

## Aula 11: Tópico 07 - Análise Semântica - parte 2

Apresenta a tabela de símbolos que:

Captura a sensibilidade ao contexto.

Permite ao compilador lembrar detalhes associados ao nomes

Fundamental na detecção de erros semânticos e geração de código.

Apresentado as questões do projeto, implementação, estrutura de dados e acesso, tamanho de tabela, única tabela x varias tabelas.

### Escopo x tabela de símbolos

- Operação **inserir**:
  - Não pode escrever por cima de declarações anteriores
  - Mas deve ocultá-las temporariamente
- Operação **verificar**:
  - Deve sempre acessar o escopo mais próximo (regra do aninhamento)
- Operação **remover**:
  - Deve remover apenas declarações no escopo mais próximo
  - Deve restaurar as declarações anteriormente ocultadas

No primeiro projeto que se cria trabalha com a tabela de símbolos e o conceito de pilhas de lista ligada, cada tabela de símbolos é para um único escopo.

Primeiro projeto feito sem utilizar o Antlr. Explicado como funciona

O segundo projeto é criado utilizando o Antlr, chamado 'alguma-semantico'.

Verificação de tipo:

```
public static TabelaDeSimbolos.TipoAlguma verificarTipo(TabelaDeSimbolos tabela,
AlgumaParser.ExpressaoAritmeticaContext ctx) {
    TabelaDeSimbolos.TipoAlguma ret = null;
    for (var ta : ctx.termoAritmetico()) {
//(necessário atualizar JAVA para funcionar a variável 'var')
        TabelaDeSimbolos.TipoAlguma aux = verificarTipo(tabela, ta);
        if (ret == null) {
            ret = aux;
        } else if (ret != aux && aux != TabelaDeSimbolos.TipoAlguma.INVALIDO) {
            adicionarErroSemantico(ctx.start, "Expressão " + ctx.getText() + " contém tipos
incompatíveis");
            ret = TabelaDeSimbolos.TipoAlguma.INVALIDO;
        }
    }
    return ret;
}
```