**Desarrollo de aplicaciones multiplataforma**

**Acceso a datos**

**Tema 2-Práctica 4: Métodos**

**Mohamed El Younousi Bentalha**

Se desea modelar un objeto triángulo para almacenar sus características (la base y la altura), y almacenar en la BBDD cientos de triángulos pudiendo calcular el área de cada triángulo. Se podría crear el siguiente esquema:

* Un tipo llamado tipo\_triángulo con los atributos **base number, altura number** y una **función** llamada  **area return number.** Recordar que el área de un triángulo se calcula como (base\*altura)/2.

**Respuesta:**

CREATE OR REPLACE TYPE tipo\_triangulo AS OBJECT (

base NUMBER,

altura NUMBER,

MEMBER FUNCTION area RETURN NUMBER

);

CREATE OR REPLACE TYPE BODY tipo\_triangulo AS

MEMBER FUNCTION area RETURN NUMBER IS

BEGIN

RETURN (self.base \* self.altura) / 2;

END;

END;

/

* Crear el cuerpo **body** para dicho tipo.

**Respuesta:**

* Crear una tabla relacional llamada triangulos para almacenar los triángulos, con las columnas **Id number** y **triangulo** de **tipo\_triangulo.**

**Respuesta:**

CREATE TABLE triangulos (

id NUMBER,

triangulo tipo\_triangulo

);

* Insertar dos triángulos con los siguientes valores

Id=1, base=5, altura=5

Id=2, base=10, altura=10

**Respuesta:**

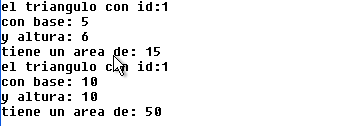
INSERT INTO triangulos VALUES (1, tipo\_triangulo(5, 5));

INSERT INTO triangulos VALUES (2, tipo\_triangulo(10, 10));

* Listar todos los triángulos.

**Respuesta:**

* Crear un bloque PL/SQL para recorrer la tabla triángulos e invocar al método área, de forma que en el resultado obtengamos esto:



**Respuesta:**

DECLARE

v\_id triangulos.id%TYPE;

v\_base triangulos.triangulo.base%TYPE;

v\_altura triangulos.triangulo.altura%TYPE;

v\_area NUMBER;

BEGIN

FOR tri\_rec IN (SELECT id, triangulo FROM triangulos) LOOP

v\_id := tri\_rec.id;

v\_base := tri\_rec.triangulo.base;

v\_altura := tri\_rec.triangulo.altura;

v\_area := tri\_rec.triangulo.area();

DBMS\_OUTPUT.PUT\_LINE('El triángulo con id: ' || v\_id);

DBMS\_OUTPUT.PUT\_LINE('con base: ' || v\_base);

DBMS\_OUTPUT.PUT\_LINE('y altura: ' || v\_altura);

DBMS\_OUTPUT.PUT\_LINE('tiene un área de: ' || v\_area);

END LOOP;

END;

/