

## 1. Objetivo e etapas

O objetivo do projeto é desenvolver um compilador para uma linguagem de programação simplificada baseada em uma linguagem já existente e pode ser desenvolvido individualmente ou em equipes de no máximo três membros. O projeto deve ser desenvolvido em três etapas valendo os seguintes pontos:

1) especificação - 3,0 pontos;

2) *front-end* - 4,0 pontos;

3) *back-end* - 3,0 pontos.

\* Membro de equipe ou equipe que não apresentar no dia agendado será penalizado com metade da nota obtida na apresentação.

## 2. Datas

10 a 18 de novembro – apresentação da especificação agendada com os monitores.

18 de novembro – submissão da especificação no SIGAA até 23:59.

25 e 27 de novembro – apresentação do *front-end*.

27 de novembro – submissão do *front-end* no SIGAA até 23:59.

2 e 4 de dezembro – apresentação do *back-end*.

4 de dezembro - submissão do *back-end* no SIGAA até 23:59

## 3. Desenvolvimento

### 3.1. Especificação

Primeiramente, escolher a linguagem de programação já existente ao qual a linguagem simplificada a ser desenvolvida é baseada. *Apenas uma* dentre as seguintes linguagens existentes deve ser escolhida: *a) javascript, b) solidity ou c) lua.*

Apresentar ao professor ou monitores a linguagem escolhida e os itens a seguir com exemplos, valendo 0,5 ponto cada:

- I. Tipos básicos e derivados
- II. Operadores e a precedência dos operadores
- III. Regras para uso de variáveis, constantes e seu escopo
- IV. Declaração de funções
- V. Funções nativas
- VI. Sentenças de atribuição e controle

*Critério de avaliação:* quanto mais próximo da linguagem base maior a pontuação.

Após a apresentação, problemas indicados pelo professor ou monitor devem ser corrigidos, e a especificação deve ser descrita em documento de texto ou slides e submetido no SIGAA com todos os itens requisitados.

### 3.2. *Front-end*

Apresentar ao professor a implementação dos seguintes itens, valendo 1,0 ponto cada:

- a) Regras léxica e sintática.
- b) Regras semântica.
- c) Execução do compilador em linha de comando da seguinte forma:
  - I. ler da entrada padrão arquivo com programa escrito na linguagem;
  - II. gerar na saída padrão uma confirmação ou erro lexo/sintático/semântico indicando a linha onde ocorreu.
- d) Manual (arquivo *readme*) contendo informação do *front-end* sobre
  - I. Instalação de bibliotecas pré-requisitos para execução
  - II. execução com exemplos de códigos com erro e sucesso
  - III. atualizações da especificação caso necessário

Após a apresentação, problemas indicados pelo professor ou monitor devem ser corrigidos, e o *front-end* deve ser submetido no SIGAS com todos os itens requisitados comprimidos no formato ZIP.

### 3.3. *Back-end*

Apresentar ao professor a implementação dos seguintes itens, valendo 1,0 ponto cada:

- a) Regras para tradução da parse tree ou árvore sintática no passo anterior para uma linguagem intermediária, podendo ser LLVM ou JASMIN.
- b) Geração de executável em seu sistema operacional a partir da linguagem intermediária.
- c) Manual (arquivo *readme*) contendo informação do *back-end* e o *front-end* da etapa anterior (compilador completo) sobre
  - I. Instalação de bibliotecas pré-requisitos para execução
  - II. execução com exemplos de códigos com erro e sucesso

Após a apresentação, os problemas indicados pelo professor ou monitor devem ser corrigidos, e o *back-end* mais o *front-end* anteriormente desenvolvido (compilador completo), deve ser submetido no SIGAA com todos os itens requisitados comprimidos no formato ZIP.