



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico

---

2 de Julio de 2014

Teoría de lenguajes

Integrante	LU	Correo electrónico
Axel Iglesias	79/10	axeligl@gmail.com
Agustin Martínez Suñé	630/11	agusmartinez.92@gmail.com
Nahuel Lascano	476/11	laski.nahuel@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
**Universidad de Buenos Aires**

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## 1. Introducción

## 2. Decisiones

## 3. Resultados

## 4. Gramática

El esqueleto para hacer la gramática<sup>1</sup> fue tomado de la página citada en la presentación<sup>2</sup>. Sobre este esqueleto armamos nuestra gramática, que tiene varios puntos de diferencia con la que muestran de ejemplo en ese sitio.

Nuestra gramática parte de un símbolo inicial *programa* que a su vez se divide en una lista de *funciones* y una instrucción de *ploteo*.

Una función consiste en la palabra 'function' (representada por el token terminal TFUNC), un *nombre* (que no es más que un terminal que matchea con strings no empezados por números), una lista de *argumentos* (lista de *nombres* separadas por comas) entre paréntesis y un *bloque* de código.

Un *bloque* es una lista (quizás vacía) de sentencias encerradas entre llaves o una única sentencia.

Una *sentencia* puede ser una *asignacion* (un *nombre*, un símbolo de = y una *expresin*), una sentencia *ifthenelse*, un *while* (que a su vez tiene una *condicion* y otro *bloque*) o un *return* (que lleva a su lado una *expresion*).

El caso del *ifthenelse* requiere un análisis específico. Un *ifthenelse* puede tener o no tener 'else', lo cual nos trae una ambigüedad a la hora de parsear cadenas del estilo

```
if cond1 then
if cond2 then
codigo1
else
codigo2
```

Entre los problemas que encontramos, no pudimos armar la expresión regular para que el lexer borre los comentarios multilínea. Es por esto que para pasar los tests el archivo debe ser en primer lugar pasado por un script que borra los comentarios.

## 5. Ejemplos de funcionamiento

## 6. Respuestas

## 7. Manual

Modo de uso:

```
make clean && make
cat archivodetest.cod | ./borracomentarios.sed | ./parser | ./graficador.sh
```

---

<sup>1</sup>implementada en parser.y

<sup>2</sup><http://gnu.org/2009/09/18/writing-your-own-toy-compiler/>