**RECUPERACIONES DE CARTERA CASTIGADA**

El objetivo del proceso de recuperación de cartera castigada es:

- Crear una BD para clasificar el saldo de cartera castigada según su:

\* Segmento

\* Producto

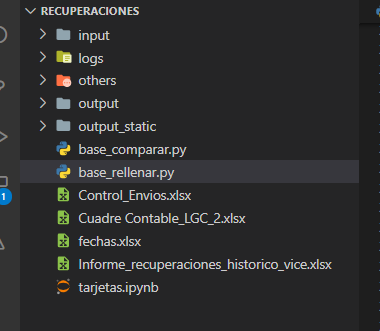
\* Banca

\* Vicepresidencia

- Lo anterior es para tener una cifra más detallada en los informes.

**EJECUCION DEL PROCESO**

El proceso se maneja con la siguiente estructura:



El orden en que se ejecuta el proceso es el siguiente:

**1) Carpeta “input” (datos de entrada):**

Contiene “insumos” que son archivos de Excel de diferentes áreas para ejecutar el proceso.

**2) Python “base\_comparar.py” (transformar datos):**

Este .py ya está bueno, NO hay q modificarlo.

Lee los archivos de Excel q están en “input”. Este .py hace lo siguiente con los archivos:

- Agrupa (junta) los archivos.

- Parametriza el nombre de los campos (arreglar los datos cambiando nombre de las columnas, quitar las tildes, arreglar letras mayúsculas y minúsculas, etc.)

- Castea las obligaciones y los ID a 17 y 15 dígitos respectivamente. CAST() en SQL sirve para convertir de un tipo de dato a otro tipo de dato

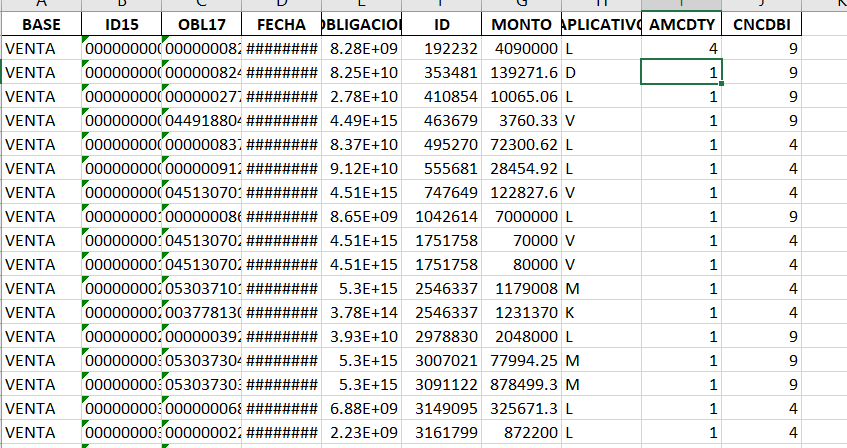
- Crea una carpeta llamada “output”

base\_consolidada\_anio\_mes.xlsx es el resultado de base\_comparar.py

**3) Carpeta “output” (datos de salida):**

Aquí hay un Excel q fue creado por base\_comparar.py. Este Excel se llama “base\_consolidada\_anio\_mes.xlsx”. El Excel es el siguiente:

proceso\_generadores.base\_consolidada\_anio\_mes



**Nombre Columnas (Campos) base\_consolidada\_mes-año.xlsx:**

**\* BASE:** Indicador para saber de qué insumo (Excel de carpeta “input”) es la información que se ve en la base agrupada

**\* ID15:** Numero de identificación (cedula, tarjeta de identidad, etc.) de 15 dígitos

\* **OBL17:** Numero de obligación de 17 dígitos

\* **FECHA:** Dato estático (no cambia) que tiene la fecha de ejecución del proceso

\* **OBLIGACION:** Numero de obligación, una obligación es una deuda

\* **ID:** Numero de ID

\* **MONTO:** Dinero recuperado (ganancia), cabe aclarar que es negativo, los valores positivos son ajustes realizados a ventas anteriores

\* **APLICATIVO:** Código del aplicativo, es una letra.

\* **AMCDTY:** Código de la modalidad, es un número.

\* **CNCDBI:** Código del segmento, es un número.

\* Las columnas **APLICATIVO**, **AMCDTY** y **CNCDBI** solamente están en la **BASE** (base de datos) **BANCO** y **VENTA**

**4) Python “base\_rellenar.py”:**

Este es el .py que hay que pasar a .sql

Cruza las tablas (Excel) y ejecuta otros requerimientos (reglas de negocio) para llenar los campos faltantes (NULL)

**SIGNIFICADOS DE LAS TABLAS**

**NombreBaseDeDatos.NombreTabla**

**Tablas NO temporales:** Son las tablas q me paso Gian, están en la zona (BD) proceso... son el resultado definitivo de los datos

**1) proceso.recuperaciones\_capa**

Tabla q se crea en consulta 1, se crea usando la tabla (hace cruce con) **resultados\_riesgos.master\_credit\_risk**

y con

**proceso\_generadores.base\_consolidada\_anio\_mes**

Cuando seleccionas **proceso\_generadores.base\_consolidada\_anio\_mes** se descarga el Excel **base\_consolidada\_2022\_enero.xlsx** (son lo mismo, son una copia)

**base\_consolidada\_anio\_mes.xlsx** es el resultado de **base\_comparar.py**

**2) proceso.recuperaciones\_cenie**

Tabla q se crea en consulta 2, busca las casillas NULL de **proceso.recuperaciones\_capa** que coinciden con **resultados\_riesgos.ceniegarc\_lz**

**3) proceso.informe\_recu\_2020102\_v2**

Tabla q se crea en consulta 3,

une **proceso.recuperaciones\_capa** y **proceso.recuperaciones\_cenie**

Es la tabla que se genera al final de las consultas 1, 2 y 3

**4) proceso\_generadores.hipotecario\_202202**

Es un Excel que esta en la carpeta “input” (datos de entrada)

**Tablas Temporales:** Están en la zona (BD) proceso\_generadores y se borran automáticamente, son para probar mi código, son copias de las tablas anteriores. Tienen el mismo **NombreTabla**, lo único que hay que cambiar es el **NombreBaseDeDatos** por proceso\_generadores:

**1) proceso\_generadores.recuperaciones\_capa**

**1.2) resultados\_riesgos.master\_credit\_risk**

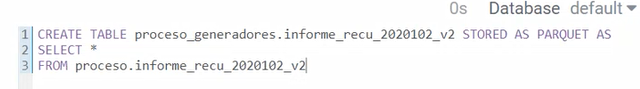
**1.3) proceso\_generadores.base\_consolidada\_anio\_mes**

**2) proceso\_generadores.recuperaciones\_cenie**

**3)** **proceso\_generadores.informe\_recu\_2020102\_v2**

**4) proceso\_generadores.hipotecario\_202202**

**CONSULTA PARA CREAR COPIAS DE TABLAS PARA PROBAR MI CODIGO .sql**



**TAREA PRACTICANTE**

Convertir el Python base\_rellenar.py a SQL para ejecutarlo en la base de datos de Bancolombia (LZ-HUE).

En el .sql primero copias y pegas este código de a continuación y después ahí si empiezas a escribir tu código:

**CONSULTA 1**

Crear tabla proceso.recuperaciones\_capa =

resultados\_riesgos.master\_credit\_risk + proceso\_generadores.base\_consolidada\_anio\_mes (base\_consolidada\_2022\_enero.xlsx / base\_comparar.py)

|  |
| --- |
| CREATE TABLE PROCESO.RECUPERACIONES\_CAPA STORED AS PARQUET AS  WITH  CAPA AS (  SELECT \* FROM (SELECT  ROW\_NUMBER() OVER(PARTITION BY oblig,num\_doc  ORDER BY (ingestion\_year\*100+ingestion\_month) DESC) AS numero,  num\_doc,  SUBSTR(oblig, 3, LENGTH(oblig)-1)AS oblig,  producto\_agrupado AS PRODUCTO\_AGRUPADO,  banca AS banca,  COD\_CLASE AS MODalidad,  desc\_segm AS desc\_segm,  vic\_ccial AS vic\_ccial  FROM resultados\_riesgos.master\_credit\_risk  BASE  WHERE numero=1)  SELECT X.BASE,  """+anio+""" AS ANHO,  """+mes+""" AS MES,  X.ID,  X.OBL17,  V.PRODUCTO\_AGRUPADO AS PRODUCTO\_AJUSTADO,  V.BANCA AS BANCA\_AJUSTADA,  v.vic\_ccial,  V.MODALIDAD,  V.DESC\_SEGM AS SGTO\_AJUSTADO,  X.MONTO,  X.aplicativo,  X.amcdty,  X.cncdbi  FROM """+tabla+""" X  LEFT JOIN CAPA V ON CAST(X.OBL17 AS DOUBLE)=CAST(V.oblig AS DOUBLE)  AND CAST(X.ID AS BIGINT)=CAST(V.num\_doc AS BIGINT)  ; |

**CONSULTA 2**

Crear tabla proceso.recuperaciones\_cenie =

proceso.recuperaciones\_capa + resultados\_riesgos.ceniegarc\_lz

|  |
| --- |
| *CREATE TABLE PROCESO.RECUPERACIONES\_CENIE STORED AS PARQUET AS*  *WITH*  *CENIE AS (*  *SELECT \* FROM (SELECT ROW\_NUMBER() OVER(PARTITION BY obl341,id*  *ORDER BY (ingestion\_year\*100+ingestion\_month) DESC) AS numero,*  *id,*  *obl341,*  *pcons AS pcons,*  *banca AS banca,*  *vic\_ccial AS vic\_ccial,*  *clf AS modalidad,*  *segdesc AS segdesc*  *FROM resultados\_riesgos.ceniegarc\_lz) BASE WHERE numero=1)*  *SELECT x.base,*  *"""+anio+""" AS anho,*  *"""+mes+""" AS mes,*  *x.id,*  *x.obl17,*  *v.pcons AS producto\_ajustado,*  *v.banca AS banca\_ajustada,*  *v.vic\_ccial AS vic\_ccial,*  *v.modalidad,*  *v.segdesc AS sgto\_ajustado,*  *x.monto,*  *X.aplicativo,*  *X.amcdty,*  *X.cncdbi*  *FROM PROCESO.RECUPERACIONES\_CAPA X*  *LEFT JOIN CENIE V ON CAST(X.OBL17 AS DOUBLE)=CAST(V.OBL341 AS DOUBLE)*  *AND CAST(X.ID AS DOUBLE)=CAST(V.ID AS DOUBLE)*  *WHERE X.PRODUCTO\_AJUSTADO IS NULL*  *;* |

**CONSULTA 3**

Crear tabla proceso.informe\_recu\_2020102\_v2 =

proceso.recuperaciones\_capa + proceso.recuperaciones\_cenie

Consulta que parametriza los campos

|  |
| --- |
| WITH  BASE AS  (SELECT \*  FROM PROCESO.RECUPERACIONES\_CAPA  WHERE PRODUCTO\_AJUSTADO IS NOT NULL    UNION ALL  SELECT \*  FROM PROCESO.RECUPERACIONES\_CENIE  )    SELECT base,  anho,  mes,  id,  lpad(cast(obl17 AS string),  17,  '0') AS obl17,  CASE  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '37781%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '40998%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '4491%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '45130%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '53037%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '5491%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '53069%' THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IS NULL  AND CAST(obl17 AS STRING) LIKE '10000%' THEN 'LEASING'  WHEN producto\_ajustado IN ('Cartera Consumo',  'Vehículo',  'LIBRE INVERSION')  AND modalidad = '2' THEN 'LIBRE INVERSION'  WHEN producto\_ajustado IN ('Cartera Comercial',  'Vehículo',  'CARTERA ORDINARIA')  AND modalidad = '1' THEN 'CARTERA ORDINARIA'  WHEN producto\_ajustado IN ('Credipago virtual') THEN 'CREDIPAGO'  WHEN producto\_ajustado IN ('Ex-Empleado Libranza',  'Libranza',  'LIBRANZA')  AND modalidad = '2' THEN 'LIBRANZA'  WHEN producto\_ajustado IN ('Tarjeta de Crédito',  'TARJETA DE CREDITO') THEN 'TARJETA DE CREDITO'  WHEN producto\_ajustado IN ('CCT SUFI',  'CPE CORTO PLAZO SUFI',  'CPE LARGO PLAZO SUFI',  'MOTOS GAMA BAJA SUFI',  'COTIDIANIDAD') THEN 'COTIDIANIDAD'  WHEN producto\_ajustado IN ('MOVILIDAD',  'VEHÍCULOS SUFI') THEN 'MOVILIDAD'  ELSE upper(producto\_ajustado)  END AS producto\_ajustado,  CASE  WHEN banca\_ajustada = 'Personas y Pyme' THEN 'NEPYP'  WHEN banca\_ajustada = 'PPyE' THEN 'NEPYP'  WHEN banca\_ajustada = 'Empresarial' THEN 'NEPYP'  WHEN banca\_ajustada IN ('BEG',  'VEG')  AND sgto\_ajustado = 'EMPRESARIAL' THEN 'NEPYP'  WHEN banca\_ajustada IN ('BEG',  'VEG')  AND sgto\_ajustado != 'EMPRESARIAL' THEN 'NCORP'  ELSE banca\_ajustada  END AS banca\_ajustada,  CASE  WHEN vic\_ccial IS NULL  AND banca\_ajustada = 'Personas y Pyme'  AND sgto\_ajustado NOT IN ('GOBIERNO DE RED',  'PYMES',  'NEGOCIOS E INDEPEND',  'MICROFINANZAS',  'MI NEGOCIO',  'MICROPYME') THEN 'PERSONAS'  WHEN vic\_ccial IS NULL  AND banca\_ajustada = 'Personas y Pyme'  AND sgto\_ajustado NOT IN ('GOBIERNO DE RED',  'PYMES',  'NEGOCIOS E INDEPEND',  'MICROFINANZAS',  'MI NEGOCIO',  'MICROPYMES') THEN 'PYMES'  ELSE upper(vic\_ccial)  END AS vic\_ccial,  modalidad,  CASE  WHEN sgto\_ajustado IN ('NEGOCIOS E INDEPEND',  'MICROFINANZAS',  'MI NEGOCIO',  'MICROPYME') THEN 'NEGOCIOS & INDEPEND'  WHEN sgto\_ajustado IN ('PLUS',  'PERSONAL PLUS') THEN 'PLUS'  ELSE UPPER(sgto\_Ajustado)  END AS sgto\_ajustado,  monto,  aplicativo,  amcdty,  cncdbi  FROM base  ; |

**NOTA: A PARTIR DE ESTE PASO SE ACABAN LOS CRUCES CON TABLAS, EL RESTO DE INFORMACION SON REGLAS DE NEGOCIO, HAY QUE TENER EN CUENTA PARA VALIDAR QUE LAS REGLAS QUEDEN BIEN PROGRAMADAS Y QUE NO SE APLIQUEN LAS REGLAS PARA LOS CAMPOS QUE HAYAN CRUZADO EN LAS TABLAS, ES DECIR, LOS SIGUIENTES PASOS SOLO APLICAN PARA LOS REGISTROS QUE HASTA ESTE PASO HAYAN LLEGADO COMO NULOS O VACIOS**

**PUNTO 1**

**Validación de Nulos de la Base de proceso\_generadores.hipotecario\_202202**

Hacer LEFT JOIN (cruce) por ID de

proceso\_generadores.informe\_recu\_2020102\_v2 con proceso\_generadores.hipotecario\_202202

El LEFT JOIN se hace por:

t1 = proceso\_generadores.informe\_recu\_2020102\_v2 (id)

**=**

t2 = proceso\_generadores.hipotecario\_202202 (numero\_de\_indetificacion\_del\_cliente)

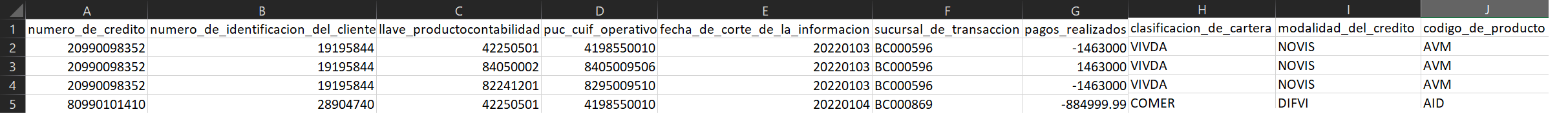
Del resultado entre estas 2 tablas anteriores es que voy a llenar las casillas NULL si se cumplen ciertas condiciones

Las tablas que necesito son:

t1 = proceso\_generadores.informe\_recu\_2020102\_v2



t2 = proceso\_generadores.hipotecario\_202202



La tabla resultante debe tener los siguientes nombres de columnas:

- Todas las columnas de proceso\_generadores.informe\_recu\_2020102\_v2

\* id

\* modalidad

\* producto\_ajustado

\* sgto\_ajustado

\* vic\_ccial

\* banca\_ajustada

\* base

\* anho

\* mes

\* obl17

\* monto

\* aplicativo

\* amcdty

\* cncdbi

- De proceso\_generadores.hipotecario\_202202 solamente contiene las columnas:

\* numero\_de\_indetificacion\_del\_cliente

\* clasificacion\_de\_cartera

Se tiene que validar lo siguiente:

-Si la columna de clasificacion\_de\_cartera es “COMER”, la modalidad es 1, si no, la modalidad es 3

**modalidad**

|  |
| --- |
| SI (clasificacion\_de\_cartera = 'COMER'  AND clasificación\_de\_cartera IS NOT NULL  AND modalidad IS NULL  AND base = 'HIPOTECARIO')  modalidad = '1'  SI NO (clasificación\_de\_cartera <> 'COMER'  AND clasificación\_de\_cartera IS NOT NULL  AND modalidad IS NULL  AND base = 'HIPOTECARIO')  modalidad = '3' |

-Si la modalidad es 1 entonces:

producto\_ajustado=OTROS HIPOTECARIO

sgto\_ajustado=NEGOCIOS & INDEPEND

vic\_ccial = PYMES

-Si la modalidad es otra diferente a 1:

producto\_ajustado=HIPOTECARIO VIVIENDA

sgto\_ajustado=PERSONAL

vic\_ccial = PERSONAS

En ambos casos la banca\_ajustada es NEPYP

**producto\_ajustado**

|  |
| --- |
| SI (modalidad = '1'  AND base = 'HIPOTECARIO'  AND producto\_ajustado IS NULL)  producto\_ajustado='OTROS HIPOTECARIO'  SI NO (modalidad <> '1'  AND base = 'HIPOTECARIO'  AND producto\_ajustado IS NULL)  producto\_ajustado='HIPOTECARIO VIVIENDA' |

**sgto\_ajustado**

|  |
| --- |
| SI (modalidad = '1'  AND base = 'HIPOTECARIO'  AND sgto\_ajustado IS NULL)  sgto\_ajustado='NEGOCIOS & INDEPEND'  SI NO (modalidad <> '1'  AND base = 'HIPOTECARIO'  AND sgto\_ajustado IS NULL)  sgto\_ajustado='PERSONAL' |

**vic\_ccial**

|  |
| --- |
| SI (modalidad = '1'  AND base = 'HIPOTECARIO'  AND vic\_ccial IS NULL)  vic\_ccial = 'PYMES'  SI NO (modalidad <> '1'  AND base = 'HIPOTECARIO'  AND vic\_ccial IS NULL)  vic\_ccial = 'PERSONAS' |

**banca\_ajustada**

|  |
| --- |
| SI (base = 'HIPOTECARIO'  AND banca\_ajustada IS NULL)  banca\_ajustada = 'NEPYP' |

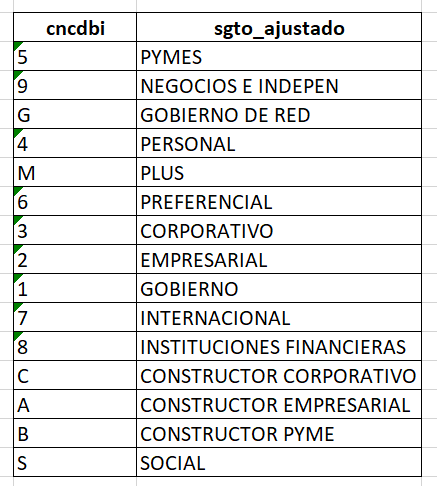
**PUNTO 2**

**2) Validación de NULL de la base de banco y venta (es la misma norma para ambos casos)**

Antes de empezar se debe tener preparada una tabla con:

- código del segmento

- producto completo





Trabajar con la misma tabla del punto 1

en TODOS los condicionales TIENE q estar esto:

AND (TRIM(base) IN ('BANCO','VENTA'))

AND (TRIM(nombre\_columna\_q\_estoy\_llenandole\_los\_NULL) IN("",NULL))

Si la columna aplicativo es L ó E se deben invertir los números de la modalidad:

Si el AMCDTY es 1, la modalidad se cambia por 2

Si el AMCDTY es 2, la modalidad se cambia por 1

Si el AMCDTY es 4, la modalidad se mantiene en 4

**PARA EL RESTO DE APLICATIVOS, LA MODALIDAD ES IGUAL AL AMCDTY**

NO hacer NADA cuando:

amcdty IS NULL

aplicativo IS NULL

**modalidad**

|  |
| --- |
| //invertir los números de la modalidad  SI (TRIM(base) IN('BANCO', 'VENTA'))  AND (TRIM(modalidad) IN("",NULL))  AND (TRIM(aplicativo) IN('L','E'))  SI (amcdty = 1)  modalidad = '2'  SINO (amcdty = 2)  modalidad = '1'  SINO (amcdty = 4)  modalidad = '4'  // Para el resto de aplicativos, la modalidad es igual al amcdty  SINO (TRIM(base) IN('BANCO','VENTA'))  AND (TRIM(modalidad) IN("",NULL))  AND (TRIM(aplicativo) NOT IN('L','E'))  SI (amcdty = 1)  modalidad = '1'  SINO (amcdty = 2)  modalidad = '2'  SINO (amcdty = 4)  modalidad = '4' |

CON LA NUEVA MODALIDAD GENERADA EN EL PASO ANTERIOR SE HACE LO SIGUIENTE:

Si modalidad es 2, el producto\_ajustado es LIBRE INVERSION

Si modalidad es 1, el producto\_ajustado es CARTERA ORDINARIA

Si modalidad es 4, el producto\_ajustado es CARTERA MICROCREDITO

Si el aplicativo es K, M ó V el producto\_ajustado es TARJETA DE CREDITO

Si el aplicativo es D, el producto\_ajustado es SOBREGIROS

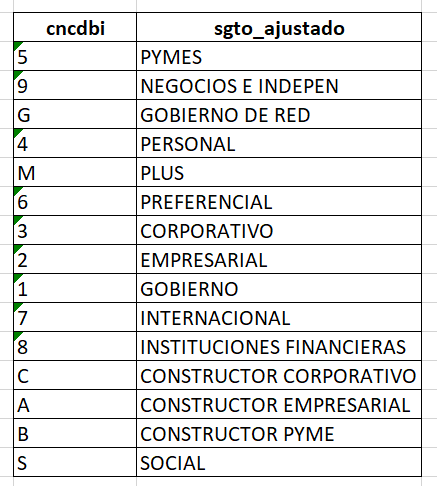
**producto\_ajustado**

|  |
| --- |
| SI (TRIM(base) IN('BANCO','VENTA'))  AND (TRIM(producto\_ajustado) IN("",NULL))  AND (TRIM(aplicativo) IN('L','E'))  SI (TRIM(modalidad) = '2')  producto\_ajustado = 'LIBRE INVERSION'  SINO (TRIM(modalidad) = '1')  producto\_ajustado = 'CARTERA ORDINARIA'  SINO (TRIM(modalidad) = '4')  producto\_ajustado = 'CARTERA MICROCREDITO'  // SINO (NO hacer NADA) |

|  |
| --- |
| SI (TRIM(base) IN('BANCO','VENTA'))  AND (TRIM(producto\_ajustado) IN("",NULL))  SI (TRIM(aplicativo) IN('K','M','V'))  producto\_ajustado = 'TARJETA DE CREDITO'  SINO (TRIM(aplicativo)='D')  producto\_ajustado = 'SOBREGIROS'  // SINO (NO hacer NADA) |

Reemplazar casillas vacías o NULL de la columna llamada sgto\_ajustado

La columna CNCDBI es el código\_del\_segmento, el segmento se consigue cruzando contra la tabla de segmento, nos traemos el SEGM cruzando contra el COD



**sgto\_ajustado**

|  |
| --- |
| SI (TRIM(base) IN('BANCO','VENTA'))  AND (TRIM(sgto\_ajustado) IN("",NULL))  SI (TRIM(cncdbi)='5')  sgto\_ajustado = 'PYMES'  SINO (TRIM(cncdbi)='9')  sgto\_ajustado = 'NEGOCIOS E INDEPEN'  SINO (TRIM(cncdbi)='G')  sgto\_ajustado = 'GOBIERNO DE RED'  SINO (TRIM(cncdbi)='4')  sgto\_ajustado = 'PERSONAL'  SINO (TRIM(cncdbi)='M')  sgto\_ajustado = 'PLUS'  SINO (TRIM(cncdbi)='6')  sgto\_ajustado = 'PREFERENCIAL'  SINO (TRIM(cncdbi)='3')  sgto\_ajustado = 'CORPORATIVO'  SINO (TRIM(cncdbi)='2')  sgto\_ajustado = 'EMPRESARIAL'  SINO (TRIM(cncdbi)='1')  sgto\_ajustado = 'GOBIERNO'  SINO (TRIM(cncdbi)='7')  sgto\_ajustado = 'INTERNACIONAL'  SINO (TRIM(cncdbi)='8')  sgto\_ajustado = 'INSTITUCIONES FINANCIERAS'  SINO (TRIM(cncdbi)='C')  sgto\_ajustado = 'CONSTRUCTOR CORPORATIVO'  SINO (TRIM(cncdbi)='A')  sgto\_ajustado = 'CONSTRUCTOR EMPRESARIAL'  SINO (TRIM(cncdbi)='B')  sgto\_ajustado = 'CONSTRUCTOR PYME'  SINO (TRIM(cncdbi)='S')  sgto\_ajustado = 'SOCIAL'  // SINO (NO hacer NADA) |