

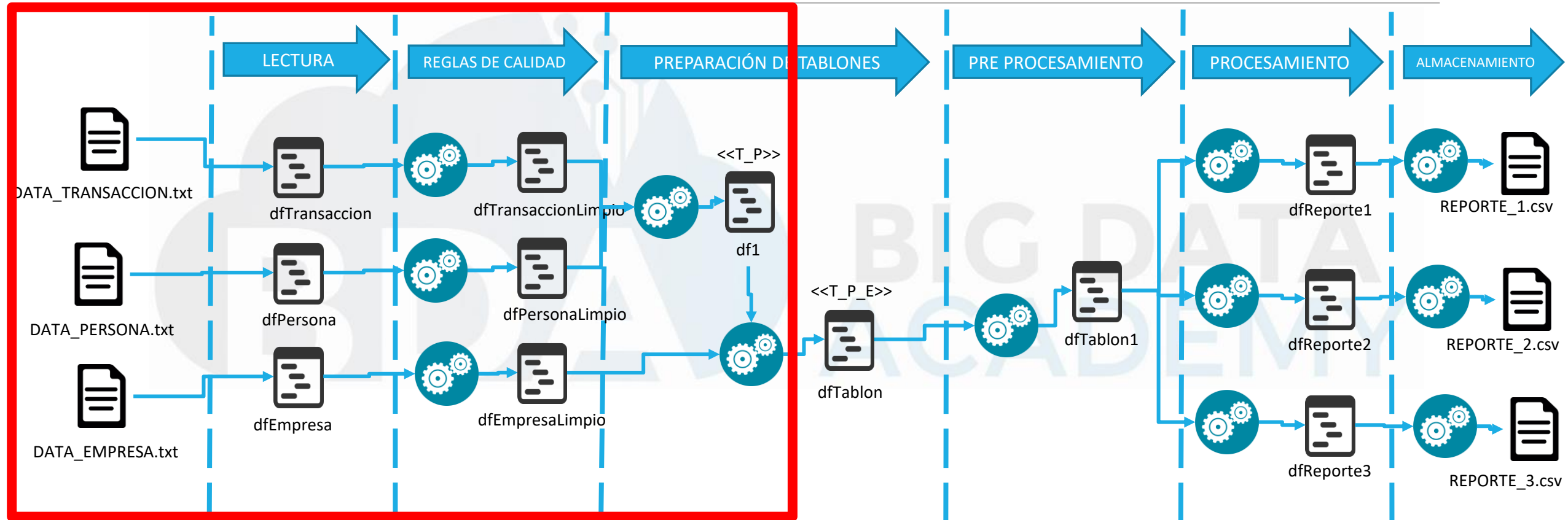
---

# Patrón de Diseño CHECKPOINT

---

# Proceso a implementar

Mientras más pasos tenga nuestro proceso, más memoria RAM necesitaremos, La memoria RAM se acabará en algún momento. **Supongamos que la RAM se acaba en el "df1"**



# Desbordamiento de memoria RAM

Si tratamos de crear un nuevo dataframe, obtendremos el siguiente error:

```
at org.apache.spark.sql.Dataset.reduce(Dataset.scala:1454)
... 48 elided
Caused by: java.lang.OutOfMemoryError: Java heap space
scala> 20/06/09 08:28:58 ERROR TaskContextImpl: Error in TaskComple
java.lang.IllegalStateException: Block broadcast_24 not found
at org.apache.spark.storage.BlockInfoManager$$anonfun$2.ap
```

Este error indica **“se acabó la memoria RAM, no puedes crear más dataframes”**

¿Cómo lo solucionamos?

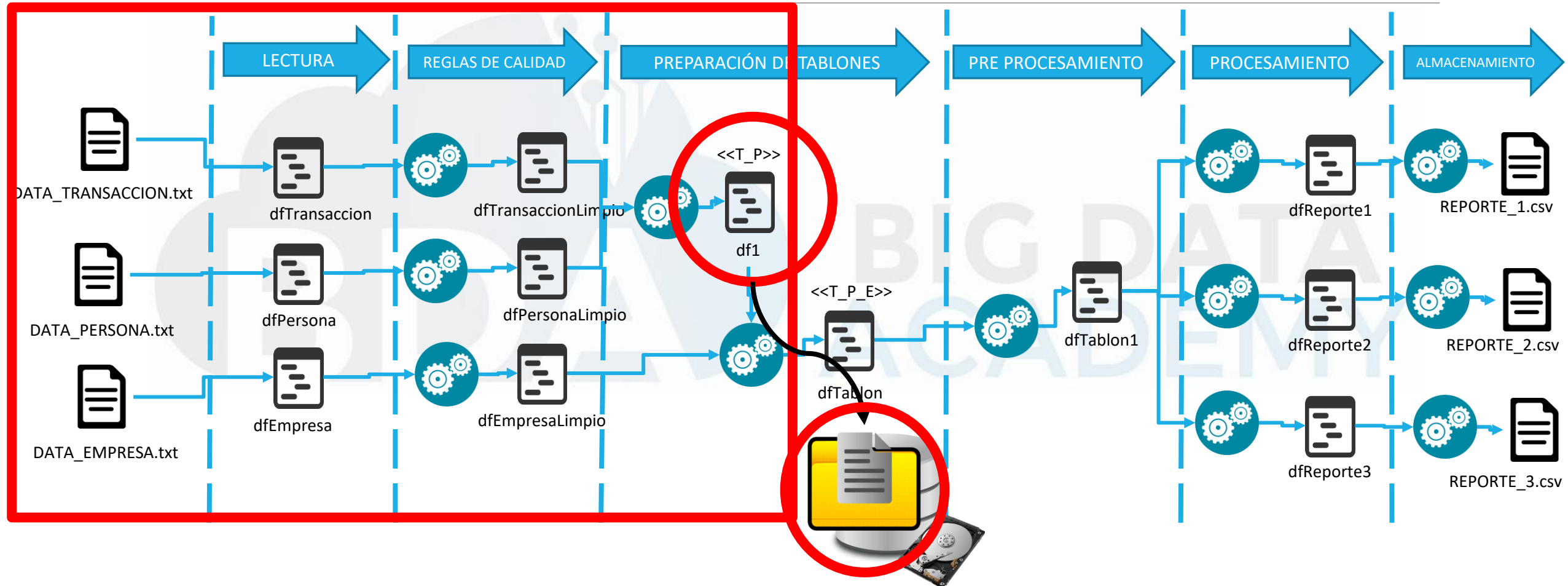
---

# Patrón de Diseño CHECKPOINT

---

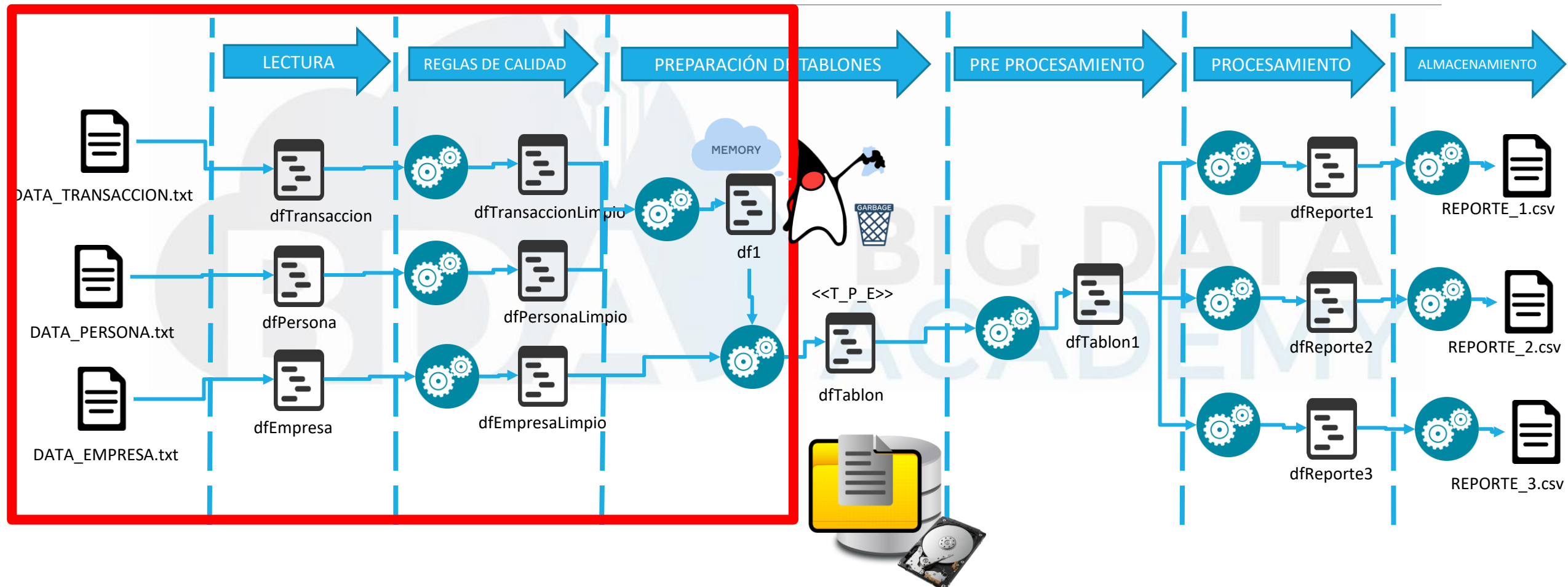
# Almacenamiento del “df1” en un archivo

Almacenamos al dataframe “df1” en un archivo en el disco duro (a esto se le conoce como **crear un “checkpoint”**)



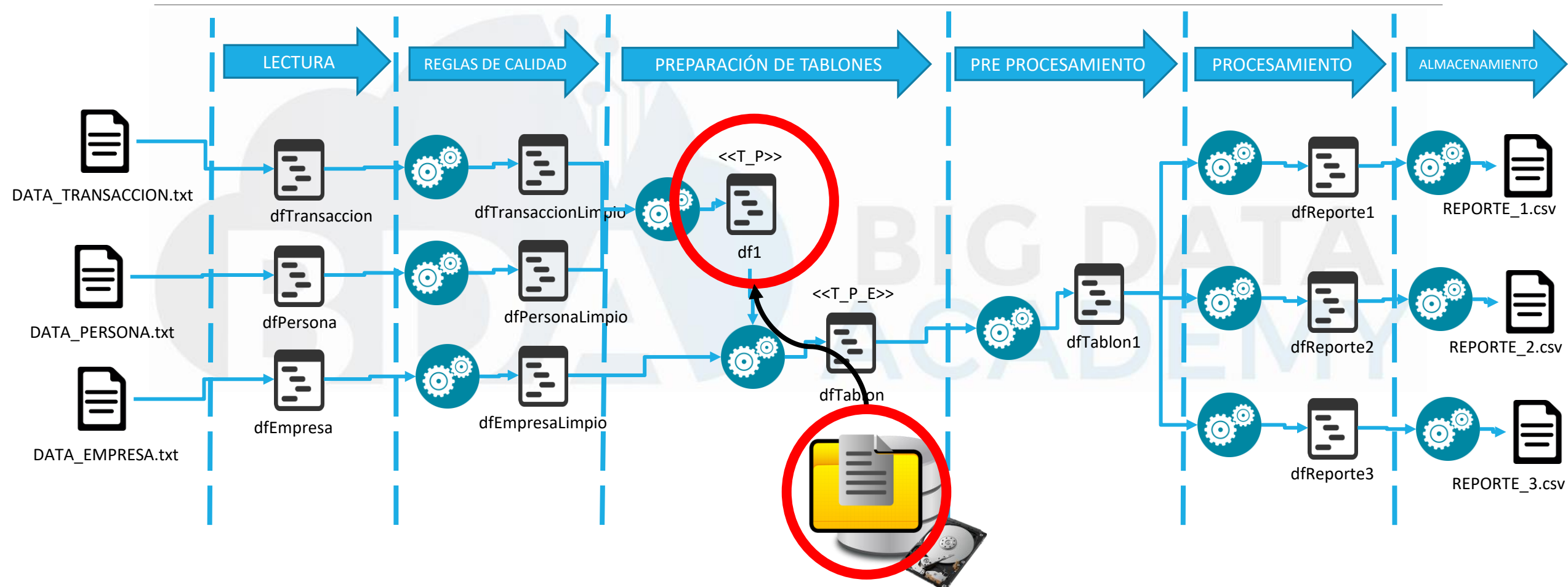
# Llamada al Garbage Collector

**Llamamos al Garbage Collector para eliminar todos los dataframes**, incluyendo el “df1”. Cuando se fuerza el llamado del Garbage Collector, se eliminan todos los dataframes, incluyendo aquellos que estén en la caché.



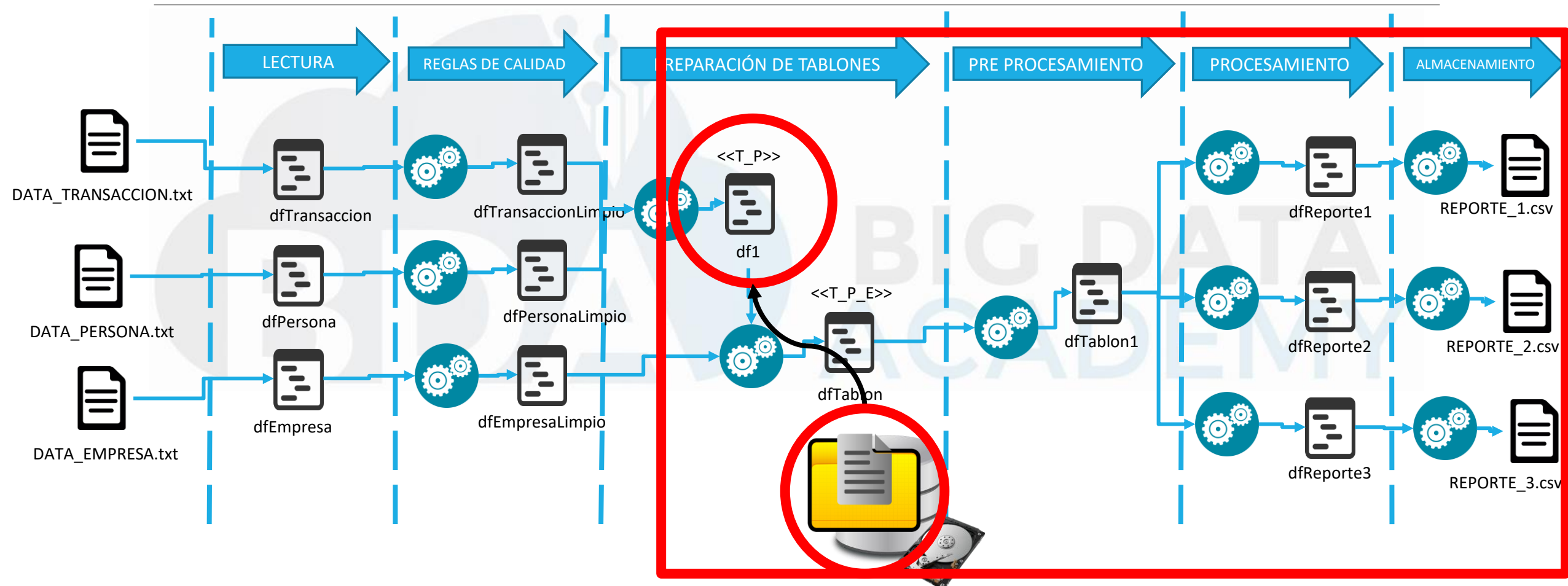
# Recuperación del dataframe

La memoria RAM se encuentra vacía nuevamente, recuperamos desde el archivo al dataframe "df1"



# El proceso continúa

**Como la memoria RAM se vació, el proceso puede seguir ejecutándose**





# ¿Y si nuevamente se acaba la RAM?

Si nuevamente se acaba la RAM:

```
at org.apache.spark.sql.Dataset.reduce(Dataset.scala:1454)
... 48 elided
Caused by: java.lang.OutOfMemoryError: Java heap space
scala> 20/06/09 08:28:58 ERROR TaskContextImpl: Error in TaskComple
java.lang.IllegalStateException: Block broadcast_24 not found
at org.apache.spark.storage.BlockInfoManager$$anonfun$2.ap
```

**Volvemos a aplicar el patrón de diseño CHECKPOINT** para liberar nuevamente la memoria RAM