

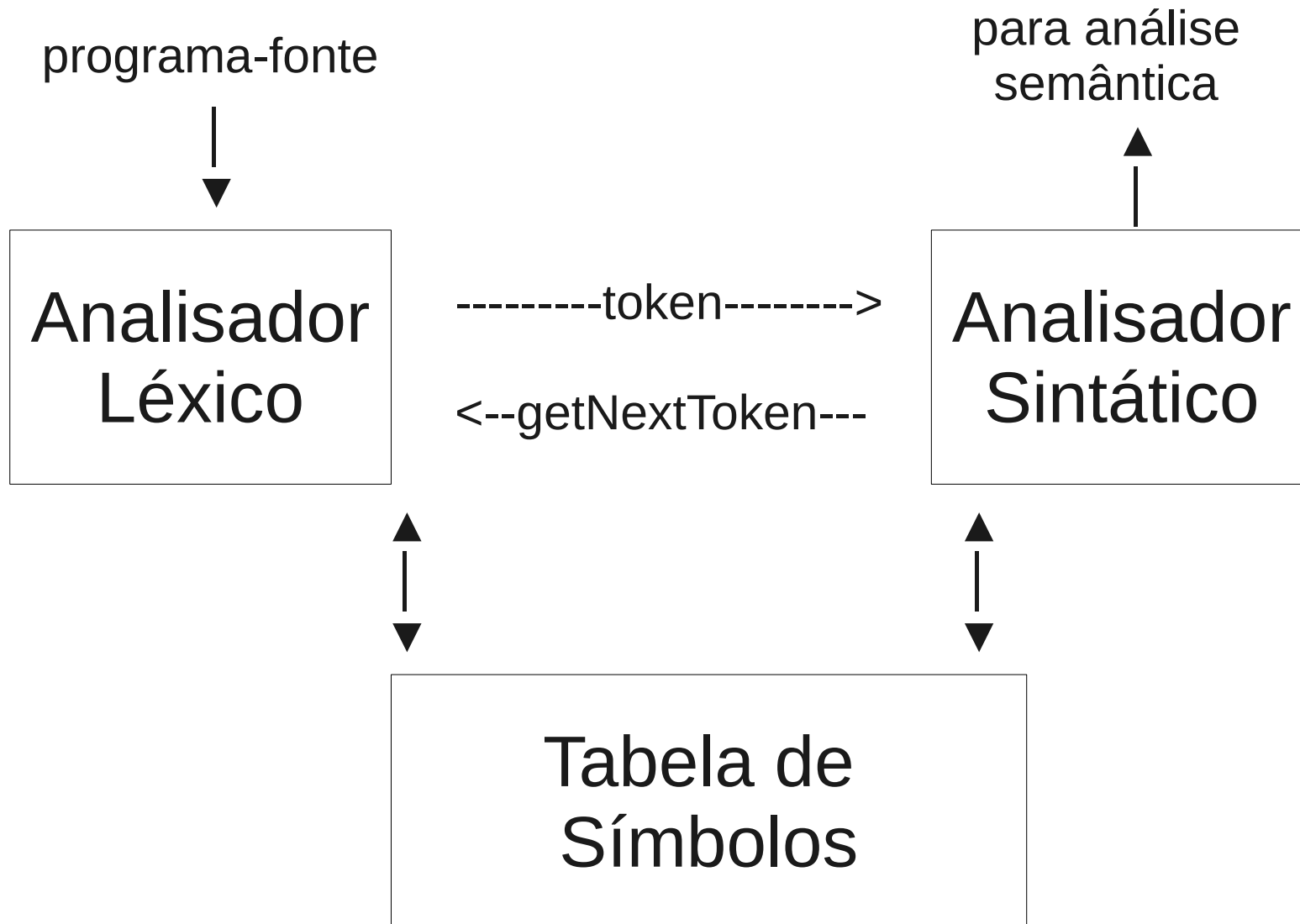
# Analizador Léxico

- O papel do Analizador Léxico
  - Análise Léxica x Análise Sintática
  - Tokens, Padrões e Lexemas
  - Atributos de tokens
  - Erros léxicos
- Especificação de tokens
  - Expressões regulares
  - Definições regulares
- Reconhecimento de tokens
  - Diagramas de transição
  - Reconhecimento de palavras reservadas e identificadores.

# O papel do analisador léxico.

- Ler os caracteres do programa-fonte, agrupá-los em lexemas e produzir como saída uma sequência de tokens para cada lexema no programa-fonte.
- Enviar o fluxo de tokens para o analisador sintático.
- Registrar na tabela de símbolos informações sobre os identificadores encontrados.
  - Eliminar comentários e espaços em branco
  - Correlacionar mensagens de erros geradas pelo compilador com o programa-fonte.

# O papel do analisador léxico.



# Análise Léxica x Análise Sintática

Existem vários motivos para separar análise léxica da análise sintática:

- Simplicidade do projeto é o mais importante
- Melhor eficiência do compilador
- Melhor portabilidade do compilador

# Tokens, Padrões e Lexemas

- Um TOKEN é um par consistindo de um nome e um valor de atributo(opcional). O nome do token é um símbolo abstrato que representa o tipo de unidade léxica.(Esperado como entrada pelo Analisador Sintático)
- O PADRÃO é uma descrição da forma que os lexemas de um token podem assumir. Pode ser definido por uma expressão regular.
- Um LEXEMA é uma sequência de caracteres do programa-fonte que casa com o padrão para um token e é identificado pelo analisador léxico como uma instância desse token.

# Tokens, Padrões e Lexemas

<u>TOKEN</u>	<u>Descrição Informal</u>	<u>LEXEMAS</u>
<b>if</b>	caracteres i, f	if
<b>else</b>	caracteres e, l, s, e	else
<b>comparação</b>	< ou > ou <= ou >= ou == ou !=	<=, !=
<b>id</b>	letra seguida por letra e dígito	pi, x1
<b>número</b>	qualquer constante numérica	3,14159 , -2
<b>literal</b>	sequência de caracteres entre “	“ hoje “

# Atributos de token

Quando mais de um lexema casar com um padrão, o analisador precisa oferecer às demais fases do compilador informações adicionais sobre qual foi o lexema casado.

Por exemplo quando dois tokens **número** forem encontrados, o atributo pode ser o valor do número.

Porém quando um token precisar de mais que um atributo, é melhor armazená-los na tabela de símbolos e passar como valor do atributo um ponteiro para a tabela de símbolos.

# Erros Léxicos

É difícil para o analisador léxico saber, sem o auxílio de outros componentes, que existe um erro no código-fonte.

- Por exemplo : `fi` é um identificador ou um `if` incorreto ???

O erro que pode ser identificado pelo analisador léxico é um caracter inválido no programa-fonte.

O Tratamento para esse tipo de erro é o que chamamos de Modo de Pânico onde removermos os caracteres seguintes até encontrar um token bem formado.



# Especificação de Tokens

## Expressões Regulares.

<u>TOKEN</u>	<u>Padrão</u>	<u>LEXEMAS</u>
<b>if</b>	if	if
<b>else</b>	else	else
<b>Comparação</b>	(<   >   <=   >=   ==   != )	<=, !=
<b>id</b>	(a b c .... z).(a b c .... z 0 1 2.... 9)*	pi, x1
<b>Número</b>	(- ).(0 1 2 ... 9)+[(,(0 1 2 ... 9)+  ε]	3,14159 , -2
<b>literal</b>	“(a b c .... z 0 1 .... 9   _)”	“ hoje “

# Especificações de tokens

## Definições regulares

letra  $\rightarrow$  a|b|c|.....|z|A|B|C|...|Z

digito  $\rightarrow$  0 | 1 | 2 | ... | 9

id  $\rightarrow$  letra(letra|digito)\*

# Reconhecimento de Tokens

Diagrama de Transição

Reconhecimento de Palavras Reservadas e Ids

No quadro negro !!