Nombre: Angel Caiza

EJERCICIO:

DADA LA SIG BD.

CUENTAS=(NUM\_CUE, TIPO\_CUE, SALDO\_CUE, ID\_CLI, NOM\_CLI, APE\_CLI)

TRANSFERENCIAS=(NUM\_TRA, VALOR\_TRA, FEC\_HOR\_TRA, NUM\_CUE\_ORI, NUM\_CUE\_DES)

SEQ    ,    500   ,   SYSDATE  ,     1001   ,    1002

CREAR UN PROCEDURE TRANSFERIR(CTA\_ORI , CTA\_DES , VALOR)

1001  ,  1002  ,  500

EL PROCEDIMIENTO DEBE GARANTIZAR LAS PROPIEDADES [ACID](https://sistemaseducaciononline.uta.edu.ec/mod/url/view.php?id=121348).

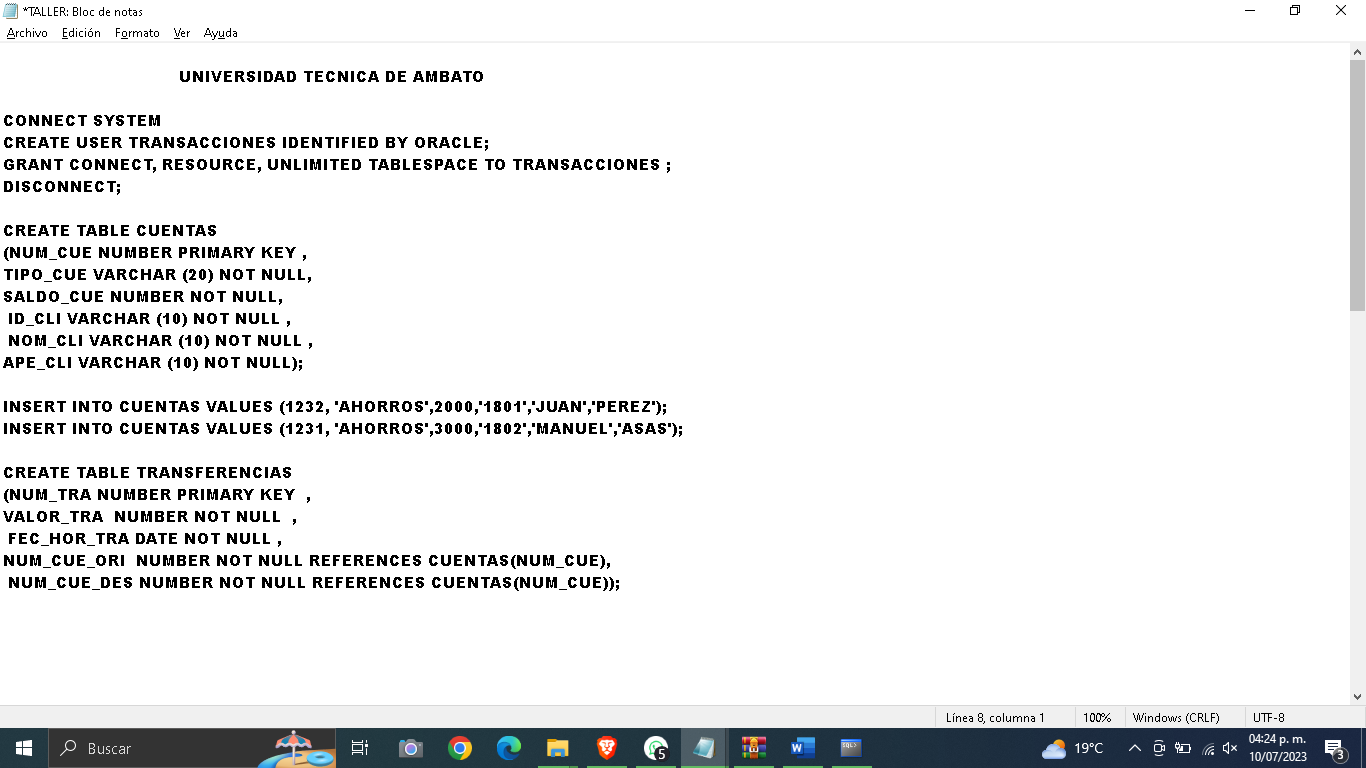
- SELECT SALDO FROM C\_ORIGEN

- UPDATE SALDO C\_ORIGEN (RESTAR 500)

- UPDATE SALDO C\_DESTINO (SUMAR 500)

- INSERT INTO TRANSACCIONES LA TRANSACCION REALIZADA

- COMMIT;





Creación de las tablas .

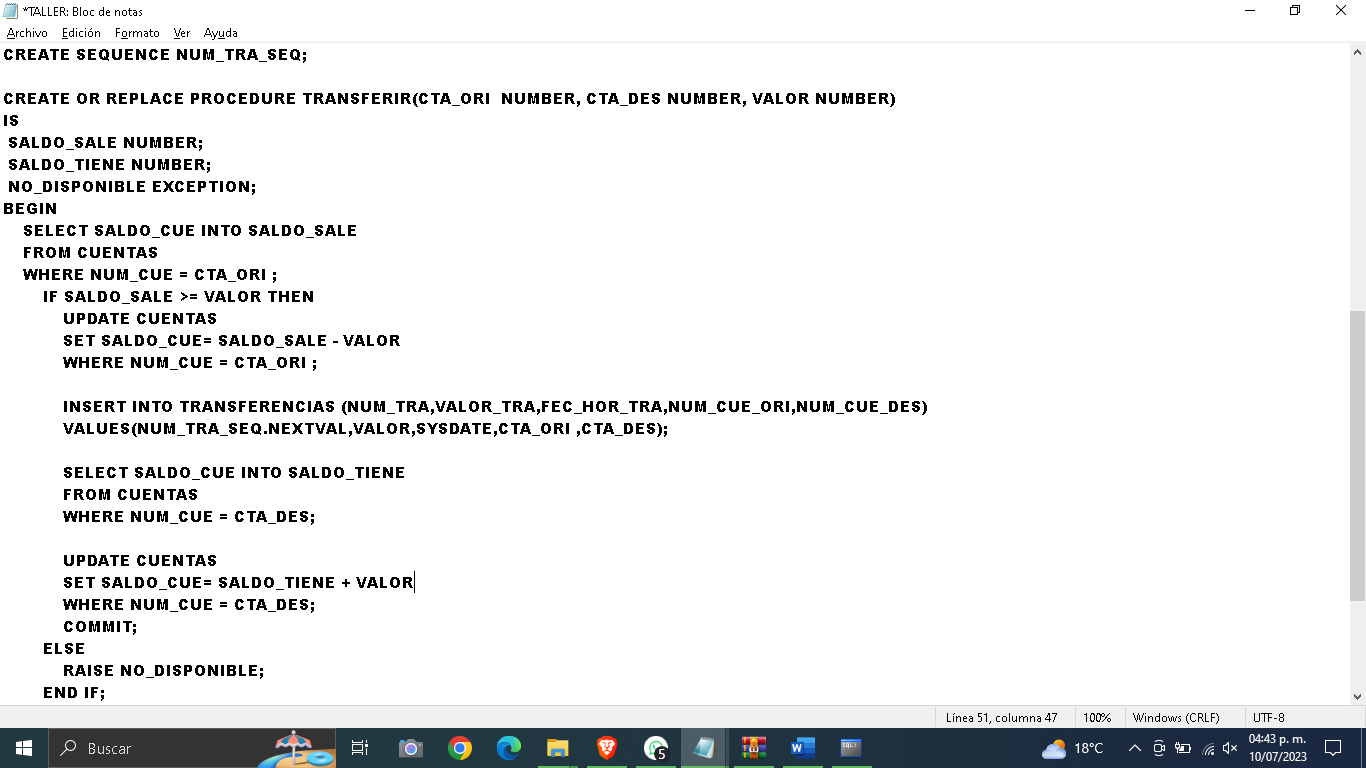
- SELECT SALDO FROM C\_ORIGEN

A-ATOMIC

La propiedad ATOMIC se encuentra presente cuando al ejecutar el procedimiento y verificar el saldo, en caso de que el saldo sea suficiente ejecutará

el programa

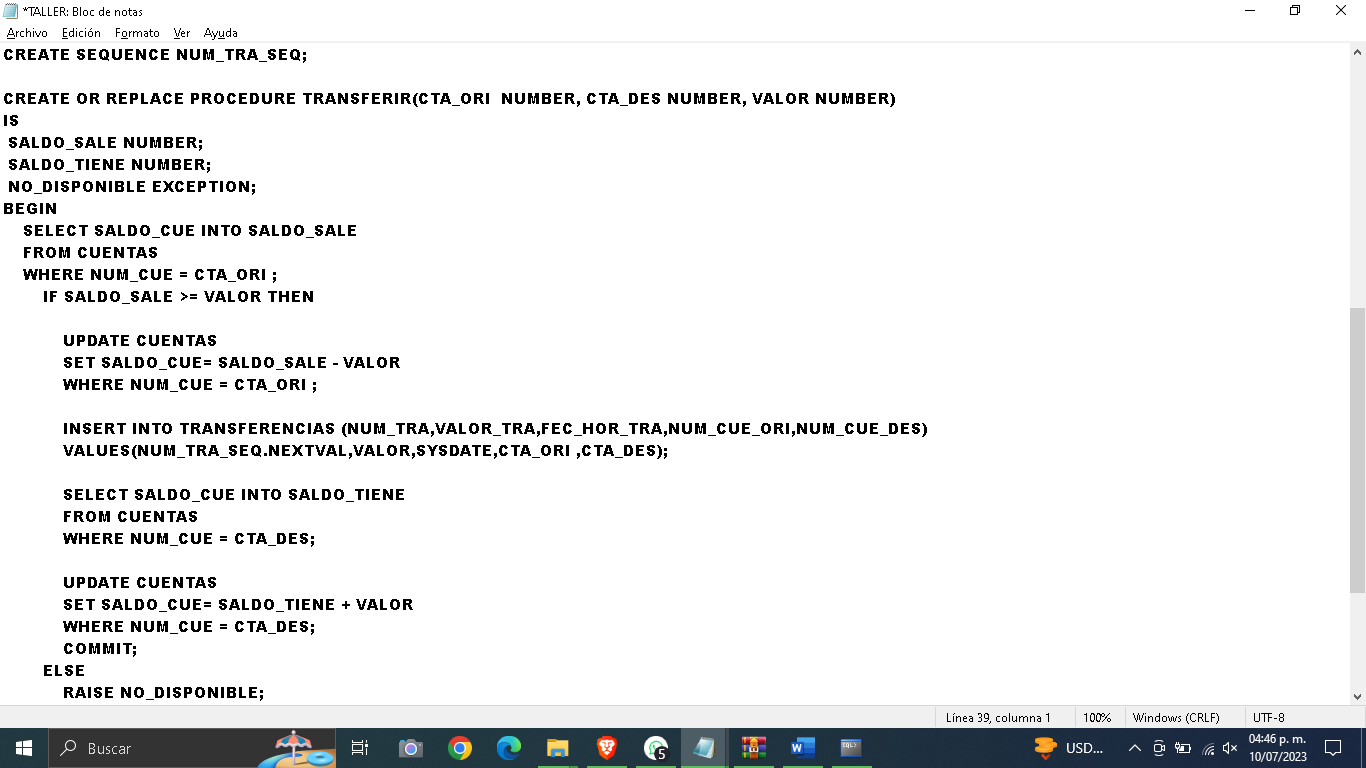
caso contrario simplemente arrojará una excepción y no ejecutará nada.





- UPDATE SALDO C\_ORIGEN (RESTAR 500)





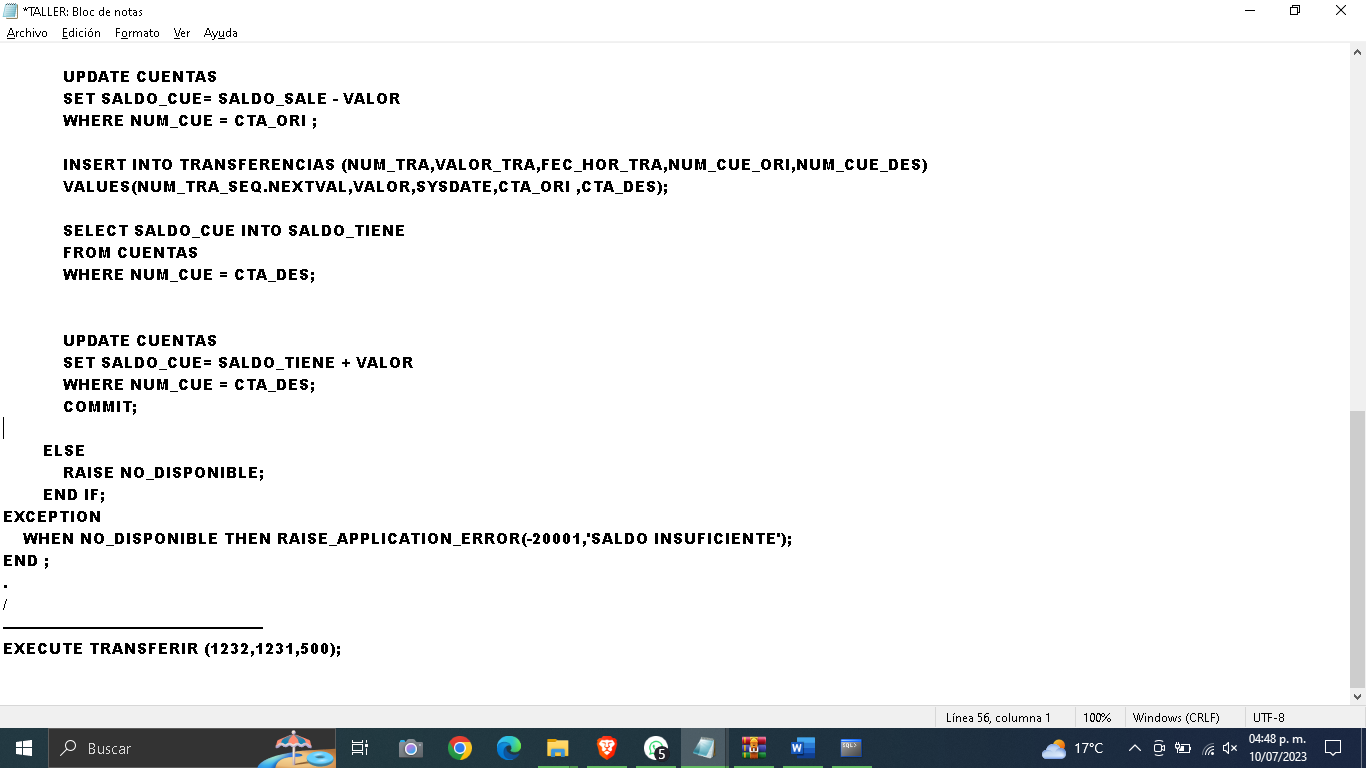


- UPDATE SALDO C\_DESTINO (SUMAR 500)

C-CONSISTENT

La tabla Transferencias tiene dos referencias al los números de cuentas de la tabla Cuentas, por ende, cuando se ejecuta el bloque PLSQL, todas las sentencias

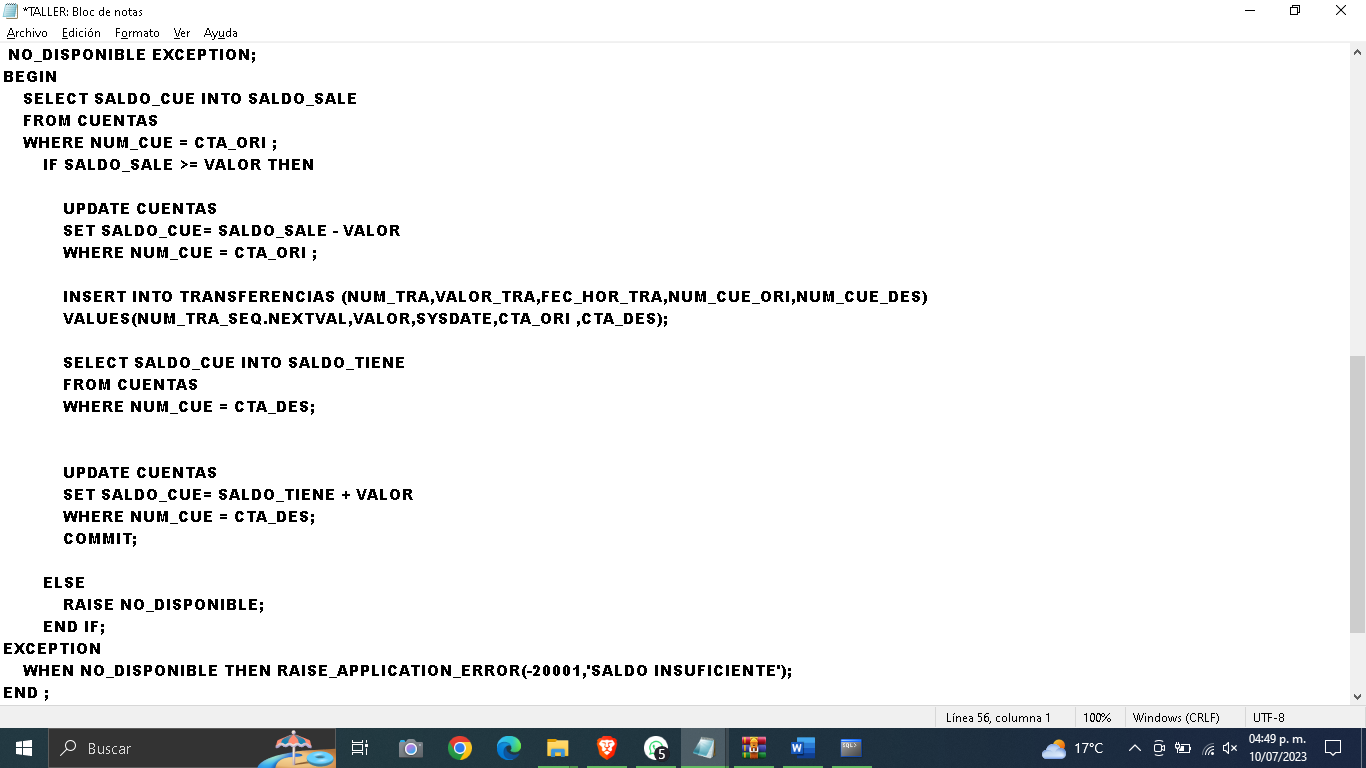
DML verifican que las claves principales enviadas como parámetros existan en la tabla Cuentas, caso contrario, si no existan las claves enviadas no hará nada.





- INSERT INTO TRANSACCIONES LA TRANSACCION REALIZADA

I-ISOLATED

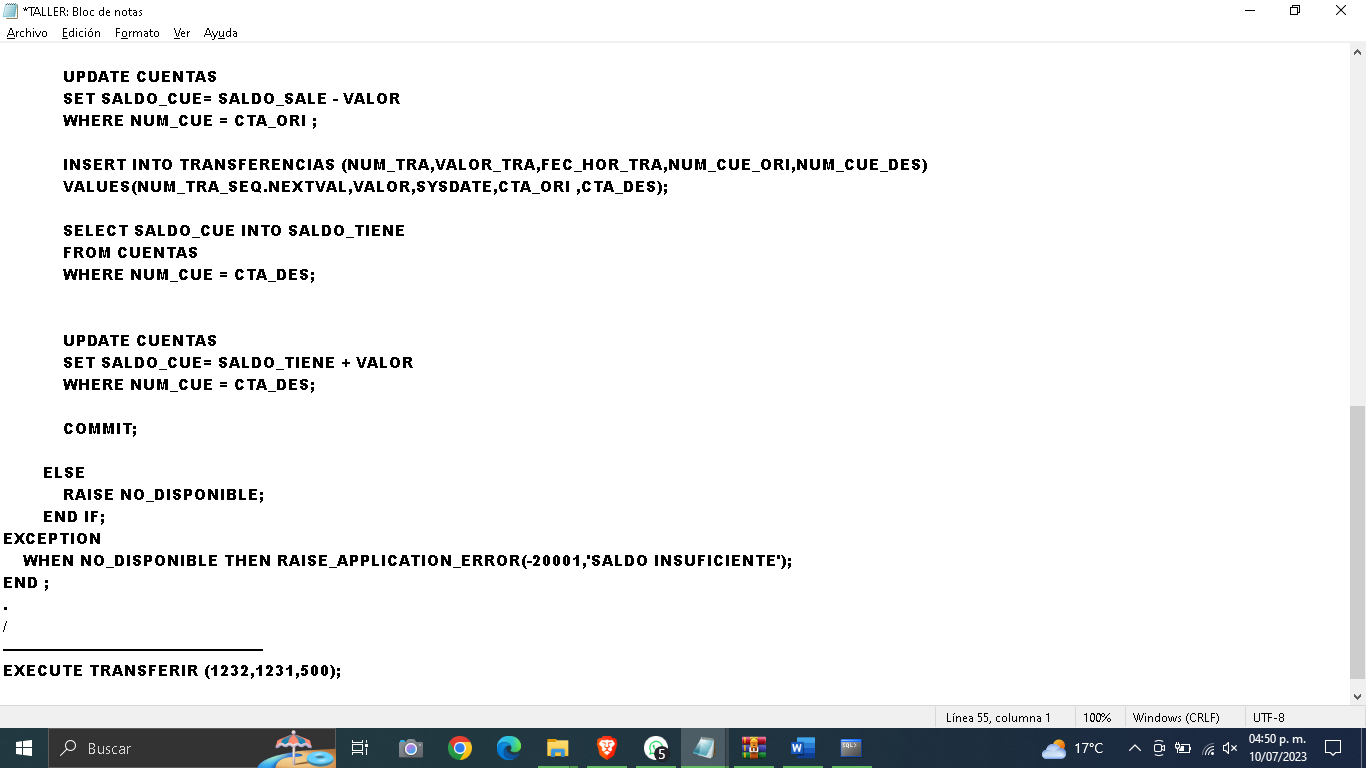
Una vez que se ejecute el bloque PLSQL, los cambios serán visibles inmediatamente, especialmente por la sentencia "COMMIT" que guarda los cambios ejecutados.



- COMMIT;

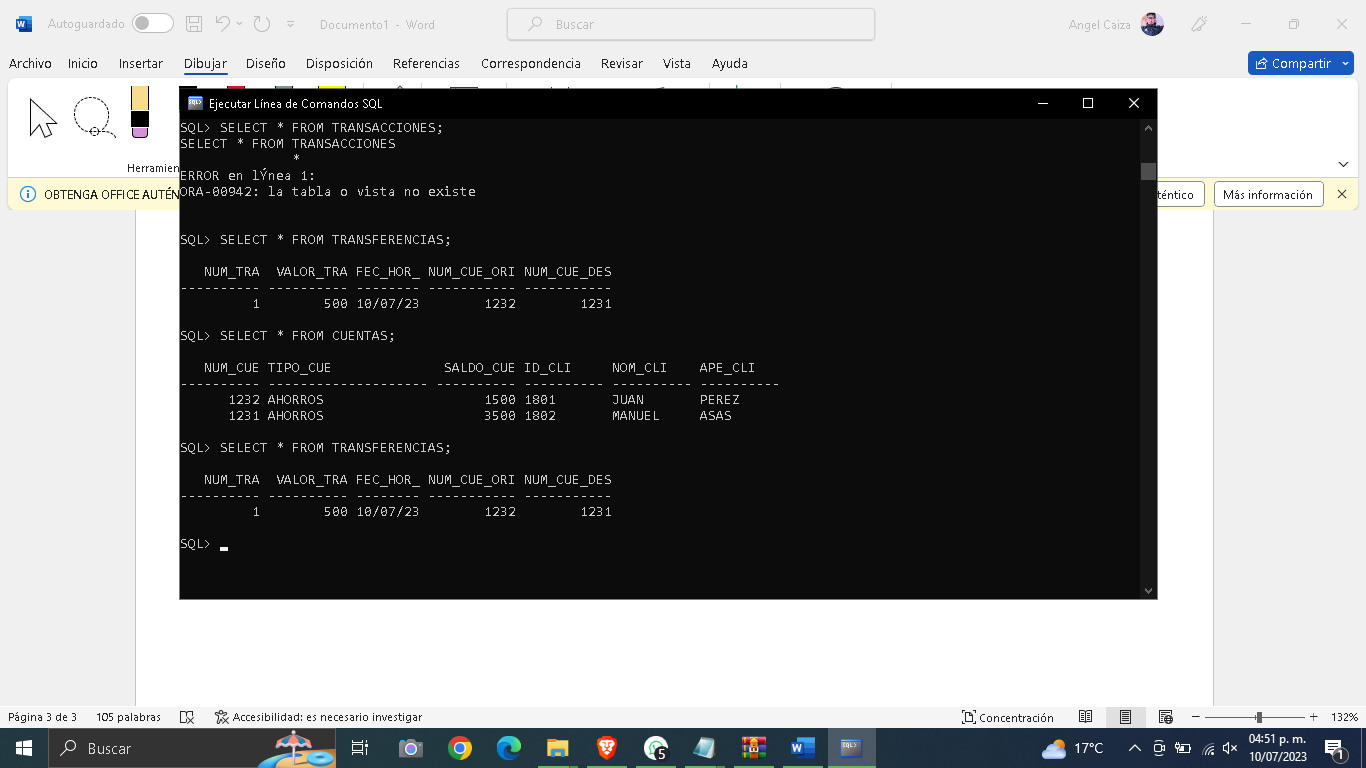
D-DURABILITY

El bloque PLSQL tiene agregado la sentencia "COMMIT" que permite que los cambios realizados se almacenen permanentemente en la base de datos.





**RESULTADOS**



|  |
| --- |
| CREATE TABLE CUENTAS  (NUM\_CUE NUMBER PRIMARY KEY ,  TIPO\_CUE VARCHAR (20) NOT NULL,  SALDO\_CUE NUMBER NOT NULL,  ID\_CLI VARCHAR (10) NOT NULL ,  NOM\_CLI VARCHAR (10) NOT NULL ,  APE\_CLI VARCHAR (10) NOT NULL);  INSERT INTO CUENTAS VALUES (1232, 'AHORROS',2000,'1801','JUAN','PEREZ');  INSERT INTO CUENTAS VALUES (1231, 'AHORROS',3000,'1802','MANUEL','ASAS');  CREATE TABLE TRANSFERENCIAS  (NUM\_TRA NUMBER PRIMARY KEY ,  VALOR\_TRA NUMBER NOT NULL ,  FEC\_HOR\_TRA DATE NOT NULL ,  NUM\_CUE\_ORI NUMBER NOT NULL REFERENCES CUENTAS(NUM\_CUE),  NUM\_CUE\_DES NUMBER NOT NULL REFERENCES CUENTAS(NUM\_CUE));  --------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  CREATE SEQUENCE NUM\_TRA\_SEQ;  CREATE OR REPLACE PROCEDURE TRANSFERIR(CTA\_ORI NUMBER, CTA\_DES NUMBER, VALOR NUMBER)  IS  SALDO\_SALE NUMBER;  SALDO\_TIENE NUMBER;  NO\_DISPONIBLE EXCEPTION;  BEGIN  SELECT SALDO\_CUE INTO SALDO\_SALE  FROM CUENTAS  WHERE NUM\_CUE = CTA\_ORI ;  IF SALDO\_SALE >= VALOR THEN  UPDATE CUENTAS  SET SALDO\_CUE= SALDO\_SALE - VALOR  WHERE NUM\_CUE = CTA\_ORI ;  INSERT INTO TRANSFERENCIAS (NUM\_TRA,VALOR\_TRA,FEC\_HOR\_TRA,NUM\_CUE\_ORI,NUM\_CUE\_DES)  VALUES(NUM\_TRA\_SEQ.NEXTVAL,VALOR,SYSDATE,CTA\_ORI ,CTA\_DES);  SELECT SALDO\_CUE INTO SALDO\_TIENE  FROM CUENTAS  WHERE NUM\_CUE = CTA\_DES;  UPDATE CUENTAS  SET SALDO\_CUE= SALDO\_TIENE + VALOR  WHERE NUM\_CUE = CTA\_DES;  COMMIT;  ELSE  RAISE NO\_DISPONIBLE;  END IF;  EXCEPTION  WHEN NO\_DISPONIBLE THEN RAISE\_APPLICATION\_ERROR(-20001,'SALDO INSUFICIENTE');  END ;  .  /  ----------------------------------------------------  EXECUTE TRANSFERIR (1232,1231,500); |