**TRABAJO PRÁCTICO**

**MODELO DE PUNTO DE VENTA**

**AÑO 2021**

**Integrantes del grupo**

| **Cédula** | **Apellido** | **Nombre** | **Sección** |
| --- | --- | --- | --- |
| 4662525 | Medina Recalde | Oliver Isaías | NA |
| 6129422 | Amarilla Leguizamón | Oscar Alexander | NA |
| 4659580 | Gómez Cárdenas | Eduardo David | NA |

**PARTE 3**

**CURSORES, PROCEDIMIENTOS, FUNCIONES, TIPOS, PAQUETES**

1. Crear el paquete **PCK\_PUNTO\_VENTA** con los siguientes elementos:
   1. El tipo tabla indexada T\_DETALLE compuesto de los siguientes elementos:

* ID\_PRODUCTO
* CANTIDAD
  1. El procedimiento **P\_INSERTAR\_MOVIMIENTO** que recibe como parámetros: cod\_sucursal, fecha\_operacion, cod\_operacion, id\_persona, id\_usuario, descripcion\_operacion, nro\_caja.

El procedimiento deberá insertar un registro en la tabla D\_MOVIMIENTO\_OPERACIONES validando lo siguiente:

* Verificar la fecha\_operacion, la cual debe ser del año actual y anterior o igual a la fecha del sistema. No se admiten fechas adelantadas.
* Verificar el código de operación, si el código de operación es de uso cajero:
  + Debe verificar que el nro de caja sea not null y corresponda a una caja existente.
  + Obtener el número de timbrado de la caja, a partir de dicho número, acceder a la tabla D\_TIMBRADO para obtener el número actual de factura
  + Verificar que el timbrado esté vigente con respecto a la fecha\_operacion introducida, y que el numero\_actual\_factura sea inferior al campo hasta\_nro\_factura. Si alguna de estas condiciones no se cumple, deberá abortar la operación lanzando un error personalizado.
  + Si el numero\_actual\_factura pasó la validación anterior, asigna con dicho valor el campo nro\_comprobante, y también se asigna el nro\_timbrado, Finalmente actualiza la tabla D\_TIMBRADO incrementando el campo numero\_actual\_factura en 1, siempre que dicho incremento no supere el campo hasta\_numero\_factura.
* Si el código de operación **no** es de uso cajero, entonces:
  + Acceder a la tabla D\_TIPO\_COMPROBANTE\_SECUENCIA correspondiente al tipo de comprobante de la operación, correspondiente al año vigente, y obtener el campo nro\_comprobante\_actual, y asignar al número de comprobante. La columna timbrado queda nulo.
  + Actualizar la tabla D\_TIPO\_COMPROBANTE\_SECUENCIA incrementando el campo nro\_comprobante\_actual en 1.
* Asignar las siguientes columnas:
  + FECHA\_INSERT = con la fecha del sistema
  + TIPO\_REGISTRO = ‘A’
  + DESCRIPCION\_OPERACION, si el parámetro recibido está nulo, asignará la descripción correspondiente al código de operación de la tabla D\_OPERACIONES, de lo contrario asignará el valor del parámetro.
  1. El procedimiento **P\_ACTUALIZAR\_STOCK** que recibe como parámetros id\_producto, cod\_sucursal, cantidad, uso\_stock

El procedimiento debe actualizar la tabla D\_STOCK\_SUCURSAL, que almacena el stock de un producto en una sucursal, en base a los parámetros recibidos.

* + Si el valor del parámetro USO\_STOCK es 2 (sumar), debe aumentar la CANTIDAD\_EXISTENCIA en la tabla D\_STOCK\_SUCURSAL del producto dado por el ID\_PRODUCTO, y en la sucursal determinada por el COD\_SUCURSAL.
  + Si el valor del parámetro USO\_STOCK es 1 (restar), debe disminuir la CANTIDAD\_EXISTENCIA en la tabla D\_STOCK\_SUCURSAL del producto dado por el ID\_PRODUCTO, y en la sucursal determinada por el COD\_SUCURSAL. Ello siempre que la (CANTIDAD\_EXISTENCIA - CANTIDAD\_OPERACION) >= STOCK\_MINIMO. Si no se cumple la condición precedente se aborta la operación lanzando un código de error personalizado creado para el efecto.
  1. El procedimiento **P\_INSERTAR\_DETALLE** que recibe como parámetros: id\_operacion, id\_producto, cod\_medida, cantidad\_operacion.

El procedimiento deberá:

Recuperar la sucursal, el código de movimiento y el correspondiente uso\_stock del mismo, a partir de la cabecera D\_MOVIMIENTO\_OPERACIONES.

Insertar un registro en la tabla D\_DETALLE\_OPERACIONES asignando los siguientes campos:

* precio\_operacion = precio de última compra del producto + la aplicación del porcentaje de beneficio.
* importe\_operacion = precio\_operacion x cantidad\_operacion
* cod\_tipo\_iva se obtiene a partir del producto
* porcentaje\_iva se obtiene a partir del tipo de iva
* importe\_iva = (precio\_operacion x cantidad\_operacion) / divisor\_iva\_incluido
* importe\_descuento e importe\_recargo se asigna 0

Actualizar el stock invocando al procedimiento **P\_ACTUALIZAR\_STOCK**, enviando los parámetros requeridos

* 1. La función **F\_VER\_DETALLE** que recibe como parámetro el ID\_OPERACION y devuelve una variable del tipo T\_DETALLE. La función deberá
  + Verificar que exista la operación correspondiente al ID\_OPERACION introducido. Si no existe, dará error.
  + Si existe, recorrerá los movimientos de detalle y llenará una variable del tipo T\_DETALLE, la que será retornada.
  + Solución

| -- Se crea el paquete  CREATE OR REPLACE PACKAGE PCK\_PUNTO\_VENTA AS  PROCEDURE P\_INSERTAR\_MOVIMIENTO(  cod\_sucursal IN D\_MOVIMIENTO\_OPERACIONES.cod\_sucursal%TYPE,  fecha\_operacion IN D\_MOVIMIENTO\_OPERACIONES.fecha\_operacion%TYPE,  cod\_operacion IN D\_MOVIMIENTO\_OPERACIONES.cod\_operacion%TYPE,  id\_persona IN D\_MOVIMIENTO\_OPERACIONES.id\_persona%TYPE,  id\_usuario IN D\_MOVIMIENTO\_OPERACIONES.id\_usuario%TYPE,  descripcion\_operacion IN D\_MOVIMIENTO\_OPERACIONES.descripcion\_operacion%TYPE,  nro\_caja IN D\_MOVIMIENTO\_OPERACIONES.NRO\_CAJA%TYPE  );  PROCEDURE P\_ACTUALIZAR\_STOCK(  v\_id\_producto IN D\_STOCK\_SUCURSAL.ID\_PRODUCTO%TYPE,  v\_cod\_sucursal IN D\_STOCK\_SUCURSAL.COD\_SUCURSAL%TYPE,  v\_cantidad IN D\_STOCK\_SUCURSAL.CANTIDAD\_EXISTENCIA%TYPE,  v\_uso\_stock IN D\_OPERACIONES.USO\_STOCK%TYPE  );  /\*  a) El tipo tabla indexada T\_DETALLE compuesto de los siguientes elementos:  ID\_PRODUCTO  CANTIDAD  \*/  TYPE r\_articulo IS RECORD (  ID\_PRODUCTO D\_DETALLE\_OPERACIONES.ID\_PRODUCTO%TYPE,  CANTIDAD D\_DETALLE\_OPERACIONES.CANTIDAD\_OPERACION%TYPE  );  TYPE t\_detalle IS TABLE OF  r\_articulo INDEX BY BINARY\_INTEGER;  FUNCTION F\_VER\_DETALLE(ID\_OPERACION IN D\_MOVIMIENTO\_OPERACIONES.ID\_OPERACION%TYPE) RETURN T\_DETALLE;  END;  -- AQUI EL CUERPO  CREATE OR REPLACE PACKAGE BODY PCK\_PUNTO\_VENTA  IS  PROCEDURE P\_INSERTAR\_MOVIMIENTO(  cod\_sucursal IN D\_MOVIMIENTO\_OPERACIONES.cod\_sucursal%TYPE,  fecha\_operacion IN D\_MOVIMIENTO\_OPERACIONES.fecha\_operacion%TYPE,  cod\_operacion IN D\_MOVIMIENTO\_OPERACIONES.cod\_operacion%TYPE,  id\_persona IN D\_MOVIMIENTO\_OPERACIONES.id\_persona%TYPE,  id\_usuario IN D\_MOVIMIENTO\_OPERACIONES.id\_usuario%TYPE,  descripcion\_operacion IN D\_MOVIMIENTO\_OPERACIONES.descripcion\_operacion%TYPE,  nro\_caja IN D\_MOVIMIENTO\_OPERACIONES.NRO\_CAJA%TYPE  )  IS  V\_DESC\_OPERACION D\_MOVIMIENTO\_OPERACIONES.DESCRIPCION\_OPERACION%TYPE;  V\_NRO\_COMPROBANTE D\_MOVIMIENTO\_OPERACIONES.NRO\_COMPROBANTE%TYPE;  V\_NRO\_TIMBRADO D\_MOVIMIENTO\_OPERACIONES.NRO\_TIMBRADO%TYPE;  v\_codigo\_operacion D\_OPERACIONES.COD\_OPERACION%TYPE;  v\_numero\_caja D\_CAJAS.NRO\_CAJA%TYPE;  BEGIN  /\*  Verificar la fecha\_operacion, la cual debe ser del año actual y anterior o igual a la  fecha del sistema. No se admiten fechas adelantadas.  \*/  IF fecha\_operacion <= SYSDATE THEN  /\*  Verificar el código de operación, si el código de operación es de uso cajero:  \*/    SELECT OP.cod\_operacion into v\_codigo\_operacion FROM  D\_OPERACIONES OP WHERE USO\_CAJERO = 1 AND OP.cod\_operacion = cod\_operacion fetch first rows only;  IF v\_codigo\_operacion IS NOT NULL THEN  /\*  Debe verificar que el nro de caja sea not null y corresponda a una caja existente.  \*/  SELECT C.NRO\_CAJA into v\_numero\_caja FROM D\_CAJAS C WHERE C.NRO\_CAJA = NRO\_CAJA FETCH FIRST ROWS ONLY;  IF v\_numero\_caja IS NOT NULL THEN  /\*  Obtener el número de timbrado de la caja, a partir de dicho número, acceder a la tabla D\_TIMBRADO  para obtener el número actual de factura.  Verificar que el timbrado esté vigente con respecto a la fecha\_operacion introducida,  y que el numero\_actual\_factura sea inferior al campo hasta\_nro\_factura. Si alguna de estas condiciones  no se cumple, deberá abortar la operación lanzando un error personalizado.  \*/  DECLARE  CURSOR C\_TIMBRADO IS  SELECT C.NRO\_TIMBRADO, T.NUMERO\_ACTUAL\_FACTURA FROM D\_CAJAS C  JOIN D\_TIMBRADO T ON T.NRO\_TIMBRADO = C.NRO\_TIMBRADO  WHERE C.NRO\_CAJA = NRO\_CAJA  AND T.FECHA\_HASTA\_TIMBRADO > fecha\_operacion  AND T.NUMERO\_ACTUAL\_FACTURA < T.HASTA\_NUMERO\_FACTURA;  BEGIN  /\*  Si el numero\_actual\_factura pasó la validación anterior, asigna con dicho  valor el campo nro\_comprobante, y también se asigna el nro\_timbrado,  \*/  FOR REG IN C\_TIMBRADO LOOP  V\_NRO\_COMPROBANTE := REG.NUMERO\_ACTUAL\_FACTURA;  V\_NRO\_TIMBRADO := REG.NRO\_TIMBRADO;  END LOOP;  /\*  Finalmente actualiza la tabla D\_TIMBRADO incrementando el campo  numero\_actual\_factura en 1, siempre que dicho incremento no supere el  campo hasta\_numero\_factura.  \*/  UPDATE D\_TIMBRADO SET NUMERO\_ACTUAL\_FACTURA = NUMERO\_ACTUAL\_FACTURA + 1  WHERE NRO\_TIMBRADO = V\_NRO\_TIMBRADO AND NUMERO\_ACTUAL\_FACTURA <= HASTA\_NUMERO\_FACTURA;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Ocurrio un error');  END;  END IF;  ELSE  /\*  ● Si el código de operación no es de uso cajero, entonces:  - Acceder a la tabla D\_TIPO\_COMPROBANTE\_SECUENCIA correspondiente  al tipo de comprobante de la operación, correspondiente al año vigente, y  obtener el campo nro\_comprobante\_actual, y asignar al número de  comprobante. La columna timbrado queda nulo.  - Actualizar la tabla D\_TIPO\_COMPROBANTE\_SECUENCIA incrementando el  campo nro\_comprobante\_actual en 1.  \*/  SELECT NRO\_COMPROBANTE\_ACTUAL INTO V\_NRO\_COMPROBANTE  FROM D\_TIPO\_COMPROBANTE\_SECUENCIA TCS  JOIN D\_TIPO\_COMPROBANTE TC ON TC.COD\_TIPO\_COMPROBANTE = TCS.COD\_TIPO\_COMPROBANTE  JOIN D\_MOVIMIENTO\_OPERACIONES DMO ON TC.COD\_TIPO\_COMPROBANTE = DMO.COD\_TIPO\_COMPROBANTE  WHERE DMO.cod\_operacion = cod\_operacion AND ANHO = extract(year from SYSDATE);    UPDATE D\_TIPO\_COMPROBANTE\_SECUENCIA SET NRO\_COMPROBANTE\_ACTUAL = NRO\_COMPROBANTE\_ACTUAL + 1  WHERE COD\_TIPO\_COMPROBANTE = (  SELECT TCS.COD\_TIPO\_COMPROBANTE  FROM D\_TIPO\_COMPROBANTE\_SECUENCIA TCS  JOIN D\_TIPO\_COMPROBANTE TC ON TC.COD\_TIPO\_COMPROBANTE = TCS.COD\_TIPO\_COMPROBANTE  JOIN D\_MOVIMIENTO\_OPERACIONES DMO ON TC.COD\_TIPO\_COMPROBANTE = DMO.COD\_TIPO\_COMPROBANTE  WHERE DMO.cod\_operacion = cod\_operacion AND ANHO = extract(year from SYSDATE)    ) AND ANHO = (  SELECT TCS.ANHO  FROM D\_TIPO\_COMPROBANTE\_SECUENCIA TCS  JOIN D\_TIPO\_COMPROBANTE TC ON TC.COD\_TIPO\_COMPROBANTE = TCS.COD\_TIPO\_COMPROBANTE  JOIN D\_MOVIMIENTO\_OPERACIONES DMO ON TC.COD\_TIPO\_COMPROBANTE = DMO.COD\_TIPO\_COMPROBANTE  WHERE DMO.cod\_operacion = cod\_operacion AND ANHO = extract(year from SYSDATE)  );  END IF;  IF descripcion\_operacion IS NULL THEN  SELECT DESC\_OPERACION INTO V\_DESC\_OPERACION FROM D\_OPERACIONES OP WHERE OP.COD\_OPERACION = cod\_operacion;  ELSE  V\_DESC\_OPERACION := descripcion\_operacion;  END IF;  INSERT INTO D\_MOVIMIENTO\_OPERACIONES (  ID\_OPERACION,  fecha\_operacion,  cod\_sucursal,  cod\_operacion,  id\_persona,  nro\_caja,  id\_usuario,  cod\_tipo\_comprobante,  NRO\_COMPROBANTE,  TIPO\_REGISTRO,  DESCRIPCION\_OPERACION,  NRO\_TIMBRADO,  FECHA\_INSERT  ) VALUES(  (SELECT MAX(ID\_OPERACION)+1 FROM D\_MOVIMIENTO\_OPERACIONES),  fecha\_operacion,  cod\_sucursal,  cod\_operacion,  id\_persona,  nro\_caja,  id\_usuario,  1,  V\_NRO\_COMPROBANTE,  'A',  V\_DESC\_OPERACION,  V\_NRO\_TIMBRADO,  SYSDATE  );  ELSE  DBMS\_OUTPUT.PUT\_LINE('FECHA INCORRECTA');  END IF;    END;  /\*  c) El procedimiento P\_ACTUALIZAR\_STOCK que recibe como parámetros id\_producto,  cod\_sucursal, cantidad, uso\_stock  El procedimiento debe actualizar la tabla D\_STOCK\_SUCURSAL, que almacena el stock de  un producto en una sucursal, en base a los parámetros recibidos.  • Si el valor del parámetro USO\_STOCK es 2 (sumar), debe aumentar la  CANTIDAD\_EXISTENCIA en la tabla D\_STOCK\_SUCURSAL del producto dado por el  ID\_PRODUCTO, y en la sucursal determinada por el COD\_SUCURSAL.  • Si el valor del parámetro USO\_STOCK es 1 (restar), debe disminuir la  CANTIDAD\_EXISTENCIA en la tabla D\_STOCK\_SUCURSAL del producto dado por el  ID\_PRODUCTO, y en la sucursal determinada por el COD\_SUCURSAL. Ello siempre que la  (CANTIDAD\_EXISTENCIA - CANTIDAD\_OPERACION) >= STOCK\_MINIMO. Si no se cumple  \*/  PROCEDURE P\_ACTUALIZAR\_STOCK(  v\_id\_producto IN D\_STOCK\_SUCURSAL.ID\_PRODUCTO%TYPE,  v\_cod\_sucursal IN D\_STOCK\_SUCURSAL.COD\_SUCURSAL%TYPE,  v\_cantidad IN D\_STOCK\_SUCURSAL.CANTIDAD\_EXISTENCIA%TYPE,  v\_uso\_stock IN D\_OPERACIONES.USO\_STOCK%TYPE  )  IS  BEGIN  IF V\_USO\_STOCK = 2 THEN  UPDATE D\_STOCK\_SUCURSAL SET CANTIDAD\_EXISTENCIA = CANTIDAD\_EXISTENCIA + V\_CANTIDAD  WHERE ID\_PRODUCTO = V\_ID\_PRODUCTO  AND COD\_SUCURSAL = v\_cod\_sucursal;  ELSIF V\_USO\_STOCK = 1 THEN  UPDATE D\_STOCK\_SUCURSAL SET CANTIDAD\_EXISTENCIA = CANTIDAD\_EXISTENCIA - V\_CANTIDAD  WHERE ID\_PRODUCTO = V\_ID\_PRODUCTO  AND COD\_SUCURSAL = v\_cod\_sucursal  AND (CANTIDAD\_EXISTENCIA - V\_CANTIDAD) >= STOCK\_MINIMO;  ELSE  DBMS\_OUTPUT.PUT\_LINE('\*\*\*USO STOCK NO VALIDO\*\*\*');  END IF;  END;    /\*  d) El procedimiento P\_INSERTAR\_DETALLE que recibe como parámetros: id\_operacion,  id\_producto, cod\_medida, cantidad\_operacion.  El procedimiento deberá:  Recuperar la sucursal, el código de movimiento y el correspondiente uso\_stock del  mismo, a partir de la cabecera D\_MOVIMIENTO\_OPERACIONES.  Insertar un registro en la tabla D\_DETALLE\_OPERACIONES asignando los siguientes  campos:  ● precio\_operacion = precio de última compra del producto + la aplicación del  porcentaje de beneficio.  ● importe\_operacion = precio\_operacion x cantidad\_operacion  ● cod\_tipo\_iva se obtiene a partir del producto  ● porcentaje\_iva se obtiene a partir del tipo de iva  ● importe\_iva = (precio\_operacion x cantidad\_operacion) / divisor\_iva\_incluido  ● importe\_descuento e importe\_recargo se asigna 0  Actualizar el stock invocando al procedimiento P\_ACTUALIZAR\_STOCK, enviando los  parámetros requeridos  \*/  PROCEDURE P\_INSERTAR\_DETALLE(  v\_id\_operacion IN D\_MOVIMIENTO\_OPERACIONES.ID\_OPERACION%TYPE,  v\_id\_producto IN D\_DETALLE\_OPERACIONES.ID\_PRODUCTO%TYPE,  v\_cod\_medida IN D\_DETALLE\_OPERACIONES.COD\_MEDIDA%TYPE,  v\_cantidad\_operacion IN D\_DETALLE\_OPERACIONES.CANTIDAD\_OPERACION%TYPE  )  IS  CURSOR C\_DETALLE\_INSERT IS  SELECT SUC.DESC\_SUCURSAL, SUC.COD\_SUCURSAL, DMO.ID\_OPERACION, OP.USO\_STOCK  FROM D\_MOVIMIENTO\_OPERACIONES DMO  JOIN D\_SUCURSAL SUC ON DMO.COD\_SUCURSAL = SUC.COD\_SUCURSAL  JOIN D\_OPERACIONES OP ON DMO.COD\_OPERACION = OP.COD\_OPERACION  WHERE DMO.ID\_OPERACION = V\_ID\_OPERACION;  v\_uso\_stock D\_OPERACIONES.USO\_STOCK%TYPE;  v\_cod\_sucursal D\_STOCK\_SUCURSAL.COD\_SUCURSAL%TYPE;  v\_id\_operacion\_ D\_DETALLE\_OPERACIONES.ID\_OPERACION%TYPE;  BEGIN  FOR REG IN C\_DETALLE\_INSERT LOOP  v\_uso\_stock := REG.USO\_STOCK;  v\_cod\_sucursal := REG.COD\_SUCURSAL;  v\_id\_operacion\_ := REG.ID\_OPERACION;  END LOOP;  INSERT INTO D\_DETALLE\_OPERACIONES (  ID\_REGISTRO,  ID\_OPERACION,  ID\_PRODUCTO,  COD\_MEDIDA,  CANTIDAD\_OPERACION,  PRECIO\_OPERACION,  IMPORTE\_OPERACION,  IMPORTE\_DESCUENTO,  cod\_tipo\_iva,  porcentaje\_iva,  importe\_iva,  importe\_recargo)  VALUES(  (SELECT MAX(ID\_REGISTRO) +1 FROM D\_DETALLE\_OPERACIONES),  v\_id\_operacion\_,  V\_ID\_PRODUCTO,  v\_cod\_medida,  v\_cantidad\_operacion,  (SELECT PRECIO\_ULTIMA\_COMPRA + PRECIO\_ULTIMA\_COMPRA \* PORCENTAJE\_BENEFICIO  FROM D\_PRODUCTOS WHERE ID\_PRODUCTO = V\_ID\_PRODUCTO),  (SELECT (PRECIO\_ULTIMA\_COMPRA + PRECIO\_ULTIMA\_COMPRA \* PORCENTAJE\_BENEFICIO) \* v\_cantidad\_operacion  FROM D\_PRODUCTOS WHERE ID\_PRODUCTO = V\_ID\_PRODUCTO),  0,  (SELECT I.COD\_TIPO\_IVA  FROM D\_TIPO\_IVA I  JOIN D\_PRODUCTOS P ON I.COD\_TIPO\_IVA = P.COD\_TIPO\_IVA  WHERE P.ID\_PRODUCTO = V\_ID\_PRODUCTO)  ,  (SELECT porcentaje\_iva FROM D\_TIPO\_IVA I  JOIN D\_PRODUCTOS P ON P.COD\_TIPO\_IVA = I.COD\_TIPO\_IVA  WHERE P.ID\_PRODUCTO = V\_ID\_PRODUCTO),  (SELECT ((P.PRECIO\_ULTIMA\_COMPRA + P.PRECIO\_ULTIMA\_COMPRA \* P.PORCENTAJE\_BENEFICIO) \* v\_cantidad\_operacion )/ I.divisor\_iva\_incluido  FROM D\_PRODUCTOS P  JOIN D\_TIPO\_IVA I ON I.COD\_TIPO\_IVA = P.COD\_TIPO\_IVA  WHERE P.ID\_PRODUCTO = V\_ID\_PRODUCTO),  0  );  P\_ACTUALIZAR\_STOCK(V\_ID\_PRODUCTO, v\_cod\_sucursal, v\_cantidad\_operacion, V\_USO\_STOCK);  END;  /\*  La función F\_VER\_DETALLE que recibe como parámetro el ID\_OPERACION y devuelve  una variable del tipo T\_DETALLE. La función deberá  • Verificar que exista la operación correspondiente al ID\_OPERACION introducido. Si no  existe, dará error.  • Si existe, recorrerá los movimientos de detalle y llenará una variable del tipo T\_DETALLE,  la que será retornada.  TYPE r\_articulo IS RECORD (  ID\_PRODUCTO D\_DETALLE\_OPERACIONES.ID\_PRODUCTO%TYPE,  CANTIDAD D\_DETALLE\_OPERACIONES.CANTIDAD%TYPE  );  TYPE t\_detalle IS TABLE OF  r\_articulo INDEX BY BINARY\_INTEGER;  \*/  FUNCTION F\_VER\_DETALLE(ID\_OPERACION IN D\_MOVIMIENTO\_OPERACIONES.ID\_OPERACION%TYPE)  RETURN T\_DETALLE  IS  CURSOR C\_MOVIMIENTO IS  SELECT ID\_OPERACION  FROM D\_DETALLE\_OPERACIONES  WHERE D\_DETALLE\_OPERACIONES.ID\_OPERACION = ID\_OPERACION;  EXISTE BOOLEAN;  V\_INDICE NUMBER;  V\_DETALLE T\_DETALLE;  CURSOR C\_DETALLE IS  SELECT ID\_PRODUCTO, CANTIDAD\_OPERACION FROM D\_DETALLE\_OPERACIONES  WHERE D\_DETALLE\_OPERACIONES.ID\_OPERACION = ID\_OPERACION;  BEGIN  EXISTE := FALSE;  V\_INDICE := 0;  FOR REG\_MOV IN C\_MOVIMIENTO LOOP  IF ID\_OPERACION = REG\_MOV.ID\_OPERACION THEN  EXISTE := TRUE;  END IF;  END LOOP;  IF EXISTE = FALSE THEN  RAISE\_APPLICATION\_ERROR(-20001, 'No existe la operación');  ELSE  FOR R\_DETALLE IN C\_DETALLE LOOP  V\_DETALLE(V\_INDICE).ID\_PRODUCTO := R\_DETALLE.ID\_PRODUCTO;  V\_DETALLE(V\_INDICE).CANTIDAD := R\_DETALLE.CANTIDAD\_OPERACION;  V\_INDICE := V\_INDICE + 1;  END LOOP;  RETURN V\_DETALLE;  END IF;  END;  END; |
| --- |

**PUNTUACIÓN**

| **Ejercicio** | **Criterio** | **Puntos** |
| --- | --- | --- |
|  | Sintaxis general de la especificación del paquete y concordancia con el cuerpo | 1 |
| a) | Declara los procedimientos, excepciones y tipo de dato | 1 |
| b) | Implementa correctamente el procedimiento p\_insertar\_movimiento | 5 |
| c) | Implementa correctamente el procedimiento p\_actualizar\_stock | 3 |
| d) | Implementa correctamente el procedimiento p\_insertar\_detalle | 5 |
| e) | Implementa correctamente la función F\_VER\_DETALLE | 3 |
|  | **TOTAL** | **18** |