

# *Estruturas de Linguagem*

**Francisco Sant'Anna  
Sala 6020-B**

**francisco@ime.uerj.br**

**<http://github.com/fsantanna-uerj/EDL>**

# Online

- GitHub
  - <https://github.com/fsantanna-uerj/EDL/>
- Lista de discussão
  - <https://groups.google.com/d/forum/edl-uerj/>

# Forma de Avaliação

- Tarefas e Apresentações
  - são considerados “continuamente”
- Prova(s)
- Presença  $\geq 75\%$

# Tarefas

- Tarefa-01: **Dom, 03/09**: Lista & GitHub
- Tarefa-02: **Dom, 24/09**: Artigo
- Tarefa-03: **Dom, 01/10**: Instalação Löve
- Tarefa-04: **Dom, 08/10**: Jogoinho
- Tarefa-05: **Dom, ??/??**: Bindings
- Tarefa-06: **Dom, ??/??**: Tipos de Dados
- Tarefa-07: **Dom, ??/??**: Arrays
- Tarefa-08: **Dom, ??/??**: Closures & Co-rotinas
- Tarefa-09: **Dom, ??/??**: Interpretador

# Provas

- Prova Única: Qua, ??/??
- Prova Final: Qua, ??/??

# Bibliografia

- Concepts of Programming Languages
  - 11<sup>th</sup> edição
  - Robert W. Sebesta
- `https://www.pearsonhighered.com/program/Sebesta-Concepts-of-Programming-Languages-11th-Edition/PGM270801.html`

# Calendário

- 28/08, 29/08

- Introdução

- 17/04, 19/04

- Sintaxe, Semântica, Avaliação, Lua

- ??/??

- Lua, Löve

- ??/??

- Lab

- ??/??

- Bindings

- ??/??

- Ciclo de Vida

- ??/??

- Escopo

- ??/??, ??/??, ??/??

- Dados

- ??/??

- ADTs

- ??/??, ??/??

- Código

- Closures & Coro





# Tarefa-01: Lista & Github

(até domingo 03/09)

- Cadastrar-se na lista da turma
  - <https://groups.google.com/d/forum/edl-uerj/>
- Dar um *Fork* no repositório da disciplina
  - <https://github.com/fsantanna-uerj/EDL/>
  - Habilitar a aba the *Issues*
    - <http://softwareengineering.stackexchange.com/questions/179468/forking-a-repo-on-github-but-allowing-new-issues-on-the-fork/179470#179470>
- Adicionar um arquivo tarefa-01/RESPOSTA.md
  - texto “pessoal” qualquer formatado em *Markdown*
    - <https://help.github.com/articles/basic-writing-and-formatting-syntax/>
- Mandar um e-mail para a lista com o link do seu projeto

# Tarefa 02 - Artigo

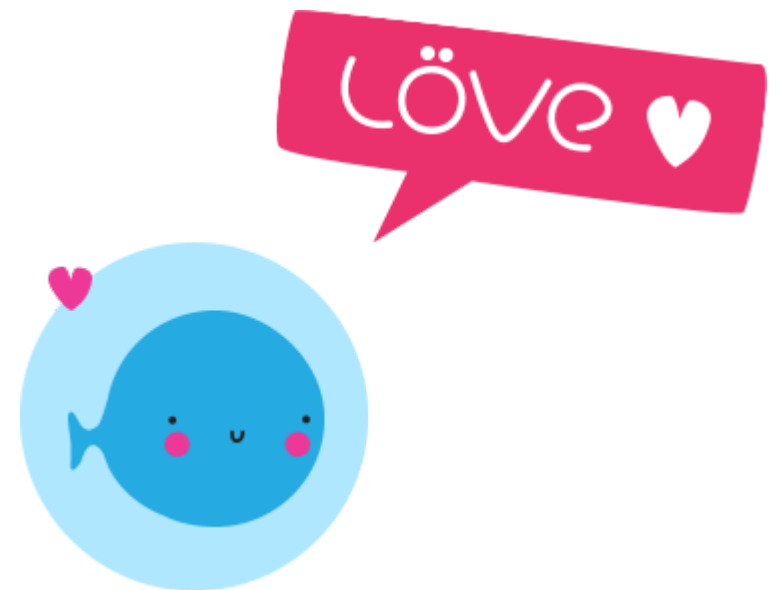
## (até domingo 24/09)

- Escolher uma linguagem com a qual você **não está familiarizado**.
  - evitar duplicatas com outros colegas
  - **instalar e escrever pequenos programas com a linguagem**
  - usar pelo menos uma funcionalidade de **alta expressividade**
    - discutir com o professor sobre qual funcionalidade
- Escrever um pequeno artigo (estilo *Wikipedia*):
  - conteúdo em `tarefa-02/ARTIGO.md`
  - [0.5] origens e influências (linha do tempo)
  - [0.5] classificação (imp/func/log/oo, est/din, usos)
  - [5.0] avaliação em comparação com linguagens que você conhece (read/write, expressividade)
  - [4.0] exemplos de código representativos (vs outra linguagem)
- Slides de apresentação (5-10 slides)
  - `tarefa-02/slides.pdf`

# Tarefa 03 - Lua/Löve

(até domingo 01/10)

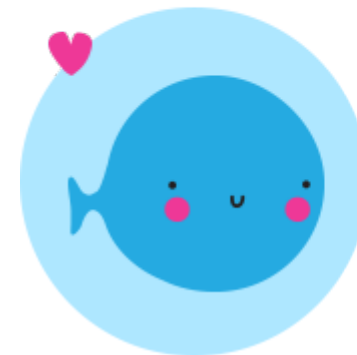
- <https://love2d.org/>
- Mostrar instalação/execução do ambiente:
  - Código no GitHub
  - Screenshot no GitHub



# Tarefa 04 - JOGUINHO

(até domingo 08/10)

- Individual
- Fazer um jogo qualquer
  - Teclado e/ou Mouse
  - Imagens e/ou Retângulos
  - Animações (i.e., tempo como input)
  - Interação entre objetos (e.g., colisão)
- Arquivo único (e sem bibliotecas externas ao Löve)
- Jogos proibidos
  - “frogger”, “arkanoid”
  - “space invaders”



# Tarefa 05 – Bindings

(até domingo ??/??)

- Para cada “binding time” de Lua (*design, compile, run*), identificar no joguinho dois exemplos (com características diferentes).
- Adicionar comentários no próprio código fonte do jogo com as identificações e explicações.
- Exemplo:

```
function inc (v)
    return v + 1
end
-- Nome: variável “v”
-- Propriedade: endereço
-- Binding time: execução
-- Explicação: dado que “v” é uma variável
-- local de uma função, seu endereço só pode
-- ser determinado em tempo de execução.
```

# Tarefa 06 - Tipos de Dados

(até domingo ??/??)

- Individual
- Identificar no jogo valores de tipos de dados não primitivos (pelo menos 4 diferentes)
  - enumeração, registro, tupla, array, dicionário, união
- Caso não existam 4 diferentes, altere o jogo para que tenham
- Usar comentários para facilitar a identificação
  - - - tarefa-06

# Tarefa 07 - Arrays

(até domingo ??/??)

- Individual
- Adicionar ao jogo uma coleção dinâmica de objetos
  - criar novos objetos periodicamente
    - timer ou evento (e.g., tecla pressionada)
  - remover objetos
    - timer ou evento (e.g., colisão)
  - objetos devem interagir entre si
    - e.g., colisão
- Descrever o ciclo de vida da coleção e de um objeto
  - escopo, tempo de vida, alocação/desalocação

# Tarefa 08 - Closures e Co-rotinas

(até domingo ??/??)

- Individual
- Usar closures no lugar de objetos
  - pelo menos uma classe (e.g., jogador, bala, bloco, etc)
- Usar co-rotinas para movimentar um objeto de forma retangular
- Usar comentários para facilitar a identificação
  - - - tarefa - 08
- Basear-se em EDL/code/sub/game-0 [1 - 2] /



# Tarefa 09 - Interpretador

(até domingo ??/??)

- Criar um interpretador para uma linguagem simples
  - expressões
    - constantes, variáveis, aritméticas
  - comandos
    - atribuição, sequência, condicional, while
  - o resultado do programa é o conteúdo da variável `ret`
  - variáveis indefinidas avaliam para 0

```
(Seq
  (Attr "x" (Add (Num 11) (Num 9)))
  (If (Var "x")
    (Attr "ret" (Var "x"))
    (Attr "ret" (Num 100))))
```

20