



Teoria da Computação e Compiladores

GRUPO 7

GABRIEL BONIFÁCIO HULUANY GONZALEZ

125111372859

JOGO DE FUTEBOL AMERICANO



[GITHUB](#)

Como funciona o futebol americano?

Resumidamente

- O campo possui 100 jardas e você precisa avançá-lo completamente para marcar um Touchdown;
- É possível avançar correndo ou passando a bola;
- Um Touchdown vale 6 pontos + 1 ponto do chute extra, totalizando 7 pontos;



Gramática

Football.g4

```
grammar Football;

raiz_programa: jogo EOF;

jogo: inicio (jogada)* fim;

inicio: time ESPACO CONTRA ESPACO time NOVA_LINHA;

jogada: time ESPACO ACAO ESPACO QUANTIDADE ESPACO JARDAS NOVA_LINHA;

fim: FIM;

time: NFL;
NFL: 'Arizona Cardinals' | 'Cardinals' | 'Atlanta Falcons' | 'Falcons' | 'Baltimore Ravens' | 'Ravens' | 'Buffalo Bills' | 'Bills' |
'Carolina Panthers' | 'Panthers' | 'Chicago Bears' | 'Bears' | 'Cincinnati Bengals' | 'Bengals' | 'Cleveland Browns' | 'Browns' |
'Dallas Cowboys' | 'Cowboys' | 'Denver Broncos' | 'Broncos' | 'Detroit Lions' | 'Lions' | 'Green Bay Packers' | 'Packers' |
'Houston Texans' | 'Texans' | 'Indianapolis Colts' | 'Colts' | 'Jacksonville Jaguars' | 'Jaguars' | 'Kansas City Chiefs' | 'Chiefs' |
'Las Vegas Raiders' | 'Raiders' | 'Los Angeles Chargers' | 'Chargers' | 'Los Angeles Rams' | 'Rams' | 'Miami Dolphins' | 'Dolphins' |
'Minnesota Vikings' | 'Vikings' | 'New England Patriots' | 'Patriots' | 'New Orleans Saints' | 'Saints' | 'New York Giants' | 'Giants' |
'New York Jets' | 'Jets' | 'Philadelphia Eagles' | 'Eagles' | 'Pittsburgh Steelers' | 'Steelers' | 'San Francisco 49ers' | '49ers' |
'Seattle Seahawks' | 'Seahawks' | 'Tampa Bay Buccaneers' | 'Buccaneers' | 'Tennessee Titans' | 'Titans' | 'Washington Commanders' | 'Commanders';
CONTRA: '@';
NOVA_LINHA: '\n';
ESPACO: ' ';
ACAO: 'correu' | 'passou';
QUANTIDADE: [0-9][0-9]*;
JARDAS: 'jardas';
FIM: 'Fim de jogo';

//NOME: PALAVRA PALAVRA*;
//PALAVRA: [a-zA-Z_0-9][a-zA-Z_0-9]* ;
//ESPACO: [ \t\r\n]+ -> skip;
```


Entrada

Input.txt

```
Chicago Bears @ Green Bay Packers
Chicago Bears correu 3 jardas
Chicago Bears passou 4 jardas
Chicago Bears correu 6 jardas
Chicago Bears passou 14 jardas
Chicago Bears correu 51 jardas
Chicago Bears passou 25 jardas
Green Bay Packers correu 5 jardas
Green Bay Packers correu 2 jardas
Green Bay Packers passou 13 jardas
Green Bay Packers correu 2 jardas
Green Bay Packers correu 1 jardas
Green Bay Packers correu 0 jardas
Chicago Bears passou 1 jardas
Chicago Bears passou 2 jardas
Chicago Bears passou 3 jardas
Chicago Bears passou 0 jardas
Green Bay Packers correu 0 jardas
Chicago Bears passou 94 jardas
Chicago Bears passou 100 jardas
Chicago Bears passou 6 jardas
Green Bay Packers correu 23 jardas
Green Bay Packers passou 7 jardas
Green Bay Packers correu 30 jardas
Green Bay Packers passou 15 jardas
Green Bay Packers correu 12 jardas
Green Bay Packers correu 6 jardas
Green Bay Packers passou 10 jardas
Fim de jogo
```

Saída

Compilado.txt

```
Boas-vindas a Semana 9 da Temporada 2023 da NFL!
Chicago Bears @ Green Bay Packers
Chicago Bears correu 3 jardas
Chicago Bears passou 4 jardas
Chicago Bears correu 6 jardas
Chicago Bears passou 14 jardas
Chicago Bears correu 51 jardas
TOUCHDOWN! Chicago Bears passou 25 jardas
Green Bay Packers correu 5 jardas
Green Bay Packers correu 2 jardas
Green Bay Packers passou 13 jardas
Green Bay Packers correu 2 jardas
Green Bay Packers correu 1 jardas
Green Bay Packers correu 0 jardas
Chicago Bears passou 1 jardas
Chicago Bears passou 2 jardas
Chicago Bears passou 3 jardas
Chicago Bears passou 0 jardas
Green Bay Packers correu 0 jardas
Chicago Bears passou 94 jardas
TOUCHDOWN! Chicago Bears passou 100 jardas
Chicago Bears passou 6 jardas
Green Bay Packers correu 23 jardas
Green Bay Packers passou 7 jardas
Green Bay Packers correu 30 jardas
Green Bay Packers passou 15 jardas
Green Bay Packers correu 12 jardas
Green Bay Packers correu 6 jardas
TOUCHDOWN! Green Bay Packers passou 10 jardas
Fim de jogo
Resultado: Chicago Bears 14 x 7 Green Bay Packers
Jardas corridas: Chicago Bears 60 x 81 Green Bay Packers
Jardas passadas: Chicago Bears 249 x 45 Green Bay Packers
Jardas totais: Chicago Bears 309 x 126 Green Bay Packers
```

MeuCompilador.java

```
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Path;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.CharStreams;
import org.antlr.v4.runtime.CommonTokenStream;
import org.antlr.v4.runtime.tree.ParseTreeWalker;

public class MeuCompilador {
    public static void main(String[] args) {
        // Ler o texto a ser convertido transformando em Stream de "char"
        String filePath = "src/Input.txt";
        CharStream entrada = null;
        try {
            entrada = CharStreams.fromPath(Path.of(filePath), StandardCharsets.UTF_8);
        } catch (IOException e) {
            System.out.println("Erro de acesso ao arquivo: ");
            e.printStackTrace();
        }
        // Passar a Stream lida para o Lexer
        FootballLexer lexer = new FootballLexer(entrada);

        // Passando o resultado da análise léxica para o processador de Tokens
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        // Passando os tokens para o Parser
        FootballParser parser = new FootballParser(tokens);
        // Pegar o ponto raiz da gramática
        FootballParser.Raiz_programaContext arvore = parser.raiz_programa();
        // Imprimir a árvore geradora do texto sendo analisado
        System.out.println(arvore.toStringTree(parser));

        // "Caminha" pela árvore disparando os métodos do Listener
        MeuFootballListener meuListener = new MeuFootballListener();
        ParseTreeWalker parseTreeWalker = new ParseTreeWalker();
        parseTreeWalker.walk(meuListener, arvore);
    }
}
```

MeuFootballListener.java

```
// Generated from Football.g4 by ANTLR 4.13.1

import org.antlr.v4.runtime.ParserRuleContext;
import org.antlr.v4.runtime.tree.ErrorNode;
import org.antlr.v4.runtime.tree.TerminalNode;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;

public class MeuFootballListener extends FootballBaselister {
    FileWriter linguagemFinal = null;
    String time_visitante = "", time_mandante = "";
    int ataque_visitante = 0, ataque_mandante = 0, pontos_visitante = 0, pontos_mandante = 0, jardas_corridas_visitante = 0, jardas_corridas_mandante = 0,
    jardas_passadas_visitante = 0, jardas_passadas_mandante = 0;

    public MeuFootballListener() {
        try {
            linguagemFinal = new FileWriter (new File("src/Compilado.txt"));
        } catch (IOException e) {
            System.out.println("Erro de criação do arquivo: ");
            e.printStackTrace();
        }
    }

    @Override public void enterRaiz_programa(FootballParser.Raiz_programaContext ctx) { }

    @Override public void exitRaiz_programa(FootballParser.Raiz_programaContext ctx) { }

    @Override public void enterJogo(FootballParser.JogoContext ctx) { }

    @Override public void exitJogo(FootballParser.JogoContext ctx) { }
```

```

@Override public void enterInicio(FootballParser.InicioContext ctx) {
    try {
        org.antlr.v4.runtime.tree.ParseTree no0 = ctx.getChild(0);
        org.antlr.v4.runtime.tree.ParseTree no1 = ctx.getChild(1);
        org.antlr.v4.runtime.tree.ParseTree no2 = ctx.getChild(2);
        org.antlr.v4.runtime.tree.ParseTree no3 = ctx.getChild(3);
        org.antlr.v4.runtime.tree.ParseTree no4 = ctx.getChild(4);
        org.antlr.v4.runtime.tree.ParseTree no5 = ctx.getChild(5);
        if(no0 != null && no2 != null && no4 != null){
            Random random = new Random();
            int num = 1 + random.nextInt(18);
            linguagemFinal.write("Boas-vindas a Semana " + num + " da Temporada 2023 da NFL!\n" + no0.getText() + no1.getText() + no2.getText() +
no3.getText() + no4.getText() + no5.getText());
            linguagemFinal.flush();
            time_visitante = String.valueOf(no0.getText());
            time_mandante = String.valueOf(no4.getText());
        }
    } catch (IOException e) {
        System.out.println("Erro de escrita no arquivo");
        e.printStackTrace();
    }
}

```

```

@Override public void exitInicio(FootballParser.InicioContext ctx) { }

```

```

@Override public void enterJogada(FootballParser.JogadaContext ctx) {
    try {
        org.antlr.v4.runtime.tree.ParseTree no0 = ctx.getChild(0);
        org.antlr.v4.runtime.tree.ParseTree no1 = ctx.getChild(1);
        org.antlr.v4.runtime.tree.ParseTree no2 = ctx.getChild(2);
        org.antlr.v4.runtime.tree.ParseTree no3 = ctx.getChild(3);
        org.antlr.v4.runtime.tree.ParseTree no4 = ctx.getChild(4);
        org.antlr.v4.runtime.tree.ParseTree no5 = ctx.getChild(5);
        org.antlr.v4.runtime.tree.ParseTree no6 = ctx.getChild(6);
        org.antlr.v4.runtime.tree.ParseTree no7 = ctx.getChild(7);
        if(no0.getText().equals(time_visitante)){
            ataque_mandante = 0;
            if(ataque_visitante < 100) {
                ataque_visitante+= Integer.parseInt(String.valueOf(no4.getText()));
            }
        }
    }
}

```

```

        if(ataque_visitante >= 100) {
            ataque_visitante = 0;
            pontos_visitante += 7;
            linguagemFinal.write("TOUCHDOWN! ");
            linguagemFinal.flush();
        }
        if(no2.getText().equals("correu")){
            jardas_corridas_visitante+= Integer.parseInt(String.valueOf(no4.getText()));
        }
        if(no2.getText().equals("passou")){
            jardas_passadas_visitante+= Integer.parseInt(String.valueOf(no4.getText()));
        }
    }

    if(no0.getText().equals(time_mandante)){
        ataque_mandante = 0;
        if(ataque_mandante < 100) {
            ataque_mandante+= Integer.parseInt(String.valueOf(no4.getText()));
        }
        if(ataque_mandante >= 100) {
            ataque_mandante = 0;
            pontos_mandante += 7;
            linguagemFinal.write("TOUCHDOWN! ");
            linguagemFinal.flush();
        }
        if(no2.getText().equals("correu")){
            jardas_corridas_mandante+= Integer.parseInt(String.valueOf(no4.getText()));
        }
        if(no2.getText().equals("passou")){
            jardas_passadas_mandante+= Integer.parseInt(String.valueOf(no4.getText()));
        }
    }
    linguagemFinal.write(no0.getText() + no1.getText() + no2.getText() + no3.getText() + no4.getText() + no5.getText() + no6.getText() +
no7.getText());
    linguagemFinal.flush();
} catch (IOException e) {
    System.out.println("Erro de escrita no arquivo");
    e.printStackTrace();
}
}

```



```

@Override public void exitJogada(FootballParser.JogadaContext ctx) { }

@Override public void enterFim(FootballParser.FimContext ctx) {
    try {
        org.antlr.v4.runtime.tree.ParseTree no0 = ctx.getChild(0);
        linguagemFinal.write(no0.getText());
        linguagemFinal.write("\nResultado: " + time_visitante + " " + pontos_visitante + " x " + pontos_mandante + " " + time_mandante);
        linguagemFinal.write("\nJardas corridas: " + time_visitante + " " + jardas_corridas_visitante + " x " + jardas_corridas_mandante + " " +
time_mandante);
        linguagemFinal.write("\nJardas passadas: " + time_visitante + " " + jardas_passadas_visitante + " x " + jardas_passadas_mandante + " " +
time_mandante);
        linguagemFinal.write("\nJardas totais: " + time_visitante + " " + (jardas_corridas_visitante + jardas_passadas_visitante) + " x " +
(jardas_corridas_mandante + jardas_passadas_mandante) + " " + time_mandante);
        linguagemFinal.flush();
    } catch (IOException e) {
        System.out.println("Erro de escrita no arquivo");
        e.printStackTrace();
    }
}

@Override public void exitFim(FootballParser.FimContext ctx) { }

@Override public void enterTime(FootballParser.TimeContext ctx) { }

@Override public void exitTime(FootballParser.TimeContext ctx) { }

@Override public void enterEveryRule(ParserRuleContext ctx) { }

@Override public void exitEveryRule(ParserRuleContext ctx) {
    super.exitEveryRule(ctx);
}

@Override public void visitTerminal(TerminalNode node) { }

@Override public void visitErrorNode(ErrorNode node) { }
}

```