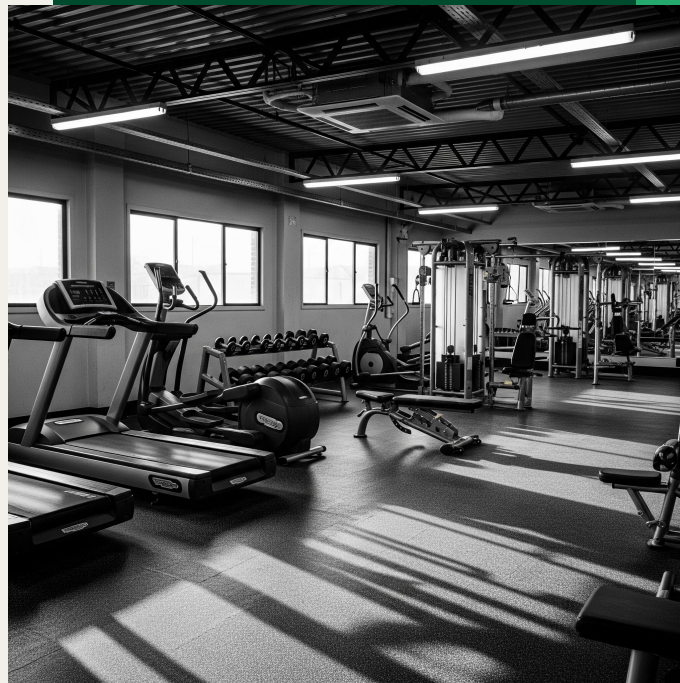


# FitScript

Uma DSL para  
treinos de  
academia





# O problema com as listas de exercício clássicas



## ■ Contexto

Geralmente, treinos de academia são listas estáticas de exercícios (no papel ou em *apps*) que, na maioria das vezes, não levam em consideração seu *status* físico e mental.

A **ideia do FitScript** é fazer com que seja possível **programar seus treinos** como ***scripts***, ao invés de listas. Os exercícios de um treino podem ser incluídos ou modificados a depender do nível de energia do usuário, tempo disponível ou batimento cardíaco.



# Exemplo

(.fit)

```
routine "Leg Day" {  
    let energy = read_sensor(ENERGY_LEVEL);  
  
    exercise "Squat" { sets: 3; reps: 8;  
weight: 110kg; }  
  
    if (energy > 7) {  
        exercise "Leg Press" { sets: 3;  
reps: 12; weight: 225kg; }  
    }  
  
    loop 3 times {  
        exercise "Calf Raises" { sets: 1;  
reps: 15; }  
    }  
}
```

---

# Arquitetura do Projeto



# FitWatch VM

- 01 Program Counter (PC)
- 02 Registradores
- 03 Sensores
- 10 Instruções

FitWatch é uma VM mínima que visa simular o funcionamento de um *smartwatch*. É Turing-completa, com suporte à manipulação de dados, condicionais e recursão.

[https://en.wikipedia.org/wiki/Counter\\_machine](https://en.wikipedia.org/wiki/Counter_machine)

(O FitWatch foi fortemente inspirado na MicrowaveVM)

---

# Características da linguagem + VM

---

## ■ INTEGRAÇÃO COM SENSORES

A linguagem é integrada com sensores (que representam dados do *smartwatch*).

No momento, três sensores são reconhecidos:

- ENERGY\_LEVEL (0 a 10)
- HEART\_RATE (batimentos)
- TIME\_AVAILABLE (minutos)

## ■ APENAS DOIS REGISTRADORES

A VM FitWatch possui apenas dois registradores. Isso é intencional, pois trata-se de uma máquina mínima que simula um *smartwatch*.

No entanto, essa limitação faz com que a geração de Assembly seja um pouco mais complicada (em especial, para *loops* mais complexos).

## ■ SEM SUPORTE PARA FUNÇÕES

A linguagem, nessa primeira versão, não possui suporte para a definição de funções. Tudo precisa ser escrito *inline*.

Essa *feature* poderia ser útil para cálculo repetido de exercícios, peso ou repetições.

# Melhorias

- 01 *Macros*/funções para rotinas reutilizáveis
- 02 Suporte à *heap*
- 03 Incluir *opcodes* de ADD, SUB e comparações (*helpers*)
- 04 Adicionar mais eventos de I/O (sensores, timers, etc.) que possam influenciar as rotinas

# Obrigado!

