

**UNIVERSIDADE FEDERAL DE RORAIMA
DISCIPLINA CONSTRUÇÃO DE COMPILADORES
PROF.: DR. LUCIANO FERREIRA
ALUNO: FELIPE DERKIAN DE SOUSA FREITAS**

TRABALHO

SLIDE 11

BOA VISTA, 31 DE OUTUBRO DE 2020

1) Pesquise e proponha uma gramática LL(1), sem ambiguidades, e sem recursões a esquerda;

Gramática LL(1) para verificar balanço de parênteses na expressão.

(1) $S \rightarrow (S) S$

(2) $S \rightarrow \&$

2) TABELA SINTÁTICA DA GRAMÁTICA

	()	\$
S	1	2	-

3) Código Implementado

```
/**
 * Author: Felipe Derkian de Sousa Freitas
 *
 * Data: 31/10/2020
 *
 * Analisador preditivo para a gramática de balanço de parênteses na expressão
 *
 * ex: ( 4 + 3 * ( 4 - 2 ) )
 *
 */

#include <stack>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
using namespace std;
```

```

bool validateExpression(char *expression)
{
    stack<char> pilha;
    bool flag = true;

    for (int i = 0; i < expression[i] != '\0' && flag; i++)
    {
        cout << "> " << expression[i] << endl;
        switch (expression[i])
        {
            case '(':
                pilha.push('(');
                break;
            case ')':
                if (pilha.empty() || pilha.top() == ')')
                {
                    flag = false;
                }
                else if (pilha.top() == '(')
                {
                    pilha.pop();
                }
            }
        }

    return (pilha.empty() && flag) ? true : false;
}

int main()
{
    char expression[1000];

    cout << "Informe a sequencia de parenteses:\n>>> ";
    scanf("%s", expression);

    cout << (validateExpression(expression) ? "Balanceado" : "Nao Balanceado") << endl;

    return EXIT_SUCCESS;
}

```