# Estructura PLSQL:

-Estructura de un bloque PLSQL

-Nunca deben ir vacios los bloques

DECLARE

Declaracion de variables

BEGIN

Ejecucion de las instrucciones

EXCEPTION

Control de excepciones

END ;

# Tabla vacia

-Sirve para mostrar salidas de pantalla

select 'VALENTINO'

from dual;

# Trunc NVL

-NVL : mostrar un valor en caso de que estuviera vacio el primero

Nvl(campo,valorAuxiliar si el campo es nulo)

-Trunc: solo muestra la fecha en un dato sysdate(fecha hora actual)

AND TRUNC(SYSDATE) BETWEEN ppa\_start\_date AND NVL(ppa\_end\_date, TRUNC(SYSDATE));

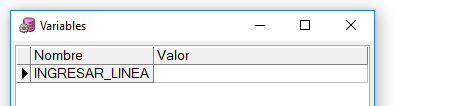
EXCEPTION

# INGRESAR\_VALOR

BEGIN

linea varchar(50):='&INGRESAR\_LINEA';

END;



# Variable:

nombre varchar(50):='MARCELO';

edad number:=25;

v\_ESC\_ESN\_HEXA cellulars.clu\_cellular\_number%type;

**tipo de datos**

v\_cod\_pais varchar2(3);*-- cantidad de caracteres '549' / comillas no se cuentan*

# SALIDA\_PANTALLA

DECLARE

nombre varchar(50):='MARCELO';

apellido varchar(50):='GONZALES';

edad number:=25;

BEGIN

dbms\_output.put\_line(nombre);

dbms\_output.put\_line(apellido);

dbms\_output.put\_line(edad);

END;

-

dbms\_output.put\_line(edad);

dbms\_output.put\_line(‘23’);

dbms\_output.put\_line(‘JORGE’);

-salto de renglon

dbms\_output.put\_line(edad||chr(13));

# Excepcion

e\_tn3 exception;

raise e\_tn3;

Exception

when e\_tn3 then

dbms\_output.put\_line('la linea no se puede recomprar');

when no\_data\_found then

dbms\_output.put\_line('No se encontraron datos');

**-Ejemplo de las 3 excepciones**

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

Dbms\_Output.Put\_Line(Substr('ERROR NO\_DATA\_FOUND: Parametro DEBUG\_ENABLE. CODIGO: ' || SQLCODE,0,1000));

WHEN TOO\_MANY\_ROWS THEN

Dbms\_Output.Put\_Line(Substr('ERROR TOO\_MANY\_ROWS: Parametro DEBUG\_ENABLE. CODIGO: ' || SQLCODE,0,1000));

WHEN OTHERS THEN

Dbms\_Output.Put\_Line(Substr('ERROR OTHERS: Parametro DEBUG\_ENABLE. CODIGO: ' || SQLCODE,0,1000));

END;

# Cursores Implicitos:

-dentro de begin-end

-tienen una variable implícita

Select CONCAT(ESC\_ESN\_HEXA,'F')

into v\_ESC\_ESN\_HEXA

From ESN\_CELLULARS

Where ESC\_CLU\_CELLULAR\_NUMBER = (select clu\_cellular\_number from cellulars where clu\_bill\_number ='3512004951')

And ESC\_END\_DATE IS NULL ;

* Opcion 1: si cada sentencia se encierra en un bloque contemplando una excepcion , y se produce un error, la excepcion se lanza pero luego sigue por el flujo descendente ejecutando los demas bloques
* Opcion 2: si varias sentencias juntas se controlan por la misma excepcion principal el programa deja de ejecutarse

-OPCION 3: CADA SENTENCIA TENGA UNA EXCEPCION NOT\_DATAFOUND Y LANZA UNA PERSONALIZADA PARA SABER DE DONDE VIENE LA EXCEPCION

BEGIN

select clu\_cellular\_number

into v\_clu\_cellular\_number

from cellulars

where clu\_bill\_number ='21212';

EXCEPTION

when no\_data\_found then

e\_error\_descripcion:='clu\_cellular\_number';

raise e\_error;

when too\_many\_rows then

e\_error\_descripcion:='error en '||' '||'clu\_cellular\_number';

raise e\_error;

END;

EXCEPTION

when e\_error then

dbms\_output.put\_line('ERROR EN '||' '||e\_error\_descripcion);

Funcion

-Ejecutar

Parametros entrada :

-Instanciar variable : declarar e iniciar .

Parametros salida: se debe declarar la variable ( nombre-tipo) / evitar insertarlo (null)

v\_result := PP\_PROVISIONING\_INTERFACE.create\_subscriber (

Login => v\_cod\_pais || nro\_celular,

AccountBalance => 8000000,

ExpiryDate => SYSDATE,

IMSI => v\_sim\_imsi,

PEDESCRIPTION => V\_PEDESCRIPTION,*--parametro salida*

Profileid => v\_profile\_tecnomen,

PrefDestList0 => NULL,

Parametros\_funcion

Salida

Entreda

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |