

AnalizadorSemantico.h

```
#ifndef ANALISADORSEMANTICO_H_
#define ANALISADORSEMANTICO_H_

#include <iostream>

#include "../analizador-sintatico/includes/ConteudoHash.h"
#include "../analizador-sintatico/includes/NoArvoreSintatica.h"
#include "../analizador-sintatico/includes/TabelaHash.h"

class AnalizadorSemantico
{
    public:
        AnalizadorSemantico( std::pair<TabelaHash*,
        NoArvoreSintatica*> );

        virtual
        ~AnalizadorSemantico( );

    protected:

    private:
        TabelaHash
        hash;

        NoArvoreSintatica*
        raiz;

        NoArvoreSintatica*
        noCorrente;

        unsigned int
        nivelLexicoAtual;

        void
        analise( );

        /*
        * Métodos vazios pois estas verificacoes ja
        * foram realizadas no 'analizador-sintatico'
        */
}
```

AnalizadorSemantico.h

```
void
verificaDeclaracao( NoArvoreSintatica* );

void
verificaRedeclaracao( NoArvoreSintatica* );

/*
 * Fim das verificacoes ja realizadas
 */

void
bloco( NoArvoreSintatica* );

void
declaracaoDeSubrotinas( NoArvoreSintatica* );

void
comandoComposto( NoArvoreSintatica* );

void
comando( NoArvoreSintatica* );

void
comandoSemRotulo( NoArvoreSintatica* );

void
comandoRepetitivo( NoArvoreSintatica* );

void
comandoCondicional( NoArvoreSintatica* );

void
comandoLeitura( NoArvoreSintatica* );

void
atribuicao( NoArvoreSintatica* );

std::string
chamadaFuncao( NoArvoreSintatica* );
```

AnalizadorSemantico.h

```
std::string
chamadaProcedimento( NoArvoreSintatica* );

std::string
expressao( NoArvoreSintatica* );

std::string
relacao( NoArvoreSintatica* );

std::string
expressaoSimples( NoArvoreSintatica* );

std::string
termo( NoArvoreSintatica* );

std::string
fator( NoArvoreSintatica* );
};

#endif /* ANALISADORSEMANTICO_H_ */
```