

# Evaluación Cursos[1,2] de Scala

## Lista y operaciones sobre ella

EPAM Latam - S4N Campus

23 de Septiembre 2021

### 1. Preliminares

1. Cree un repositorio en [GitHub](#) con el nombre <emailusuario>-calnatsuma. Donde <emailusuario> es el nombre de usuario en el su email de la compañía. Por ejemplo el nombre del repositorio de juancardona@seven4n.com sería juancardona-listainversa.
2. Clone el repositorio en un directorio de trabajo.
3. Abra dicho directorio y pueble con la estructura de directorios que se observa [1](#).
4. Con su editor favorito cree los archivos que esta en data que deben seguir el siguiente formato. Un número por línea.

1  
2  
3  
4  
5

### 2. Definición de tipos algebraicos las listas

Vamos a utilizar un concepto nuevo que son los tipos algebraicos que nos permiten definir tipos recursivos, como las listas.

```
(base) [juancardona@localhost listainversa]$ tree
.
├── build.sbt
├── data
│   ├── numeros1.txt
│   └── numeros2.txt
├── docs
│   └── listainversa.tex
├── src
│   └── main
│       ├── scala
│       │   └── co
│       │       └── s4n
│       │           └── listas
│       │               ├── Lista.scala
│       │               └── Main.scala
└── 8 directories, 6 files
(base) [juancardona@localhost listainversa]$
```

Figura 1: Estructura de directorios

## 2.1. Definición de Listas en Scala

Aunque las lista ya existe en Scala, y son mejor implementadas vamos a hacer una versión diferente para esta problema en particular.

```
1 package co.s4n.listas
3 sealed trait Lista
4 case class Vacia() extends Lista
5 case class Cons(i: Int, lst: Lista) extends Lista
```

Donde Lista es el tipo. Vacia y Const son los constructores de valores.

Una posible iteracción con las listas de puede hacer así:

```
scala> val lstVacía = Vacía()
val cero: co.s4n.listas.Vacia = Vacía()
```

```
scala> val listaUnitaria = Cons(1, Vacia())
val uno: co.s4n.listas.Cons = Cons(1, Vacia())
scala> val listaDeTres = Cons(1,Cons(2,Cons(3,Vacia())))
val cero: bco.s4n.listas.Cons = Cons(1,Cons(2,Cons(3,Vacia())))
```

## 2.2. Coincidencia de patrones en las listas propuestas

La función `esCero` nos muestra como implementar la coincidencia de patrones en la lista propuesta.

```
1 def longitud(lst: Lista): Int = lst match {
   case Vacia()      => 0
3  case Cons(i, lst) => 1 + longitud
}
```

## 3. Proyecto a implementar

En el archivo `Main` debe tener implementar el objeto `Main` como lo hizo anteriormente cuando utilizo `sbt`.

El esqueleto del `Main` se muestra a continuación:

```
package co.s4n.listas
2
import scala.io.Source
4
object Main extends App {
6  def deListALista(lst: List[Int]): Lista = lst match {
   case Nil => Vacia()
8   case (i :: lstp) => Cons(i, deListALista(lstp))
   }
10 def leerArchivo(src: String): Lista =
   deListALista(Source.fromFile(src).getLines().toList.map(_
   toInt))
12 def concatenar(lst1: Lista, lst2: Lista): Lista = ???
13 def imprimirLista(lst: Lista): String = ???
14 def invertirLista(lst: Lista): Lista = ???
15 val lista = leerArchivo(args(0))
16
17 ...
18 }
```

---

Usted debe implementar los siguientes métodos:

- `imprimirLista` imprime una lista de enteros, comenzando por “[“ y terminando “]”, donde cada número es separado por coma (“,”).
- `concatenar` toma dos listas y produce una única lista.
- `invertirLista` toma una lista y la transforma en su inversa.

Un vez implementado los anteriores métodos los utiliza para implementar el siguiente programa principal.

El programa un archivo que contiene una lista, la imprime, la invierte y la nueva lista invertida la imprime.

Una posible iteración de su programa *podría* ser similar a esta:

```
(base) [juancardona@fedora listainversa]$ sbt "run data/numeros1.txt"
[info] welcome to sbt 1.4.2 (AdoptOpenJDK Java 11.0.9)
[info] loading project definition from /home/juancardona/Workbench/s4n_scala_synch_course/proyecto
[info] loading settings for project listainversa from build.sbt ...
[info] set current project to listainversa (in build file:/home/juancardona/Workbench/s4n_scala_synch_course/)
[info] running co.s4n.listas.Main data/numeros1.txt
[5,4,3,2,1,0]
[0,1,2,3,4,5]
[success] Total time: 0 s, completed 23/09/2021, 12:29:32 p. m.
(base) [juancardona@fedora listainversa]$
```