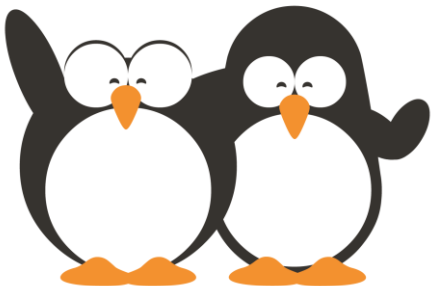
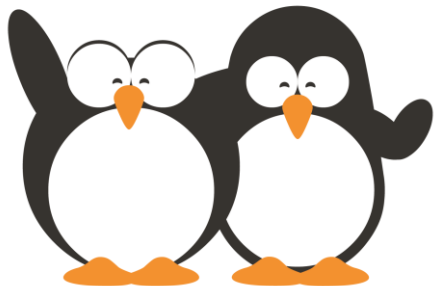


- 1.PROGRAMACIÓN ORIENTADA A OBJETOS
- 2.TABLAS (VARRAY)
- 3.EXPLICACIÓN DE LA PAC



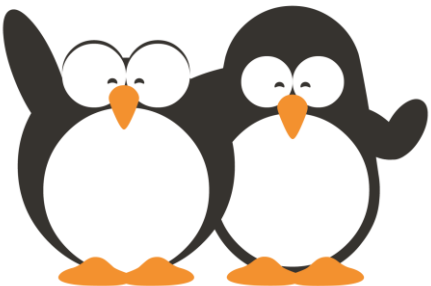
PROGRAMACIÓN ORIENTADA A OBJETOS

- Un **TIPO DE OBJETO** es un concepto abstracto que asocia a un elemento unas características (atributos) y unas funcionalidades (métodos).
- Las bases de datos objetos relacionales son aquellas bases de datos que se basan en la creación de **TIPOS DE OBJETOS** y la **CONSTRUCCIÓN** de nuevos objetos.



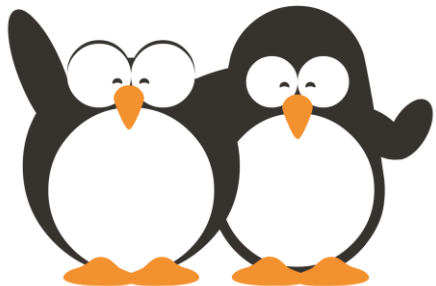
PROGRAMACIÓN ORIENTADA A OBJETOS

```
CREATE OR REPLACE TYPE T_CUADRADO AS OBJECT  
(  
  alto NUMBER(3),  
  ancho NUMBER(3),  
  MEMBER FUNCTION perimetro RETURN NUMBER  
);
```



PROGRAMACIÓN ORIENTADA A OBJETOS

```
CREATE OR REPLACE TYPE BODY T_CUADRADO AS  
MEMBER FUNCTION perimetro RETURN NUMBER  
IS  
  br NUMBER;  
BEGIN  
  br := ancho*2;  
  br := br + (alto*2);  
  RETURN br;  
END;  
END;  
/
```



PROGRAMACIÓN ORIENTADA A OBJETOS

```
CREATE TABLE cuadrados OF T_CUADRADO;
```

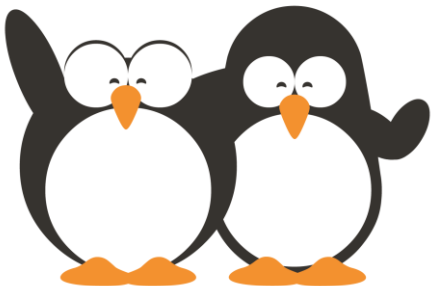
```
INSERT INTO cuadrados VALUES (10, 10);
```

```
INSERT INTO cuadrados VALUES (3,10);
```

```
INSERT INTO cuadrados VALUES (20,5);
```

```
SELECT c.alto,c.ancho,c.perimetro()FROM cuadrados c;
```

```
/
```

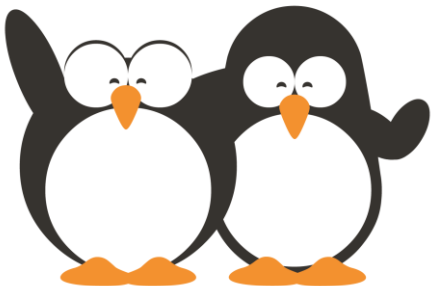


PROGRAMACIÓN ORIENTADA A OBJETOS

CONSTRUCTOR

Para crear un objeto es preciso, una vez se ha definido la clase, crear una variable y aplicar una función especial NEW.

- función especial que tiene el mismo nombre que la clase.
- función para inicializar aquellas propiedades.



PROGRAMACIÓN ORIENTADA A OBJETOS

Declare

p1 persona;

Begin

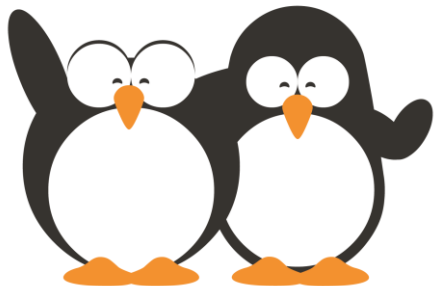
p1:=new persona('jose','espinosa','lleida');

dbms.....

insert()...

End;

```
CREATE OR REPLACE TYPE PERSONA
AS OBJECT
(
  number VARCHAR2(30),
  apellido VARCHAR2(30),
  poblacion VARCHAR2(30)
);
```



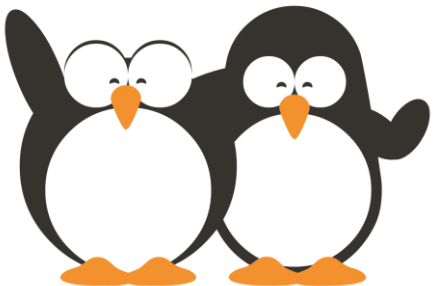
COLECCIONES EN PL/SQL

- VARRAYS:

Un varray se manipula de forma muy similar a las tablas de PL, pero se implementa de forma diferente. Los elementos en el varray se almacenan comenzando en el índice 1 hasta la longitud máxima declarada en el tipo varray.

La sintaxis general es la siguiente:

```
TYPE <nombre_tipo> IS VARRAY  
(<tamaño_maximo>) OF <tipo_elementos>;
```

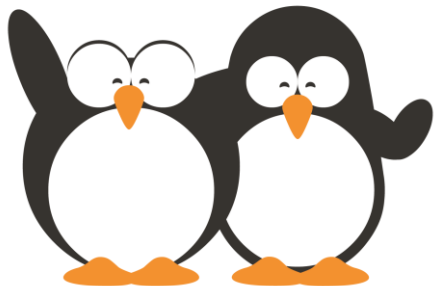


PIZARRA

<u>mi-tabla</u>	VARIABLE				
varchar2	jose	ana	adrian	Sara	marcos

Ejercicio 3 ⇒ VARIABLE

Ejercicio 4 ⇒ NESTED TABLE.

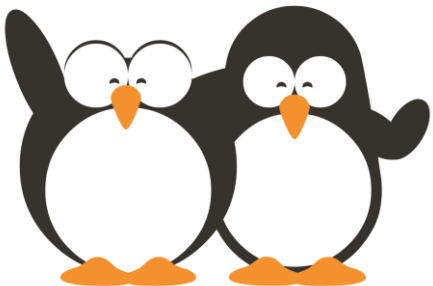


PROGRAMACIÓN ORIENTADA A OBJETOS

- Una consideración a tener en cuenta es que en la declaración de un varray el tipo de datos no puede ser de los siguientes tipos de datos:

BOOLEAN / NCHAR / NVARCHAR(n) / REF CURSOR
TABLE/ VARRAY

Sin embargo se puede especificar el tipo utilizando los atributos **%TYPE** y **%ROWTYPE**.



EJEMPLO VARRAY

DECLARE

/ Declaramos el tipo VARRAY de cinco elementos VARCHAR2 */*

TYPE t_cadena IS VARRAY(5) OF VARCHAR2(50);

/ Asignamos los valores con un constructor */*

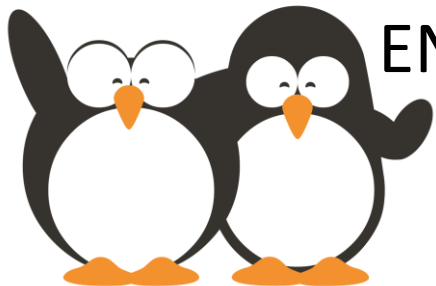
v_lista t_cadena:= t_cadena('Aitor', 'Alicia', 'Pedro', '', '');

BEGIN

v_lista(4) := 'Tita';

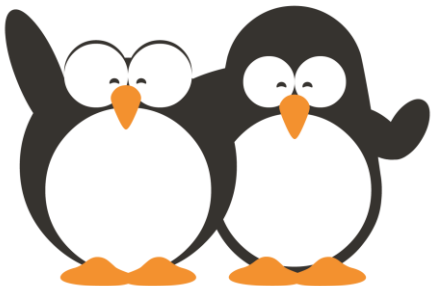
v_lista(5) := 'Ainhoa';

END;



EJEMPLO VARRAY

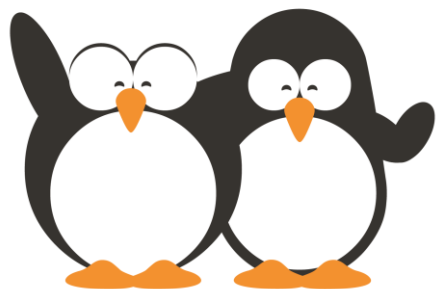
```
CREATE OR REPLACE TYPE tipoTelefono
AS VARRAY (3) of VARCHAR(10)
/
CREATE TABLE Empleado
(DNI NUMBER,
 nombre VARCHAR2(30),
 telefonos_contacto tipoTelefono);
/
INSERT INTO Empleado
VALUES ('9876543', 'Pepe', tipoTelefono ('914445566', '606445566', '934445566'));
```



PIZARRA

EMPLEADO

DNI	NOMBRE	TELÉFONO CONTACTO		
98....	Pepe	914 ...	606	934...

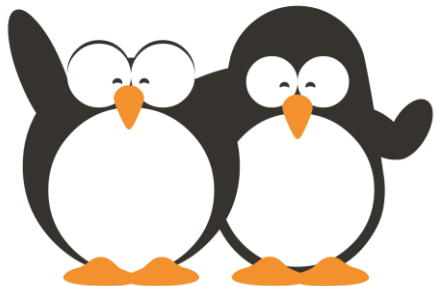


COLECCIONES PL/SQL

- NESTED TABLES O TABLAS ANIDADAS

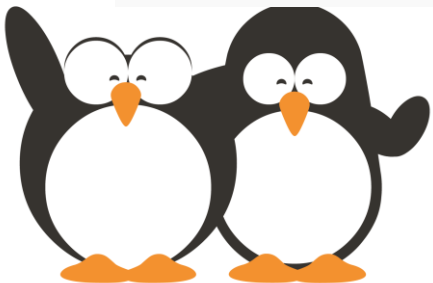
Es posible definir un tipo de datos como una tabla, y utilizar dicho tipo como el tipo de datos de la columna de otra tabla.

- De este modo, la columna contendrá una colección de valores, objetos o referencias, que se almacenarán en formato de tabla.
- Aunque Oracle almacena las filas de una NESTED TABLE sin orden



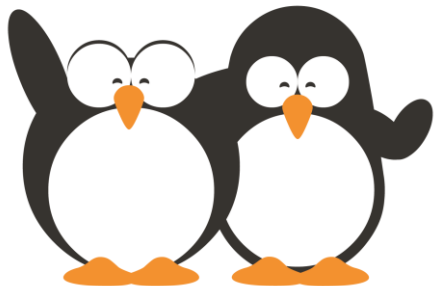
EJEMPLO NESTED TABLES

```
CREATE OR REPLACE TYPE my_tab_t AS TABLE OF  
VARCHAR2(30);  
/  
CREATE TABLE nested_table (id NUMBER, col1  
my_tab_t)  
    NESTED TABLE col1 STORE AS col1_tab;
```



EJEMPLO NESTED TABLES

```
INSERT INTO nested_table VALUES (1, my_tab_t('A'));  
INSERT INTO nested_table VALUES (2, my_tab_t('B', 'C'));  
INSERT INTO nested_table VALUES (3, my_tab_t('D', 'E', 'F'));  
COMMIT;
```



PIZARRA

NESTED TABLE

id	col 1. (NESTED TABLES)		
1	'A'		
2	'B'	'C'	
3	'D'	'E'	'F'

