```
In [ ]: from pyspark.sql import SparkSession

        from pyspark.sql.functions import *
        from pyspark.sql.types import *

        spark = SparkSession \
            .builder \
            .appName("how to read csv file") \
            .getOrCreate()
```

Cargo los datos en un dataframe de spark

```
In [ ]: df = spark.read.csv('/user/ort/obligatorio/monthly_pays.csv', header=True)
```

Vista previa de las primeras filas

```
In [ ]: df.show()
```

```
+-------+----+----------------+--------------+-------------+---------------------+-----------------------------+
|user_id|plan|mensual_cost_usd|local_currency|      country|has_monthly_purchases|purchase_value_local_currency|
+-------+----+----------------+--------------+-------------+---------------------+-----------------------------+
|      1|   C|          $38.90 |         Zloty|       Poland|                FALSE|                       $0.00 |
|      2|   D|          $62.02 |          Peso|     Colombia|                FALSE|                       $0.00 |
|      3|   C|          $99.53 |        Rupiah|    Indonesia|                 TRUE|                     $768.87 |
|      4|   C|         $128.35 |         Naira|      Nigeria|                FALSE|                       $0.00 |
|      5|   D|          $46.69 | Yuan Renminbi|        China|                 TRUE|                   $1,558.45 |
|      6|   A|          $43.57 |        Dollar|United States|                FALSE|                       $0.00 |
|      7|   A|          $38.71 |         Franc|     Cameroon|                FALSE|                       $0.00 |
|      8|   B|         $118.55 |         Ruble|       Russia|                FALSE|                       $0.00 |
|      9|   D|         $143.93 |          Euro|      Germany|                FALSE|                       $0.00 |
|     10|   C|          $62.66 |       Hryvnia|      Ukraine|                 TRUE|                   $1,221.21 |
|     11|   B|         $113.58 |        Rupiah|    Indonesia|                 TRUE|                   $2,257.96 |
|     12|   B|         $111.53 |         Franc|  Ivory Coast|                FALSE|                       $0.00 |
|     13|   D|          $61.68 |         Ruble|       Russia|                FALSE|                       $0.00 |
|     14|   C|          $75.93 |          Real|       Brazil|                 TRUE|                     $427.86 |
|     15|   B|          $81.10 |          Baht|     Thailand|                 TRUE|                   $1,218.14 |
|     16|   A|          $99.95 |          Peso|  Philippines|                 TRUE|                     $801.58 |
|     17|   D|          $80.99 |         Krona|       Sweden|                 TRUE|                   $1,439.72 |
|     18|   D|          $77.48 | Yuan Renminbi|        China|                FALSE|                       $0.00 |
|     19|   C|         $149.26 |         Ruble|       Russia|                FALSE|                       $0.00 |
|     20|   A|          $92.08 |          Peso|  Philippines|                FALSE|                       $0.00 |
+-------+----+----------------+--------------+-------------+---------------------+-----------------------------+
only showing top 20 rows
```

Cantidad de columnas del dataframe monthly_pays

In [ ]:
```
num_columns=len(df.columns)
num_columns
```

Out[ ]:
```
7
```

Nombre de las columnas de monthly_pays

In [ ]:
```
df.columns
```

```
Out[ ]: ['user_id',
         'plan',
         'mensual_cost_usd',
         'local_currency',
         'country',
         'has_monthly_purchases',
         'purchase_value_local_currency']
```

Descripción de los datos de la tabla

```
In [ ]: df.describe
```

```
Out[ ]: <bound method DataFrame.describe of DataFrame[user_id: string, plan: string, mensual_cost_usd: string, local_currenc
        y: string, country: string, has_monthly_purchases: string, purchase_value_local_currency: string]>
```

Schema de la tabla

```
In [ ]: df.printSchema()
```

```
root
 |-- user_id: string (nullable = true)
 |-- plan: string (nullable = true)
 |-- mensual_cost_usd: string (nullable = true)
 |-- local_currency: string (nullable = true)
 |-- country: string (nullable = true)
 |-- has_monthly_purchases: string (nullable = true)
 |-- purchase_value_local_currency: string (nullable = true)
```

```
In [ ]: from pyspark.sql.functions import col, regexp_replace

        df = df.withColumn("mensual_cost_usd", regexp_replace(col("mensual_cost_usd"), "\\$", ""))

        df = df.withColumn("mensual_cost_usd", col("mensual_cost_usd").cast("int"))
```

```
In [ ]: df = df.withColumn("purchase_value_local_currency", regexp_replace(col("purchase_value_local_currency"), "\\$", ""))

        df = df.withColumn("purchase_value_local_currency", regexp_replace(col("purchase_value_local_currency"), ",", "").cas
```

```
In [ ]: df = df.withColumn("has_monthly_purchases", col("has_monthly_purchases").cast("boolean"))
```

In [ ]:
```
df.printSchema()
```

```
root
 |-- user_id: string (nullable = true)
 |-- plan: string (nullable = true)
 |-- mensual_cost_usd: integer (nullable = true)
 |-- local_currency: string (nullable = true)
 |-- country: string (nullable = true)
 |-- has_monthly_purchases: boolean (nullable = true)
 |-- purchase_value_local_currency: float (nullable = true)
```

Valores Nulos o Faltantes

In [ ]:
```
total_nulos = df.select([sum(col(c).isNull().cast("int")).alias(c) for c in df.columns])

total_nulos.show()
```

```
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
|user_id|plan|mensual_cost_usd|local_currency|country|has_monthly_purchases|purchase_value_local_currency|
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
|      0|   0|               0|            15|      0|                    0|                            0|
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
```

In [ ]:
```
from pyspark.sql.window import Window
from pyspark.sql import functions as F

windowSpec = Window().partitionBy("country")

monthly_pays_refined = df.withColumn(
    "local_currency",
    F.when(F.col("local_currency").isNull(), F.first("local_currency", True).over(windowSpec)).otherwise(F.col("local
)

monthly_pays_refined.show()
```

```
+-------+----+----------------+--------------+-------------+---------------------+---------------------------+
|user_id|plan|mensual_cost_usd|local_currency|      country|has_monthly_purchases|purchase_value_local_currency|
+-------+----+----------------+--------------+-------------+---------------------+---------------------------+
|    783|   B|             113|       Afghani|  Afghanistan|                 true|                     1209.0|
|    829|   C|             140|       Afghani|  Afghanistan|                false|                        0.0|
|    835|   D|             100|       Afghani|  Afghanistan|                 true|                    2785.16|
|    412|   C|              36|          Euro|Aland Islands|                 true|                    2320.89|
|     67|   B|             149|           Lek|      Albania|                 true|                    1535.58|
|    104|   D|              48|           Lek|      Albania|                 true|                    2433.05|
|    879|   A|              78|           Lek|      Albania|                 true|                    1465.89|
|     47|   B|              99|          Peso|    Argentina|                false|                        0.0|
|     52|   C|             111|          Peso|    Argentina|                 true|                     259.11|
|    111|   D|             121|          Peso|    Argentina|                false|                        0.0|
|    143|   C|             104|          Peso|    Argentina|                 true|                     699.79|
|    164|   D|              66|          Peso|    Argentina|                 true|                    1272.38|
|    262|   B|              92|          Peso|    Argentina|                 true|                     2472.3|
|    306|   A|              76|          Peso|    Argentina|                false|                        0.0|
|    323|   C|              41|          Peso|    Argentina|                false|                        0.0|
|    339|   D|             143|          Peso|    Argentina|                 true|                     157.22|
|    463|   C|              42|          Peso|    Argentina|                false|                        0.0|
|    512|   D|             122|          Peso|    Argentina|                 true|                    2220.36|
|    518|   D|              90|          Peso|    Argentina|                 true|                     2297.9|
|    561|   B|             119|          Peso|    Argentina|                 true|                    2002.05|
+-------+----+----------------+--------------+-------------+---------------------+---------------------------+
only showing top 20 rows
```

In [ ]:
```
total_nulos = monthly_pays_refined.select([sum(col(c).isNull().cast("int")).alias(c) for c in df.columns])

total_nulos.show()
```

```
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
|user_id|plan|mensual_cost_usd|local_currency|country|has_monthly_purchases|purchase_value_local_currency|
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
|      0|   0|               0|             0|      0|                    0|                            0|
+-------+----+----------------+--------------+-------+---------------------+-----------------------------+
```

In [ ]:
```
hdfs_path = "/user/ort/obligatorio/refined/refined_monthly_pays/"
monthly_pays_refined.write.csv(hdfs_path, header=False, mode="overwrite")
```