Estruturas de Linguagem

Francisco Sant'Anna

francisco@ime.uerj.br

http://github.com/fsantanna/EDL

Bibliografia

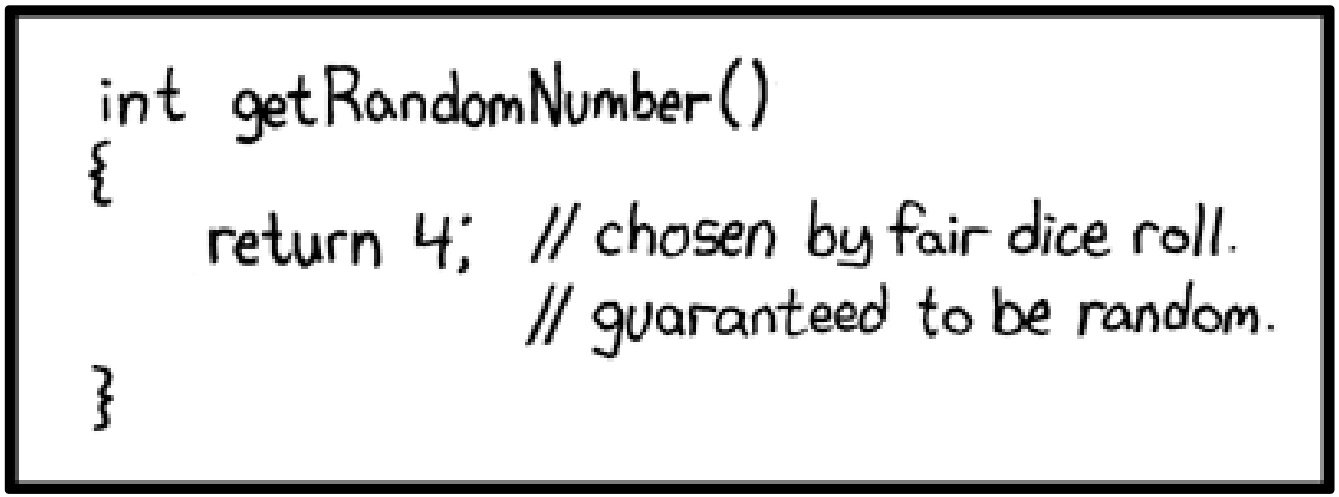
- Concepts of Programming Languages
  - 11th edição
  - Robert W. Sebesta

- https://www.pearsonhighered.com/program/Sebesta-Concepts-of-Programming-Languages-11th-Edition/PGM270801.html

Critério de Avaliação

- Trabalhos e Apresentações
  - são considerados “continuamente”

- Prova(s)



Provas

- Prova Única: **Sexta 09/12**
- Prova Final: ?

Trabalhos

- Trabalho-00: **Qua, 09/03**: GitHub
- Trabalho-01: **Dom, 16/03**: Artigo
- Trabalho-02: **Dom, 04/09**: Lua/Löve, instalação
- Trabalho-03: **Dom, 18/09**: Joguinho (grupo 2-3)
- Trabalho-04: **Dom, 25/09**: Bindings
- Trabalho-05: **Dom, 09/10**: Arrays (grupo)
- Trabalho-06: **Dom, 23/10**: Tipos de Dados (grupo)
- Trabalho-07: **Dom, 13/11**: Closures & Co-rotinas (grupo)
- Trabalho-08: **Dom, 12/12**: Interpretador Simples

Trabalho 00 - GitHub
(até quarta-feira 09/03)

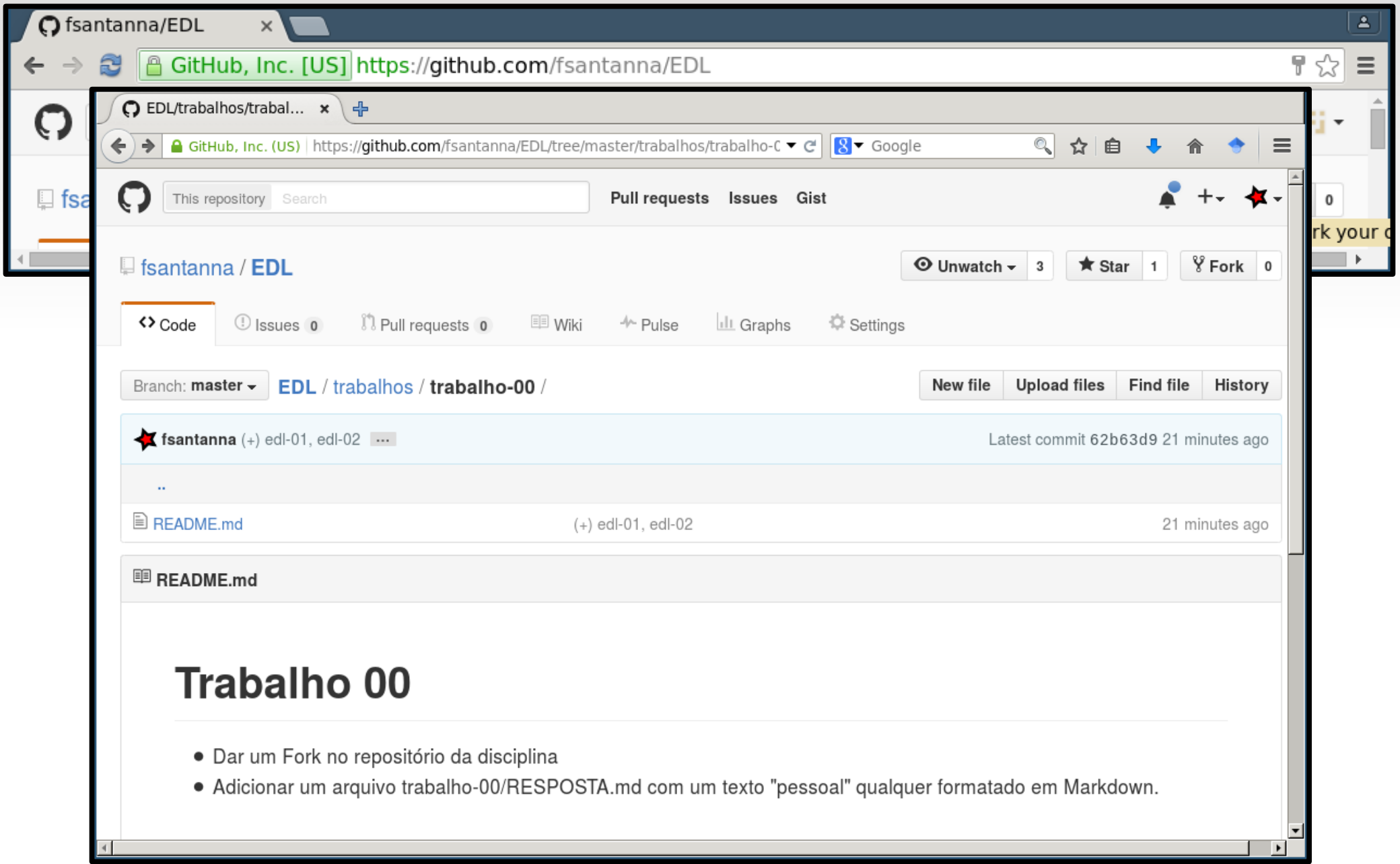
- Trabalho
  - Dar um *Fork* no repositório da disciplina
  - Adicionar um arquivo `trabalho-00/RESPOSTA.md` com um texto "pessoal" qualquer formatado em *Markdown*
- Links
  - `http://github.com/fsantanna/EDL`
  - `https://help.github.com/articles/good-resources-for-learning-git-and-github/`
  - `https://help.github.com/articles/basic-writing-and-formatting-syntax/`

Trabalhos em Grupo

- Nota única para o grupo
- Grupo divide os pontos "a gosto", **mas justificando**, ex.:
  - Fulano fez "a,b,c" => **X** pontos
  - Sicrano fez "d,e" => **Y** pontos
  - Beltrano não fez nada => **Z** pontos

    (estamos "emprestando" pontos pra ele)
  - **X** + **Y** + **Z** = nota única
- Justificativas devem ser compatíveis com *commits* do projeto

Trabalho 00 - GitHub



Trabalho 01 - Artigo

(até quarta-feira 16/03)

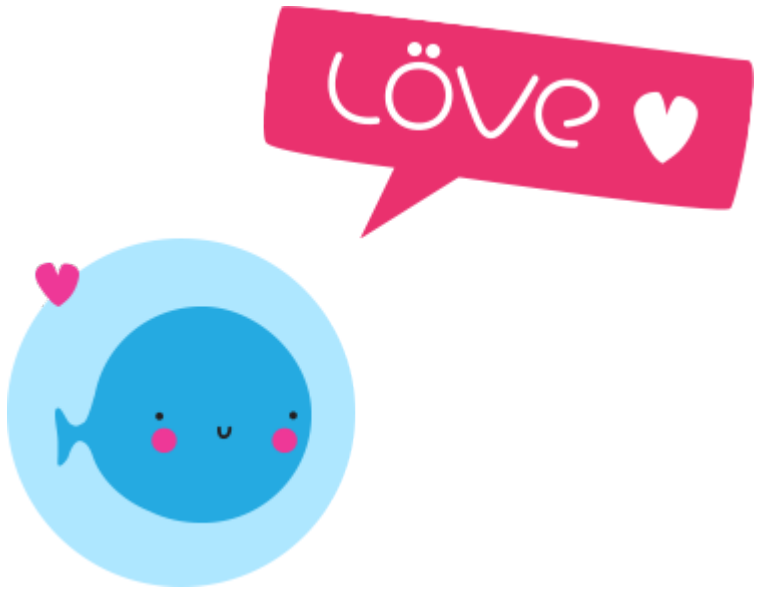
- Escolher uma linguagem com a qual você **não está familiarizado**.
  - evitar duplicatas com outros colegas
- Escrever um pequeno artigo (estilo *Wikipedia*):
  - origens e influências (linha do tempo)
  - classificação (imp/func/log/oo, est/din, usos)
  - avaliação em comparação com linguagens que você conhece (read/write, expressividade)
  - exemplos de código representativos
  - `trabalho-01/ARTIGO.md`
- Slides de apresentação (4-6 slides)
  - `trabalho-01/slides.pdf`

Trabalho 02 – Lua/Löve

(até domingo 04/09)

- https://love2d.org/

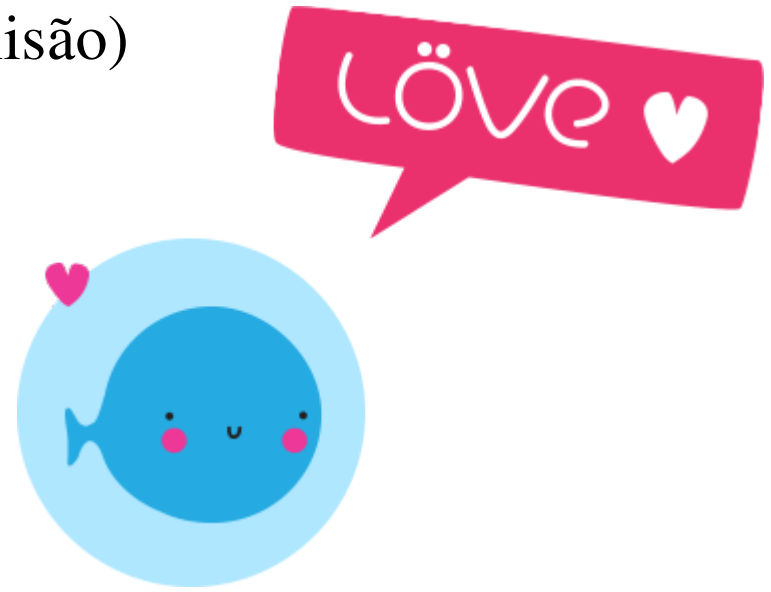
- Mostrar instalação/execução do ambiente:
  - Código no GitHub
  - Screenshot no GitHub



Trabalho 03 – Joguinho

(até domingo 18/09)

- Grupo de 2 ou 3 pessoas
- Fazer um jogo qualquer
  - Teclado e/ou Mouse
  - Imagens e/ou Retângulos
  - Animações (i.e., tempo como input)
  - Interação entre objetos (e.g., colisão)



Trabalho 04 – Bindings

(até domingo 25/09)

- 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.
```

Trabalho 05 – Arrays

(até domingo 09/10)

- Mesmo grupo do joguinho
- 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

Trabalho 06 – Tipos de Dados

(até domingo 23/10)

- Mesmo grupo do joguinho
- 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
  - `-- trabalho-06`

Trabalho 07 – Closures e Co-rotinas

(até domingo 13/11)

- Mesmo grupo do joguinho
- 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
  - `-- trabalho-07`

Trabalho 08
(até domingo 12/12)

- Criar um iterpretador 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

A Linguagem Elm

- https://www.youtube.com/watch?v=F-nTU3Wy26I
- ...