

# Práctica de PROCESADORES DE LENGUAJES

Doble Grado Ingeniería Informática-Matemáticas

Facultad de Informática

Profesor: Albert Rubio

febrero de 2024

## 1. Introducción

Las prácticas se harán en grupos de un máximo de tres estudiantes y solo debe hacer la entrega uno de los miembros del grupo. Los miembros del grupo deberán indicarse en el documento de especificación entregado y será los mismos durante todo el desarrollo de la práctica.

Todas las prácticas tienen en común la definición de un mini-lenguaje de programación, y el desarrollo de un compilador para el mismo que cubra las fases vistas en clase. Las prácticas se harán en parejas. Salvo excepciones acordadas con el profesor, el lenguaje de desarrollo será Java, y el lenguaje objeto será WebAssembly (en su formato textual). Las fases de análisis léxico y sintáctico se harán utilizando las herramientas JFlex 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, que podrá ejecutarse desde Javascript (tendréis disponibles los scripts necesarios para hacerlo). 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.

## 2. Requisitos mínimos y opcionales del lenguaje diseñado

### 2.1. 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 (que requerirá bajar con tabla de símbolos para bloques anidados). Funciones (paso de parámetros por valor o por referencia).

**Opcionales:** Punteros, registros, clases (sin ningún tipo de herencia), módulos, cláusulas de importación.

**Grupos de 3:** Es muy recomendable incluir punteros y registros.

## 2.2. Tipos

**Mínimos:** Declaración explícita del tipo de las variables. Tipos básicos predefinidos enteros y booleanos. Operadores infijos, con distintas prioridades y asociatividades para estos tipos. Tipo array. Comprobación de tipos.

**Opcionales:** Definición de tipos de usuario.

## 2.3. Conjunto de instrucciones del lenguaje

**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 (para acceso a arrays), operadores infijos y llamadas a función.

**Opcionales:** Expresiones con punteros y nombres cualificados (notación “.” en presencia de clases o registros). Instrucción case o similar con salto a cada rama en tiempo constante. , o/y métodos de clase. Instrucciones de reserva de memoria dinámica (en presencia de punteros).

**Grupos de 3:** Es muy recomendable incluir todo lo relacionado con punteros y registros.

## 2.4. Gestión de errores

**Mínimos:** Indicación del tipo de error, fila y columna. Parar la compilación. Recuperación de errores (tratar de proseguir la compilación tras un error, a fin de detectar más errores).

# 3. Entregas

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, si se hace dentro de plazo, se revisará en clase de laboratorio. Fecha límite: 4 de marzo (aunque se recomienda, si es posible, entregarlo antes).
2. Ficheros de especificación del léxico y de la sintaxis, con atributos para la construcción del árbol de sintaxis abstracta, procesados con éxito por JFlex y CUP. Fecha límite: 5 de abril.
3. Código completo con generación de código WebAssembly y ejemplos. Fecha límite: 20 de Mayo.

Pre-entrega 13 de mayo. El día 13 de mayo se debe hacer una pre-entrega de una versión incompleta de la práctica pero funcional para código sencillo. Esta versión servirá para detectar problemas importantes que impidan la evaluación de la práctica. En tal caso, el grupo recibirá un mensaje indicando los problemas.