

# Prácticas de PROCESADORES DE LENGUAJES

Doble Grado Ingeniería Informática-Matemáticas

Facultad de Informática. *Profesor:* R. Peña

Abril de 2017

Todas las prácticas tienen en común la **definición** de un minilenguaje de programación, y el desarrollo de **un compilador** para el mismo que cubra las fases habituales. Se harán en **parejas**. Salvo excepciones acordadas con el profesor, el lenguaje de desarrollo será Java, y el lenguaje objeto será código-P, el lenguaje de la máquina virtual de Pascal.

Las fases de análisis léxico y sintáctico se harán utilizando las herramientas **JLex** y **CUP**. El resultado de las mismas será un **árbol de sintaxis abstracta**. Estas fases, y las de análisis de la semántica estática, informarán de los posibles errores, y en caso de haberlos no generarán código. Si no hay errores, el compilador generará un fichero de texto en el formato que se indica en el CV, que tendrá que ser cargado y ejecutado por la máquina-P, que también se proporciona en el CV.

A continuación se detallan los requisitos **mínimos** de cada faceta del lenguaje y los opcionales. Dadas las restricciones temporales, se aconseja ser prudente a la hora de elegir alguno(s) de estos últimos.

## 1. Requisitos mínimos y opcionales

### 1.1. Identificadores y ámbitos de definición

**Mínimos** Declaración de variables simples y de arrays de cualquier tipo, incluidos otros arrays. O bien, permitir arrays de varias dimensiones. Bloques anidados. Tabla de símbolos para bloques anidados.

**Opcionales** Punteros, registros, clases, procedimientos, funciones, módulos, cláusulas de importación.

### 1.2. Tipos

**Mínimos** Declaración explícita del tipo de las variables. Tipos predefinidos enteros y booleanos. Operadores infijos, con distintas prioridades y asociatividades para estos tipos. Tipos sin nombre. Equivalencia estructural de tipos. Algoritmo ascendente de comprobación de tipos.

**Opcionales** Tipos con nombre y definición de tipos de usuario. Equivalencia de tipos por nombre. Clases.

### 1.3. Instrucciones ejecutables

**Mínimos** Instrucción de asignación incluyendo elementos de arrays, condicional con una y dos ramas, y algún tipo de bucle. Expresiones formadas por constantes, identificadores con y sin subíndices y operadores infijos.

**Opcionales** Expresiones con punteros y nombres cualificados. Instrucción **case** o similar con salto a cada rama en tiempo constante. Llamadas a procedimientos, funciones, o/y métodos de clase. Instrucciones de reserva o/y liberación de memoria dinámica.

## 1.4. Errores

**Mínimos** Indicación del tipo de error, fila y columna. Parar la compilación.

**Opcionales** Tratar de proseguir la compilación tras un error, a fin de detectar más errores.

## 2. Entregables

Las entregas se harán a través del Campus Virtual por uno de los dos miembros de la pareja. Las tareas se cerrarán a las 23:55 del día especificado como límite para cada una.

1. Especificación en papel de la sintaxis del lenguaje y de ejemplos típicos de programas escritos en dicho lenguaje. Esta entrega necesitará el **visto bueno** del profesor para poder proseguir con la práctica.

**Fecha límite: martes 18 de Abril.** El visto bueno se dará en la clase del **viernes 21**.

2. Ficheros de especificación del léxico y de la sintaxis, con atributos para la construcción del árbol abstracto, procesados con éxito por JLex y CUP.

**Fecha límite: viernes 5 de Mayo**

3. Código de las fases de identificación de identificadores y de comprobación de tipos, con ejemplos pasados con éxito y con errores.

**Fecha límite: martes 16 de Mayo**

4. Código de la fase de generación de código, con ejemplos.

**Fecha límite: viernes 26 de Mayo**