ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Tema 0:
¿Por qué Scala?

Charles Flores

¿Por qué Scala?

Objetivos del tema:

- Conocer conceptos básicos de los lenguajes interpretados
- Conocer conceptos básicos de los lenguajes compilados
- Qué tipo de lenguaje es Java
- Entender la integración de Scala con Java

¿Qué es la programación?

La programación es el proceso utilizado para:

- idear y ordenar las acciones necesarias para realizar un proyecto,
- preparar ciertas máquinas o aparatos para que empiecen a funcionar en el momento y en la forma deseados,
- **elaborar programas** para su empleo en computadoras.

¿Qué es la programación?

La programación se rige por un conjunto de reglas, instrucciones y expresiones acotadas que tienden a asemejarse a una lengua natural (por lo general, el inglés).

La potencia de un lenguaje de programación se caracteriza por el bajo nivel de ambigüedad, partiendo de esta premisa el lenguaje más potente sería el lenguaje binario, que nos hace pensar en el lenguaje ensamblador o lenguaje máquina.

Existen distintos lenguajes de programación de alto nivel y cada uno con su léxico propio, reglas semánticas y sintácticas.

¿Qué es la programación?

- ¿Qué es un algoritmo?

Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que se han de seguirse para resolver un problema

- ¿Qué es un programa?

Un programa normalmente implementa (traduce a un lenguaje de programación concreto) uno o más algoritmos para realizar una tarea específica en una computadora.

¿Qué es la programación?

La programación puede tener distintos enfoques, diversas maneras de formular la resolución de un problema dado. A estos enfoques también se les conoce como **paradigmas**.

Un paradigma se asocia un modelo de cómputo, pero también se puede asociar a un determinado estilo de programación.

Algunos ejemplos serían:

- programación declarativa
- programación modular
- programación orientada a objetos
- programación estructurada

¿Qué es la Programación Orientada a Objetos?

Es un paradigma de programación cuya característica es la forma de obtener los resultados.

El código se organiza en unidades denominadas clases, de las cuales se crean objetos.

Los objetos son entidades que tienen un determinado "estado (atributos)", "comportamiento (métodos)" e "identidad".

```
# Python
class MyClass:
  """A simple example class"""
  i = 12345

  def f(self):
    return 'hello world'
```

¿Qué es la Programación Orientada a Objetos?

La programación orientada a objetos está basada en técnicas de programación como:

- Herencia
- Cohesión
- Abstracción
- Polimorfismo
- Acoplamiento
- Encapsulamiento

¿Qué es la programación funcional?

La programación funcional es el paradigma de la programación declarativa basado en el uso verdadero de funciones matemáticas.

En este estilo de programación las funciones son ciudadanas de primera clase, porque sus expresiones pueden ser asignadas a variables.

Se pueden crear funciones de orden superior. Es decir, funciones que pueden tomar otras funciones como argumentos o devolverlos como resultados.

Ejemplo:

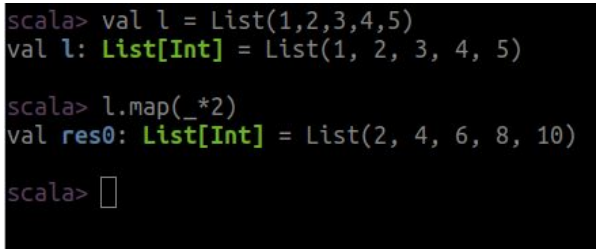
- En cálculo, una función de orden superior es el operador diferencial d/dx , que devuelve la derivada de una función f .

¿Qué es la programación funcional?

La programación funcional tiene sus raíces en el cálculo lambda, un sistema formal para investigar la naturaleza de las funciones, de la computabilidad y su relación con la recursión.

Los lenguajes funcionales priorizan el uso de recursividad y aplicación de funciones de orden superior para resolver problemas que en otros lenguajes se resolverían mediante estructuras de control (por ejemplo, bucles o ciclos).

Muchos lenguajes de programación funcionales pueden ser vistos como elaboraciones del cálculo lambda.



¿Qué es la programación funcional?

Características de la programación funcional:

- Funciones de primera clase y de orden superior.
- Recursividad.
- Evaluación estricta frente a la no estricta.
- Sistemas de tipos.

¿Qué es Scala?

Scala es un lenguaje de **programación multi-paradigma**, Orientado a Objetos y Funcional al menos.

Diseñado para expresar **patrones** comunes de programación en **forma concisa**, **elegante** y con **tipado estático**.

Todo código de Scala es compilado, generando un bytecode que se ejecuta en una máquina virtual de Java, lo cual garantiza la interoperabilidad entre ambos.

¿Qué lenguaje: R, Python o Scala?

Tres lenguajes de programación más populares en el ámbito la ciencia de datos. Los tres son multiparadigma, pero solventan casos de uso distintos

- **R**: es un lenguaje de **programación estadístico**. Soporta los paradigmas funcional (parcialmente) y orientada a objetos.
- **Python**: es un lenguaje de **programación interpretado multiparadigma** que soporta mejor la programación orientada objetos que la funcional.
- **Scala**: lenguaje de **programación de propósito general y multiparadigma** que soporta tanto la programación orientada a objetos como la funcional.

¿Qué lenguaje: R, Python o Scala?

- **R**: junto con su IDE de desarrollo (RStudio) son una herramienta potente para trabajo de **análisis de datos**, **cálculo estadísticos** y **gráficos de los datos**. También se puede hacer Machine Learning con R, con librerías como `caret`, `rpart` o `e1071`.
- **Python**: en comparación con R tiene mejor background para trabajar con **datos de tamaño mediano**. Así como R, cuenta con librerías como `NLTK`, `matplotlib`, `numpy` o `pandas` que facilitan el analizar datos o graficarlos.
- **Scala**: encaja mejor en el **escenario de Big Data**, debido a que tiene el paralelismo intrínseco y fácil escalabilidad. Facilidad de integración con **Apache Spark** (Framework de computación en distribuido)

¿Qué lenguaje: R, Python o Scala?

¿Y por qué Scala?

- Es conciso. No hace falta escribir demasiado código con este lenguaje.
- Tiene inferencia de tipos.
- Tiene un tipado estático.
- Paralelismo intrínseco.
- Compila contra la JVM y es compatible con Java.
- De fácil escalabilidad.
- Integración nativa con Spark.

Interactúa con Scala

Para empezar a interactuar con Scala nos hace falta tener un intérprete de Scala. Existen intérpretes online que no requieren realizar ninguna instalación que nos permite tener un primer contacto, como por ejemplo, pero se recomienda realizar la instalación de Scala y/o del IDE.

- online: https://scastie.scala-lang.org/
- IDE: Instalación de IntelliJ
- REPL: Intérpretes de Scala, como: consola de sbt o ammonite

Interactúa con Scala: Pre-requisitos

Antes de instalar sbt o IntelliJ, hay que tener, como mínimo, Java JDK 8 o Java JDK 11 instalado en el sistema:

- Se verifica la versión de Java instalada; accede a una consola de comandos y ejecuta: `java -version`

```
~$ java -version
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-1~18.04-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
```

- Si no se obtiene un resultado como el anterior, se debe proceder la instalación de Java 8 o Java 11 dependiendo del sistema operativo que tengas.

Interactúa con Scala: IntelliJ

Ahora hay que **instalar el IDE** de Scala: IntelliJ, dependiendo del sistema operativo que tengas hay que descargar el fichero de instalación y seguir sus instrucciones de instalación:

- windows
- linux:
  - en caso de que cuentes con Ubuntu 16 o superior, desde la línea de comandos ejecuta: `sudo snap install intellij-idea-community --classic`
  - en caso contrario, procede a descargar el instalable
- mac

Interactúa con Scala: REPL

REPL son las siglas de: Read Eval Print Loop, es la manera en la que se le conoce al intérprete interactivo de Scala. Existen varios, pero de los más populares son la consola de sbt (Scala Build Tool) y ammonite, aunque este segundo está completamente soportado en Linux y Mac, pero Windows aún está en fase experimental.

Son interfaces de línea de comando que te permite interactuar de manera rápida y sencilla, para testear código sin necesidad de un IDE y será lo que se use para los siguientes ejemplos.

- sbt: https://www.scala-sbt.org/1.x/docs/Setup.html
- ammonite: https://ammonite.io/#Ammonite-REPL

Interactúa con Scala

Procedemos a abrir una consola de línea de comandos y ejecutamos `sbt`, obteniendo una salida como la siguiente, la primera puede tardar unos minutos:

```
cflores@cflores:~$ sbt
[info] Loading global plugins from /home/cflores/.sbt/1.0/plugins
[info] Loading project definition from /home/cflores/project
[info] Set current project to cflores (in build file:/home/cflores/)
[info] sbt server started at local:///home/cflores/.sbt/1.0/server/4f8ede1b01ac5fb52e74/sock
sbt:cflores>
```

Al ejecutar sbt se ha accedido a la consola de SBT, ahora para acceder al Scala REPL, ejecutamos `console` e inmadiatamente se podría ejecutar una suma: 1+1:

```
[info] sbt server started at local:///home/cflores/.sbt/1.0/server/4f8ede1b01ac5fb52e74/sock
sbt:cflores> console
[info] Starting scala interpreter...
Welcome to Scala 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_252).
Type in expressions for evaluation. Or try :help.

scala> 1+1
res0: Int = 2

scala>
```


ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID