



UNIVERSIDADE DA CORUÑA

## DESEÑO DAS LINGUAXES DE PROGRAMACIÓN

### **Práctica 1: Análisis de las características de un conjunto de lenguajes de programación a partir de un caso práctico**

#### AUTORES

Rafael Alcalde Azpiazu  
rafael.alcalde.azpiazu  
(rafael.alcalde.azpiazu@udc.es)Eva

Suárez García  
eva.suarez.garcia  
(eva.suarez.garcia@udc.es)

#### LENGUAJES USADOS

Java  
C  
Python  
OCaml

A Coruña - September 16, 2017

# Contents

<b>1</b>	<b>Implementación en Java</b>	<b>2</b>
1.1	Descripción . . . . .	2
1.2	Análisis . . . . .	2
1.3	Justificaciones de diseño . . . . .	2
<b>2</b>	<b>Implementación en C</b>	<b>3</b>
2.1	Descripción . . . . .	3
2.2	Análisis . . . . .	3
2.3	Justificaciones de diseño . . . . .	3
<b>3</b>	<b>Implementación en Python</b>	<b>4</b>
3.1	Descripción . . . . .	4
3.2	Análisis . . . . .	4
3.3	Justificaciones de diseño . . . . .	4
<b>4</b>	<b>Implementación en OCaml</b>	<b>5</b>
4.1	Descripción . . . . .	5
4.2	Análisis . . . . .	5
4.3	Justificaciones de diseño . . . . .	5

# 1 Implementación en Java

Para este lenguaje se usó el compilador `javac` en su versión 1.8.0\_144.

## 1.1 Descripción

Java es un lenguaje de programación de alto nivel que sigue el paradigma de orientación a objetos basada en clases, aunque a día de hoy incluye elementos de otros paradigmas, como funciones-lambda.

Su sintaxis deriva en gran medida de C++, siendo un lenguaje de tipado estático fuerte. Es el propio lenguaje el que se encarga de la gestión de memoria mediante un recolector de basura, el cual es responsable de gestionar el ciclo de vida reservando memoria cuando el programador crea un objeto y liberando memoria cuando se referencia completamente al objeto.

Como en sus inicios se pensó en que fuera un lenguaje de propósito general, la compilación del código fuente se traduce a bytecode java, el cual es ejecutado sobre una máquina virtual, la cual si está diseñada para la plataforma final.

## 1.2 Análisis

## 1.3 Justificaciones de diseño

## **2 Implementación en C**

### **2.1 Descripción**

### **2.2 Análisis**

### **2.3 Justificaciones de diseño**

## **3 Implementación en Python**

### **3.1 Descripción**

### **3.2 Análisis**

### **3.3 Justificaciones de diseño**

## 4 Implementación en OCaml

### 4.1 Descripción

### 4.2 Análisis

### 4.3 Justificaciones de diseño