

Trabalho Final – Criação de um Analisador Léxico e Sintático - Ferramentas de Análise -

- O trabalho deverá ser executado com as equipes de trabalho.
- Será executado em quatro etapas, sendo que cada aula, uma etapa será entregue, no final todo o trabalho deverá ser apresentado em sua íntegra (AS).
- Utilize as ferramentas escolhidas para a geração dos analisadores, a grande maioria possui uma ferramenta para o analisador léxico e outra para o sintático, observe a documentação.
- Cada ferramenta possui uma sintaxe e uma forma de executar os analisadores léxicos e sintáticos, portanto é importante ler e estudar a documentação, para que possam entender o que está sendo implementado.
- As entradas e saídas deverão seguir o padrão da ferramenta, ou seja, por arquivo ou com console.
- A Gramática de cada etapa, deverá ser entregue em um arquivo texto editável, junto com as especificações (quantas casas o float/real assume, quantos caracteres o char e outras informações necessárias para o entendimento).
- Compacte todos os arquivos gerados em uma única pasta para enviar, pois cada ferramenta é diferenciada.
- As etapas devem ser enviadas em separado, o conjunto completo será na aula depois da AS. Precisando de alguma correção extra, deve ser efetuada por e-mail: cleiton.marques@ulbra.br.
- As datas de entregas e pesos, estão organizados/descritos nas etapas abaixo.
- A não entrega de uma etapa não invalida o trabalho, porém a mesma deverá ser apresentada na AS (Avaliação Semestral).

Etapas 1

Apresente a Gramática e os analisadores léxicos e sintáticos para o reconhecimento de declaração de variáveis no estilo da Linguagem C (leia com atenção as especificações abaixo, não deve ser implementado nem mais, nem menos).

- Devem reconhecer os tipos: `char` - `int` - `float`
- O char pode ser um caractere ou uma cadeia (se a ferramenta possui restrição de quantidade, deve ser informada).
- As variáveis podem ser declaradas com ou sem inicialização.
- Identificadores: de acordo com as regras da Linguagem C (iniciam por letras ou `'_'`, depois do segundo caractere pode ser número, letra ou `'_'` e o único caractere especial reconhecido é o `'_'`)
- Podem ser declarados individualmente ou por uma lista (sendo lista separados por vírgulas).
- A finalização de cada declaração será por ponto-e-vírgula (`;`)

Data de entrega: 27/05/2025 (terça-feira), pelo Ambiente Aula, link: Trabalho Final - Compiladores (Etapa 1). Peso: 0.5

Etapa 2

Apresente a Gramática e os analisadores léxicos e sintáticos para o reconhecimento dos comandos de seleção: `if` e `switch-case`, no estilo da Linguagem C.

Comando `if`:

- Sintaxe: `if(condição){<comandos>} else {<comandos>}`
- As condições podem ser formadas por qualquer um dos operadores relacionais descritos na tabela abaixo.
- Os comandos serão apenas matemáticos, ou seja, operações matemáticas simples: soma, subtração, multiplicação e divisão (operadores matemáticos), que serão atribuídos, por meio dos operadores de atribuição a uma variável.
- A finalização de cada comando será por ponto-e-vírgula (;).
- O comando `else` é opcional (assim, como na linguagem C).

Comando `switch-case`:

- Sintaxe:

```
switch(variável) {  
  
    case id:{<comandos> break;}  
  
    case id: case id: {<comandos> break;}  
  
    default: {<comandos>}  
  
}
```

- Mesmas informações do comando `if` na questão dos comandos.
- A finalização de cada comando será por ponto-e-vírgula (;).
- Os cases podem ser unitários ou até uma quantidade específica ou "livres" (`case id: case id:`).
- O comando `default` é opcional (assim, como na linguagem C).
- * As variáveis devem seguir a Gramática apresentada na declaração de variáveis (identificadores).
- * A Etapa 1 deve estar presente no mesmo arquivo, ou seja, junto com a Gramática e especificações.

Data de entrega: 03/06/2025 (terça-feira), pelo Ambiente Aula.link: Trabalho Final - Compiladores (Etapa 2). Peso: 1

- Operadores Matemáticos

Operador	Exemplo	Comentário
+	$x + y$	Soma x e y
-	$x - y$	Subtrai y de x
*	$x * y$	Multiplica x e y
/	x / y	Divide x por y
++	x++	$x = x + 1$ (opcional)
--	x--	$x = x - 1$ (opcional)

- Operadores de Atribuição

Operador	Exemplo	Comentário
=	$x = y$	Atribui o valor de y a x
+=	$x += y$	Equivale a $x = x + y$
-=	$x -= y$	Equivale a $x = x - y$
*=	$x *= y$	Equivale a $x = x * y$
/=	$x /= y$	Equivale a $x = x / y$

- Operadores Relacionais

Operador	Exemplo	Comentário
==	$x == y$	O conteúdo de x é igual ao de y
!=	$x != y$	O conteúdo de x é diferente de y
<=	$x <= y$	O conteúdo de x é menor ou igual ao de y
>=	$x >= y$	O conteúdo de x é maior ou igual ao de y
<	$x < y$	O conteúdo de x é menor que o de y
>	$x > y$	O conteúdo de x é maior que o de y