EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México
Instituto Tecnológico de Tijuana

Subdirección Académica
Departamento de Sistemas y Computación

Semestre:
Agosto – Diciembre 2021

Carrera:
Ingeniería en Tecnologías de la Información y Comunicaciones

Materia y serie:
Datos Masivos BDD-1704TI9A

Unidad a evaluar: Unidad I

Nombre de la Tarea:
Practice 3

Nombre del Alumno:
Flores Gonzalez Luis Diego C16211486
Sifuentes Martinez Manuel Javier 17212934

Nombre del docente:
José Christian Romero Hernández

Practice 3

1. Create a list called "list" with the elements "red", "white "," black ":
A mutable variable is declared where we enter the previous colors.

```
var list = List("red", "white", "negro")
```

Result:

```
scala> var list = List ( "red", "white", "black")
list: List [String] = List (red, white , black)
```

2. Add 5 more elements to "list" "green", "yellow", "blue", "orange", "pearl":
In this case the ": +" is used to add an element to the list Previous, consists of renewing the elements of the main list variable which changes as new elements are added.

```
list = list: + "green"
list = list: + "yellow"
list = list: + "blue"
list = list: + "orange"
list = list: + "pearl"
```

Result:

```
scala> list = list: + "blue"
list: List [String] = List (red, white, black, green, yellow, blue)

scala> list = list: + "orange"
list: List [String] = List (red, white, black, green, yellow, blue, orange)

scala> list = list: + "pearl"
list: List [String] = List (red, white, black, green, yellow, blue, orange,
pearl )
```

3. Get the elements of "list" "green", "yellow", "blue":
The simple way to show an element of a list is to specify the index of the element to search in parentheses.

```
list (3)
list (4)
list (5)
```

Result:

```
scala> list (3)
res1: String = green

scala> list (4)
res2: String = yellow
```

```
scala> list (5)
res3: String = blue
```

4. Create an array of numbers in the range 1-1000 in steps of 5 to 5.
An array is created, but with its range function, which, among its properties, is to allow creating arrangements with jumps that the user decides, in this case, A 5 is passed as the second parameter so that they are jumps of 5 in 5.

```
val arr = Array.range (0, 1001, 5)
```

5. What are the unique elements of the list List (1,3,3,4,6,7,3,7) use conversion to sets.
Lists have a function called toSet, which acts in the same way as "Set" when used in an array of data, so its function is only accessed and returns a set of non-repeating numbers within the list.

```
val List = List(1,3,3,4,6,7,3,7)
List.toSet

scala> List.toSet
res1: scala.collection.immutable.Set[Int] = Set(1, 6, 7, 3, 4)
```

6. Create a mutable map named names that contains the following "Jose", 20, "Luis", 24, "Ana", 23, "Susana" , "27".
To create a mutable map, just specify through "collection.mutable" this way it makes a map mutable, in other words, it can be modified.

```
val mutmap = collection.mutable.Map(( "Jose", 20), ("Luis", 24), ("Ana",
23), ("Susana", "27"))
```

6 a. Print all the keys on the map.
The "keys" function contained within the maps is accessed and the existing keys are listed.

```
scala> mutmap.keys
res0: Iterable[String] = Set(Susana, Ana, Luis, Jose)
```

6 b. Add the following value to the map ("Miguel", 23).
As it is mutable, through the expression “+ =” another value can be added to a map.

```
scala> mutmap + = ("Miguel" -> 23)
res1: mutmap.type = Map(Susana -> 27, Ana -> 23, Miguel -> 23, Luis ->
24, Jose -> 20)
```