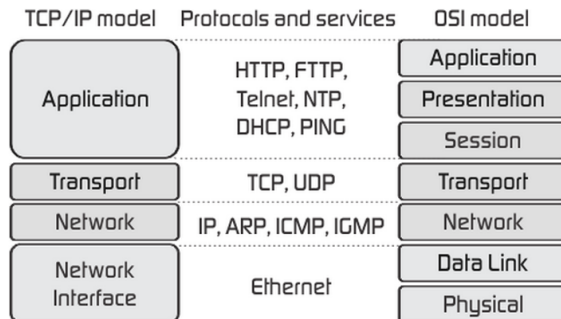
- 1 De quais aspectos dependem a alteração proposta?
- 2 De quais aspectos relacionados à viagem não depende a alteração proposta?

# Camadas de abstração

- O funcionamento de algo é separado em camadas abstratas.
- É uma forma de ocultar detalhes e permitir a separação de interesses.
- Facilita a interoperabilidade e a implementação do sistema.
- Se as interfaces entre as camadas forem bem definidas, podemos mudar uma camada e mesmo assim tudo continuar funcionando.



# Camadas de abstração



# Atividades

## Problema:

Ordene as camadas de abstração de um computador de forma que as mais básicas estejam na base e as mais complexas na parte superior.

Linguagem Assembly

Unidade lógica aritmética

Unidade central de processamento

Lógica sequencial

Linguagem de alto nível

Aplicação

Linguagem de máquina

Lógica combinacional

Transistores

Portas lógicas

Linguagem de máquina virtual

Sistema operacional

Pesquise na Internet as camadas desconhecidas.

## Questões:

Discutam em grupo:

- 1 De forma geral, quais dos níveis de abstração vocês acham que mais mudaram/evoluíram nos últimos 50 anos?
- 2 Quais seriam as mudanças que vocês visualizam para os próximos 10 anos? E para os próximos 50 anos?
- 3 O que mais influenciou e viabilizou na computação, a evolução do Hardware ou do Software?
- 4 O que mais impactou a computação, os grandes servidores ou a computação móvel?

# Próxima aula

- Estudar: Teoria → Dados digitais