

```

/* Activar la posibilidad de mostrar datos en pantalla */
SET SERVEROUTPUT ON;

/* Declaración del tipo de objeto Usuario */
CREATE OR REPLACE TYPE Usuario AS OBJECT (
    /* Declaración de los atributos */
    login VARCHAR2(10),
    nombre VARCHAR2(30),
    f_ingreso DATE,
    credito NUMBER,

    /*Declaración de los métodos */
    CONSTRUCTOR FUNCTION Usuario(login VARCHAR2, credito NUMBER) RETURN SELF AS
    RESULT,
    MEMBER PROCEDURE setNombre(Nombre VARCHAR2),
    MEMBER PROCEDURE setNombre(Nombre VARCHAR2, Apellidos VARCHAR2),
    MEMBER FUNCTION getNombre RETURN VARCHAR2
);
/

/* Cuerpo del tipo de objeto Usuario */
CREATE OR REPLACE TYPE BODY Usuario AS
    /* Implementación del método constructor */
    CONSTRUCTOR FUNCTION Usuario(login VARCHAR2, credito NUMBER) RETURN SELF AS
    RESULT IS
    BEGIN
        if(credito >= 0) then
            SELF.credito := credito;
        else
            SELF.credito := 0;
        END IF;
        RETURN;
    END;

    /* Implementación de dos métodos sobrecargados */
    MEMBER PROCEDURE setNombre(Nombre VARCHAR2) IS
    BEGIN
        SELF.Nombre := Nombre;
    END setNombre;

    MEMBER PROCEDURE setNombre(Nombre VARCHAR2, Apellidos VARCHAR2) IS
    BEGIN
        SELF.Nombre := CONCAT(Nombre, Apellidos);
    END setNombre;

    /* Implemetación de un método con retorno de valores*/
    MEMBER FUNCTION getNombre RETURN VARCHAR2 IS
    BEGIN
        RETURN SELF.nombre;
    END getNombre;
END;
/

```

```
DECLARE
    /* Declaración de una variable de tipo Usuario */
    u1 Usuario;
BEGIN
    /* Creación de un objeto Usuario */
    u1 := NEW Usuario('user1', 100);
    /* Llamada al método setNombre */
    u1.setNombre('Usuario1 ', 'Apellidos1');
    /* Uso del método getNombre */
    dbms_output.put_line(u1.getNombre);
END;
/
```