

refactoring

March 19, 2021

1 CASO PRÁCTICO 1

Se solicita refactorizar el siguiente código para validar accesos de un usuario

```
[1]: object repository {  
    def getUser(userNme: String): String = userNme.split("@")(0)  
  
    def login(usr: String, pwd: String): Boolean = {  
        var a = false  
        if (usr == "awainer") {  
            if (pwd == "123") {  
                a = true  
            }  
        }  
        return a  
    }  
}
```

defined object repository

```
[2]: def shouldBechecking_login(userNme: String, pwd: String) {  
    val usr = repository.getUser(userNme)  
    var msg = ""  
    var result = false  
    if (usr != null) {  
        if (repository.login(usr, pwd)) {  
            println("Válido")  
            return true  
        }  
    }  
    else {  
        msg = "Invalid user or password"  
        result = false  
    }  
}
```

```

else {
  msg = "Invalid user"
  result = false
}
if (msg != null) {
  println(msg)
  return false
}
}

```

<console>:34: warning: a pure expression does nothing in statement position; you
 ↳ may be omitting necessary parentheses

```

    return true
    ^

```

<console>:34: warning: enclosing method shouldBechecking_login has result type
 ↳ Unit: return value discarded

```

    return true
    ^

```

<console>:51: warning: a pure expression does nothing in statement position; you
 ↳ may be omitting necessary parentheses

```

    return false
    ^

```

<console>:51: warning: enclosing method shouldBechecking_login has result type
 ↳ Unit: return value discarded

```

    return false
    ^

```

shouldBechecking_login: (userNme: String, pwd: String)Unit

```
[3]: shouldBechecking_login("awainer@demo.com", "1234")
```

Invalid user or password

```
[4]: shouldBechecking_login("", "123")
```

Invalid user or password

2 CASO PRÁCTICO 2

Se solicita refactorizar el siguiente código para generar estadísticas de trazabilidad de un dataframe

```
[5]: val marco_de_datos = spark.read.parquet("processed")
```

```
marco_de_datos = [campaign_start_date: date, personal_type: string ... 4 more
↪fields]
```

```
[5]: [campaign_start_date: date, personal_type: string ... 4 more fields]
```

```
[6]: import org.apache.spark.sql.DataFrame
import org.apache.spark.util.SizeEstimator
import org.apache.spark.sql.functions.desc

def dfStats(marco_de_datos:DataFrame, PRM_SPARK_STATS_LEVEL:Int,
↪spark_partition_id:String):Unit = {

    val dfPartitionNum = marco_de_datos.rdd.getNumPartitions

    if (PRM_SPARK_STATS_LEVEL>0) {
        val dfSizeByte: BigInt = marco_de_datos.queryExecution.optimizedPlan.stats.
↪sizeInBytes
        val dfSizeMb: Float = dfSizeByte.toFloat / 1024.toFloat / 1024.toFloat
        val dfSizeEstimatorMb: Float = SizeEstimator.estimate(marco_de_datos).
↪toFloat / 1024.toFloat / 1024.toFloat

        println("*** STATS *** Df Stats PlanSizeMB:"+dfSizeMb+", EstimatorSizeMB:
↪"+dfSizeEstimatorMb+", PartitionNum: "+dfPartitionNum
            +", PartitionSizeMB:"+dfSizeMb / dfPartitionNum+", PartitionSizeMB:
↪"+dfSizeEstimatorMb / dfPartitionNum)
    }

    if (PRM_SPARK_STATS_LEVEL==2 || PRM_SPARK_STATS_LEVEL==4) {
        println(s"*** STATS *** Df Partitioon record count ${marco_de_datos.
↪count()}")
    }

    // check the partitions data
    if (PRM_SPARK_STATS_LEVEL==3 || PRM_SPARK_STATS_LEVEL==4) {
        println(s"*** STATS *** Df Partitioon Info")
        marco_de_datos
            .groupBy(spark_partition_id)
            .count()
            .orderBy(desc("count"))
            .show(if (dfPartitionNum>0) dfPartitionNum else 200)
    }

    if (PRM_SPARK_STATS_LEVEL==8) {
        println(s"*** STATS *** Show data:")
        marco_de_datos.show()
    }
}
```

```
}
```

```
dfStats: (marco_de_datos: org.apache.spark.sql.DataFrame, PRM_SPARK_STATS_LEVEL: Int, spark_partition_id: String)Unit
```

```
[7]: dfStats(marco_de_datos, 1, "cutoff_date")
```

```
*** STATS *** Df Stats PlanSizeMB:9.864668, EstimatorSizeMB:138.33743,
PartitionNum: 4, PartitionSizeMB:2.466167, PartitionSizeMB:34.58436
```

```
[8]: dfStats(marco_de_datos, 2, "cutoff_date")
```

```
*** STATS *** Df Stats PlanSizeMB:9.864668, EstimatorSizeMB:138.45236,
PartitionNum: 4, PartitionSizeMB:2.466167, PartitionSizeMB:34.61309
*** STATS *** Df Partition record count 1282382
```

```
[9]: dfStats(marco_de_datos, 3, "cutoff_date")
```

```
*** STATS *** Df Stats PlanSizeMB:9.864668, EstimatorSizeMB:138.88104,
PartitionNum: 4, PartitionSizeMB:2.466167, PartitionSizeMB:34.72026
*** STATS *** Df Partition Info
+-----+-----+
|cutoff_date| count|
+-----+-----+
| 2020-01-03|1282382|
+-----+-----+
```

```
[10]: dfStats(marco_de_datos, 8, "cutoff_date")
```

```
*** STATS *** Df Stats PlanSizeMB:9.864668, EstimatorSizeMB:139.5444,
PartitionNum: 4, PartitionSizeMB:2.466167, PartitionSizeMB:34.8861
*** STATS *** Show data:
+-----+-----+-----+-----+-----+-----+
-----+-----+
|campaign_start_date|personal_type|personal_id| segment_desc|
audtiminsert_date|cutoff_date|
+-----+-----+-----+-----+-----+-----+
-----+-----+
|          2019-09-30|          L|    00081633| 08. Bajo ingreso|2019-11-07
19:12:...| 2020-01-03|
|          2019-09-30|          L|    00950462| 08. Bajo ingreso|2019-11-07
19:12:...| 2020-01-03|
|          2019-09-30|          L|    02446085|          None|2019-11-07
19:12:...| 2020-01-03|
|          2019-09-30|          L|    06674137|          None|2019-11-07
```

19:12:...	2020-01-03			
	2019-09-30	L	06694902	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	07247000	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	07699179	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	08464270	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	16769553	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	17804824	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	20709077 11. IND - NO SUNAT	2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	21564496	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	25809138 11. IND - NO SUNAT	2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	29204932	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	29416368	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	40150611	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	40169609	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	41051997	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	41630769	None 2019-11-07
19:12:...	2020-01-03			
	2019-09-30	L	43192411 11. IND - NO SUNAT	2019-11-07
19:12:...	2020-01-03			

+-----+-----+-----+-----+-----+
+-----+-----+

only showing top 20 rows