



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

NOMBRE DE LOS ALUMNOS:

GALAVIZ LONA OSCAR EDUARDO (N.CONTROL: 17212993)

MARQUEZ MILLAN SEASHELL VANESSA (N.CONTROL: )

Carrera: Ingeniería Informática

Semestre: 9no

MATERIA: Datos Masivos

PROFESOR: JOSE CHRISTIAN ROMERO HERNANDEZ

Practice evaluatoria 3

Unidad 3

## Development

```
// 1-. Importar una simple sesión Spark.
import org.apache.spark.sql.SparkSession

// 2-. Utilice las líneas de código para minimizar errores
import org.apache.log4j._
Logger.getLogger("org").setLevel(Level.ERROR)

// 3-. Cree una instancia de la sesión Spark
val spark = SparkSession.builder().getOrCreate()

// 4-. Importar la librería de Kmeans para el algoritmo de agrupamiento.
import org.apache.spark.ml.clustering.KMeans

// 5-. Carga el dataset de Wholesale Customers Data
val dataset =
spark.read.option("header","true").option("inferSchema","true").format("csv").load
("Wholesale customers data.csv")
```



```
// 6-. Seleccione las siguientes columnas: Fresh, Milk, Grocery, Frozen,
Detergents_Paper, Delicassen y llamar a este conjunto feature_data
val feature_data = (dataset.select($"Fresh", $"Milk", $"Grocery", $"Frozen",
$"Detergents_Paper", $"Delicassen"))
```



```
// 7-. Importar Vector Assembler y Vector
import org.apache.spark.ml.feature.VectorAssembler
import org.apache.spark.ml.linalg.Vectors

//8-.Crea un nuevo objeto Vector Assembler para las columnas de características
como un conjunto de entrada, recordando que no hay etiquetas
val assembler = new
VectorAssembler().setInputCols(Array("Fresh","Milk","Grocery","Frozen","Detergents
_Paper","Delicassen")).setOutputCol("features")
```



```
//9-.Utilice el objeto assembler para transformar feature_data
val features = assembler.transform(feature_data)

//10-.Crear un modelo Kmeans con K=3
```

```
val kmeans = new KMeans().setK(3).setSeed(1L)
val model = kmeans.fit(features)
```



```
//11-.Evalúe los grupos utilizando Within Set Sum of Squared Errors WSSSE e
imprima los centroides.
val WSSSE = model.computeCost(features)
println(s"Within set sum of Squared Errors = $WSSSE")
```



```
println("Cluster Centers: ")
model.clusterCenters.foreach(println)
```

