

# Programação na Internet

Turma i52d

Apresentação

# Aula presencial 5ª Feira às 14:00

14:00:

- Um turno presente na sala
- Outro turno assiste remotamente via Zoom

15:30:

- Remoto para esclarecimento de dúvidas em breakout

**Aulas práticas e esclarecimento de dúvidas apenas remotos.**

# Infraestrutura

- Moodle SÓ para divulgação de Mensagens
- Github: <https://github.com/isel-leic-pi/PI-2021i-LEIC52D>
  - Código das aulas
  - Sumários
  - Enunciados de TPC
- Slack: [https://join.slack.com/t/isel-leic-pi-2020-1/shared\\_invite/zt-htt6i3o2-~la~7nPbzgHovVdi0dpnZw](https://join.slack.com/t/isel-leic-pi-2020-1/shared_invite/zt-htt6i3o2-~la~7nPbzgHovVdi0dpnZw):
  - Aderir ao canal da turma: e.g. #li52d-mgamboa

# Bibliografia

- ~~1<sup>a</sup> edition Node.js the Right Way:~~
  - ~~DB couch DB~~
- 2nd Edition Node.js 8 the Right Way. Main differences:
  - Async/await
  - DB elasticsearch
- Javascript
  - [Eloquent JavaScript - 3rd edition \(Marius Haverbeke\)](#)

# Programação na Internet

Tecnologias (pela seguinte ordem):

1. **Javascript (75%)**
2. HTTP (20%)
3. HTML e CSS (user interface) (5%)

Aplicação Web = **Servidor** + **Cliente**  
Servidor HTTP      Browser  
(2, pode 1)      (3, 2, 1)

E.g C# se ASP.net (em Pi 2002 - 2015)  
kotlin ou Java se Spring (em Daw)

# JAVASCRIPT

1º desafio

# ASYNC

2º desafio

**IO em Javascript => Async**

**Proibido IO sync => BLOQUEANTE => AVE Javascript é single thread => 1 thread**

**Objectivo do Servidor Web é atender o maximo número de pedidos => THROUGHPUT**

**Se o Servidor é Javascript => Máximo Throughput com 1 thread !!!! => Non-blocking IO**

# Ambiente Virtual de Execução

- Node.js é um AVE Javascript
- Browser é outro AVE Javascript

# Funcional

- Capacidade de passar uma Função como parametron (e.g. callback)
- Função pode ser uma Lambda
- E.g. `foo(... => ...)`
  
- Async idiomas:
  - Callback
  - Promise (em Java `CompletableFuture` em .net `Task`)
  - Async/await