

TAREA 7.1



BASE DE DATOS

I.E.S. BRIANDA DE MENDOZA

- CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA-

Alumno

Álvaro Bajo Tabero | abajot01@educastillalamancha.es

Profesor

Martín Martínez Carrilero | mmmc117@educastillalamancha.es

28 abril de 2023

1. Crea el tipo de objetos "MiembroEscolar" con los siguientes atributos:

```
codigo INTEGER,
dni VARCHAR2(10),
nombre VARCHAR2(30),
apellidos VARCHAR2(30),
sexo VARCHAR2(1),
fecha_nac DATE
```

The screenshot shows the Oracle Application Express interface. At the top, there's a user profile for 'ÁB Álvaro Bajo Tabero'. Below that is a navigation bar with 'Home', 'Application Builder', and 'SQL Workshop'. The 'SQL Workshop' tab is active, and the breadcrumb trail shows 'Home > SQL Workshop > SQL Commands'. The main area displays the SQL command to create a type object:

```
CREATE OR REPLACE TYPE MiembroEscolar AS OBJECT(
  codigo INTEGER,
  dni VARCHAR2(10),
  nombre VARCHAR2(30),
  apellidos VARCHAR2(30),
  sexo VARCHAR2(1),
  fecha_nac DATE
) NOT FINAL;
```

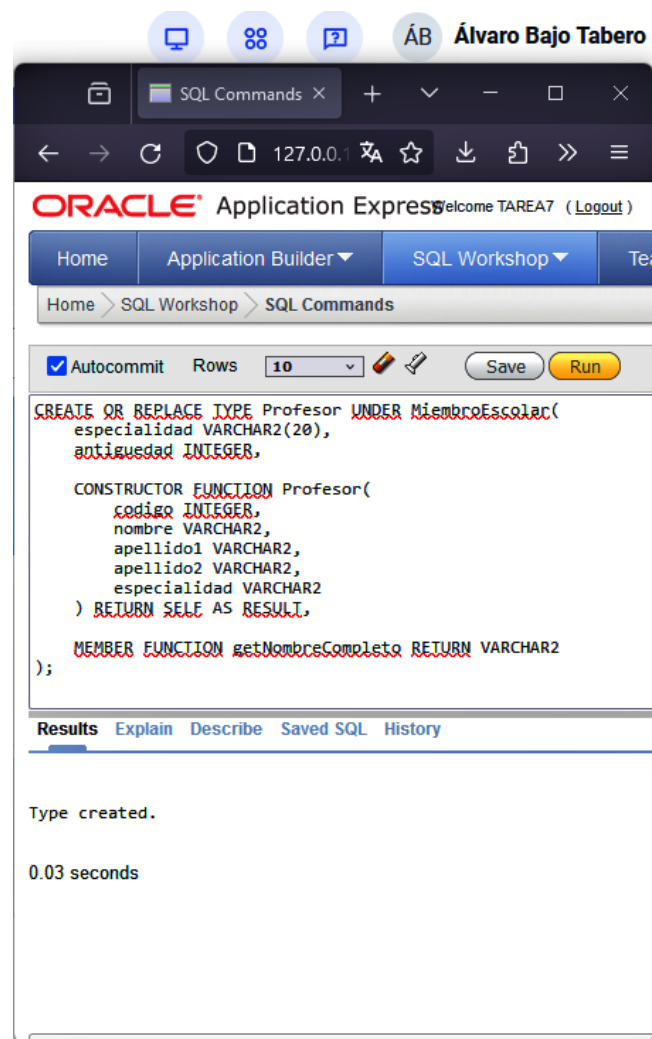
Below the command, the 'Results' tab is selected, showing the output: 'Type created.' and '0.06 seconds'.

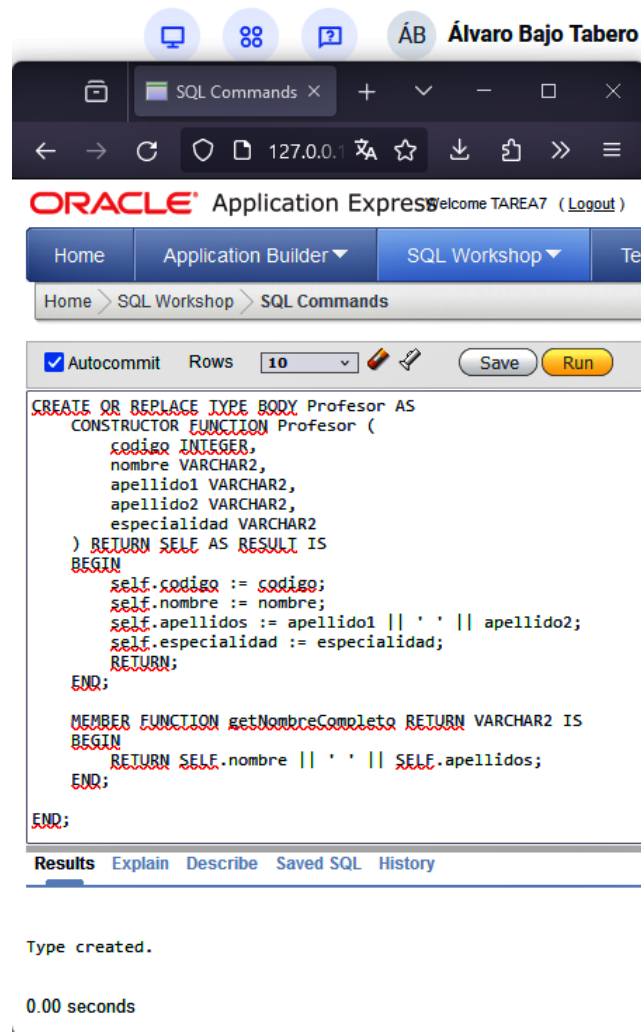
2. Crea, como tipo heredado de "MiembroEscolar", el tipo de objeto "Profesor" con los siguientes atributos y métodos:

```
especialidad VARCHAR2(20),
antigüedad INTEGER
```

Método constructor para el tipo de objetos "Profesor", en el que se indiquen como parámetros el código, nombre, primer apellido, segundo apellido y especialidad. Este método debe asignar al atributo "apellidos" los datos de primer apellido y segundo apellido que se han pasado como parámetros, uniéndolos con un espacio entre ellos.

Método "getNombreCompleto" para el tipo de objetos "Profesor" que permita obtener su nombre completo con el formato "apellidos nombre".

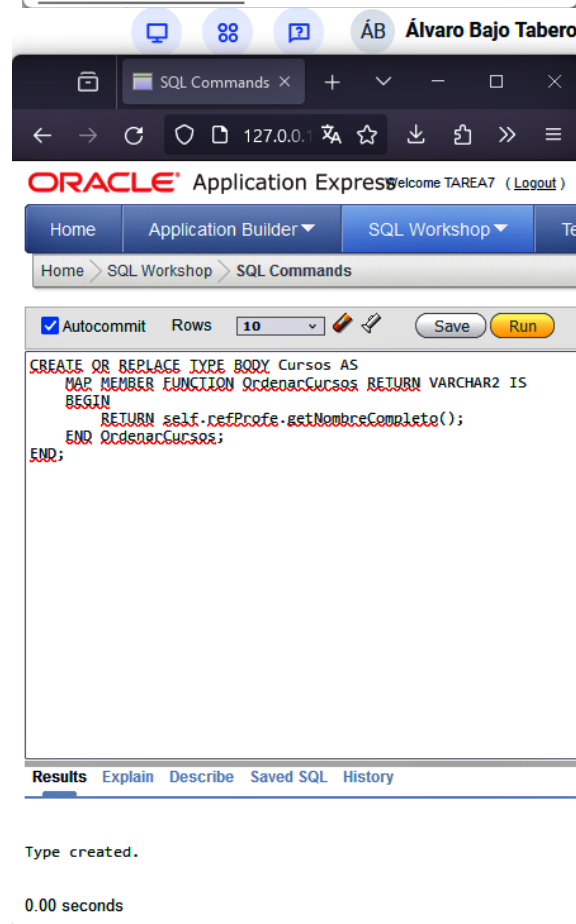
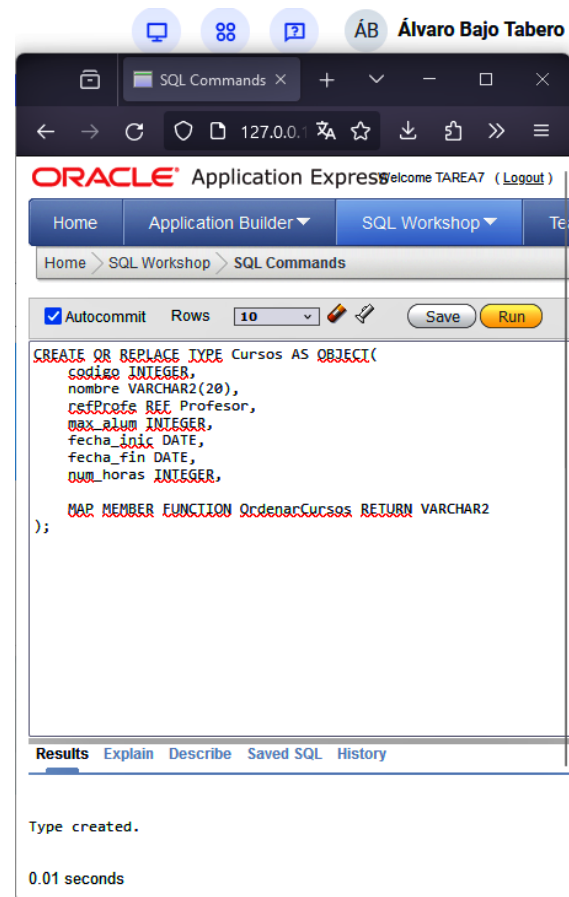




3. Crea el tipo de objeto "**Cursos**" con los siguientes atributos y métodos:

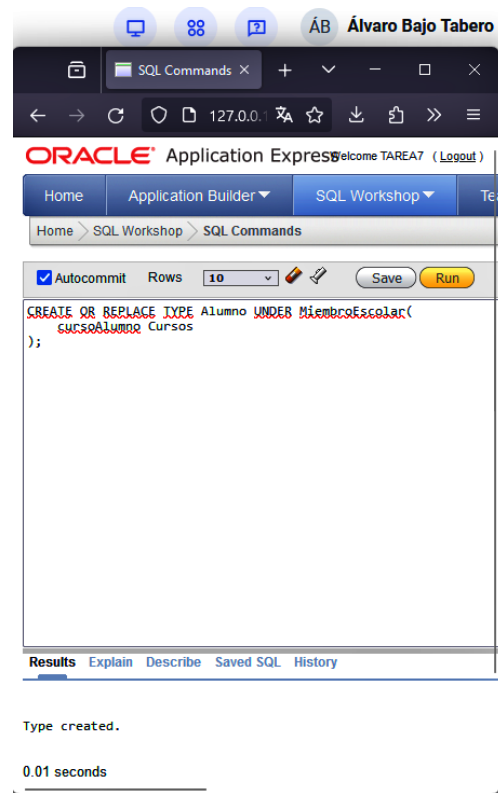
```
codigo INTEGER,
nombre VARCHAR2(20),
refProfe REF Profesor,
max_Alumn INTEGER,
fecha_Inic DATE,
fecha_Fin DATE,
num_Horas INTEGER
```

Método MAP "**ordenarCursos**" para el tipo "**Cursos**". Este método debe retornar el nombre completo del profesor al que hace referencia cada curso. Para obtener el nombre debes utilizar el método **getNombreCompleto** que se ha creado anteriormente.

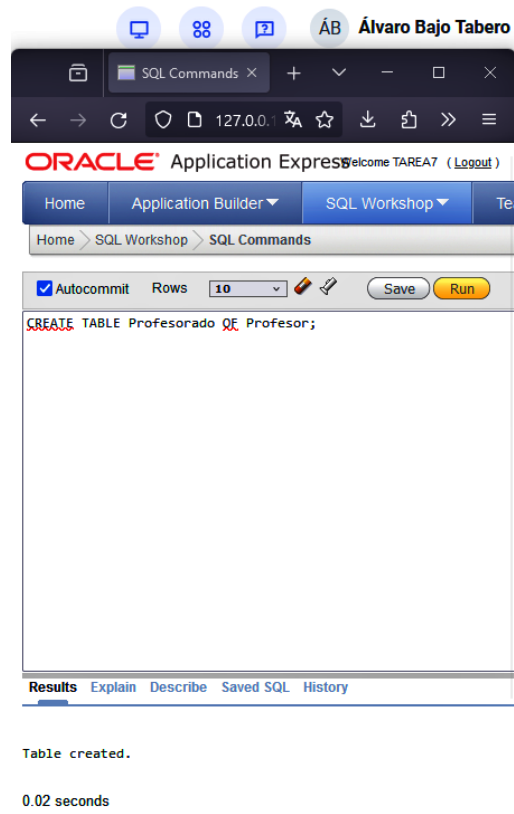


4. Crea, como tipo heredado de "MiembroEscolar", el tipo de objeto "Alumno" con los siguientes atributos:

cursoAlumno Cursos



5. Crea un tipo de tabla "Profesorado" de objetos "Profesor".

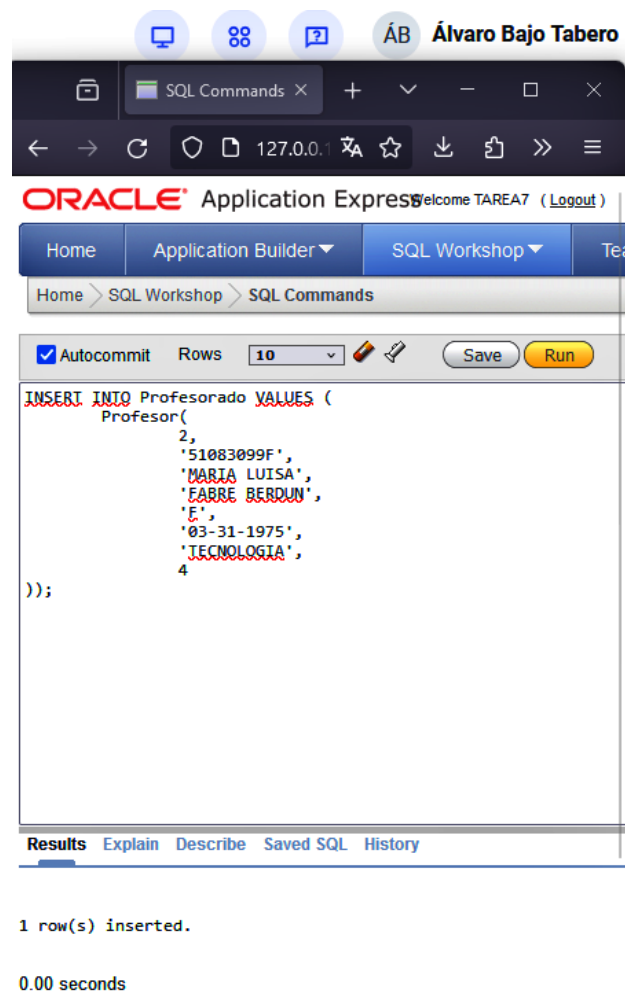


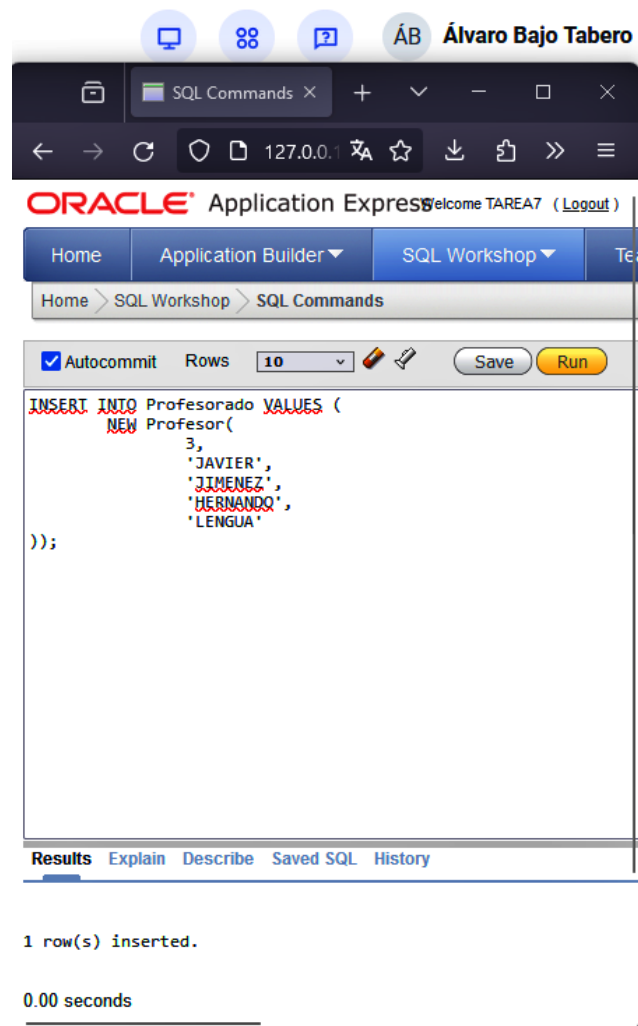
6. Inserta en dicha tabla dos objetos "Profesor". El primero de ellos con los datos:

```
codigo: 2
dni: 51083099F
nombre: MARIA LUISA
apellidos: FABRE BERDUN
sexo: F
fecha_nac: 31/03/1975
especialidad: TECNOLOGIA
antiguedad: 4
```

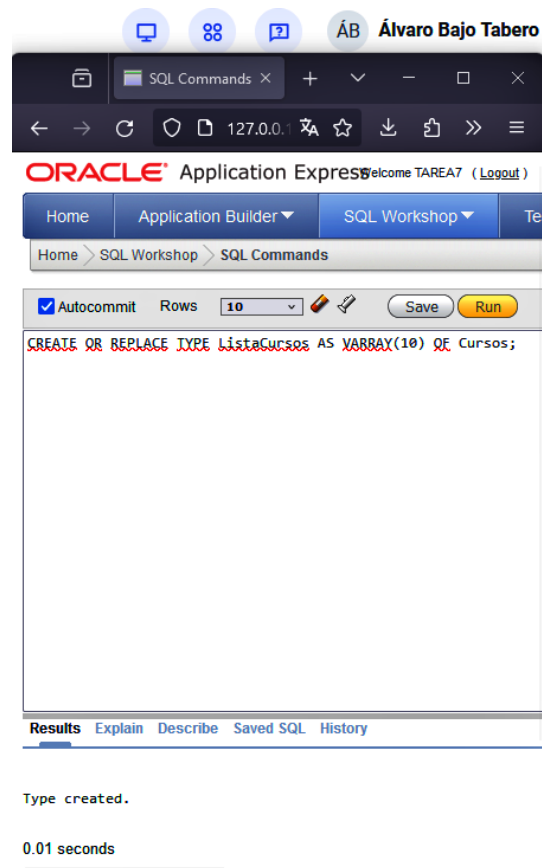
El segundo objeto "Profesor" debes crearlo usando el método constructor que has realizado anteriormente. Debes usar los siguientes datos:

```
codigo: 3
nombre: JAVIER
apellidos: JIMENEZ HERNANDO
especialidad: LENGUA
```

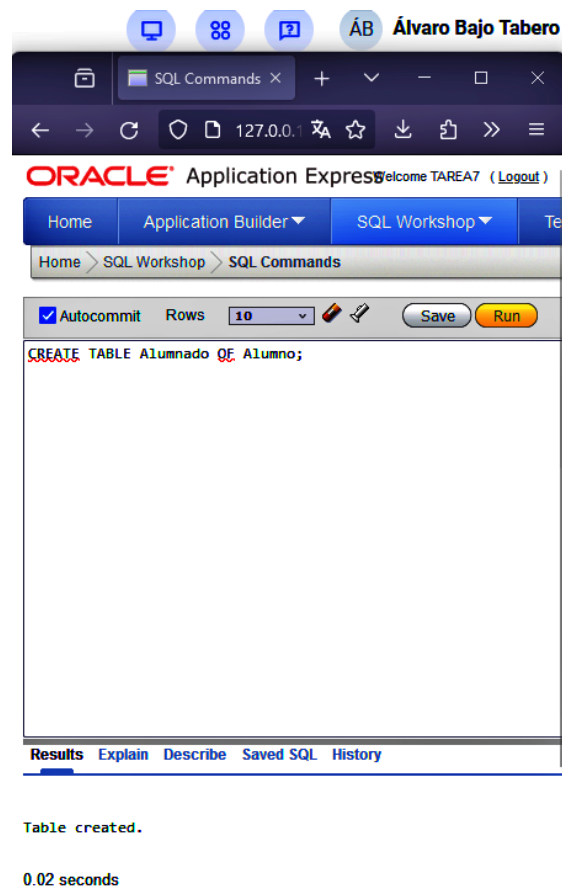




7. Crea un tipo colección **VARRAY** llamada "**ListaCursos**" en la que se puedan almacenar hasta 10 objetos "**Cursos**".



8. Crea una tabla "Alumnado" de objetos "Alumno".



9. Crea un bloque de código que haga todo lo siguiente:

a) Guarda en una instancia "**listaCursos1**" de dicha lista, los dos cursos siguientes:

El primer curso que debes almacenar en dicha lista debe tener los siguientes datos:

```
codigo: 1
nombre: Curso 1
refProfe: Referencia al profesor cuyo codigo es 3.
max_Alumn: 20
fecha_Inic: 1/6/2011
fecha_Fin: 30/6/2011
num_Horas: 30
```

El segundo curso que debes almacenar en dicha lista debe tener los siguientes datos:

```
codigo: 2
nombre: Curso 2
refProfe: Referencia al profesor cuyo DNI es 51083099F.
max_Alumn: 20
fecha_Inic: 1/6/2011
fecha_Fin: 30/6/2011
num_Horas: 30
```

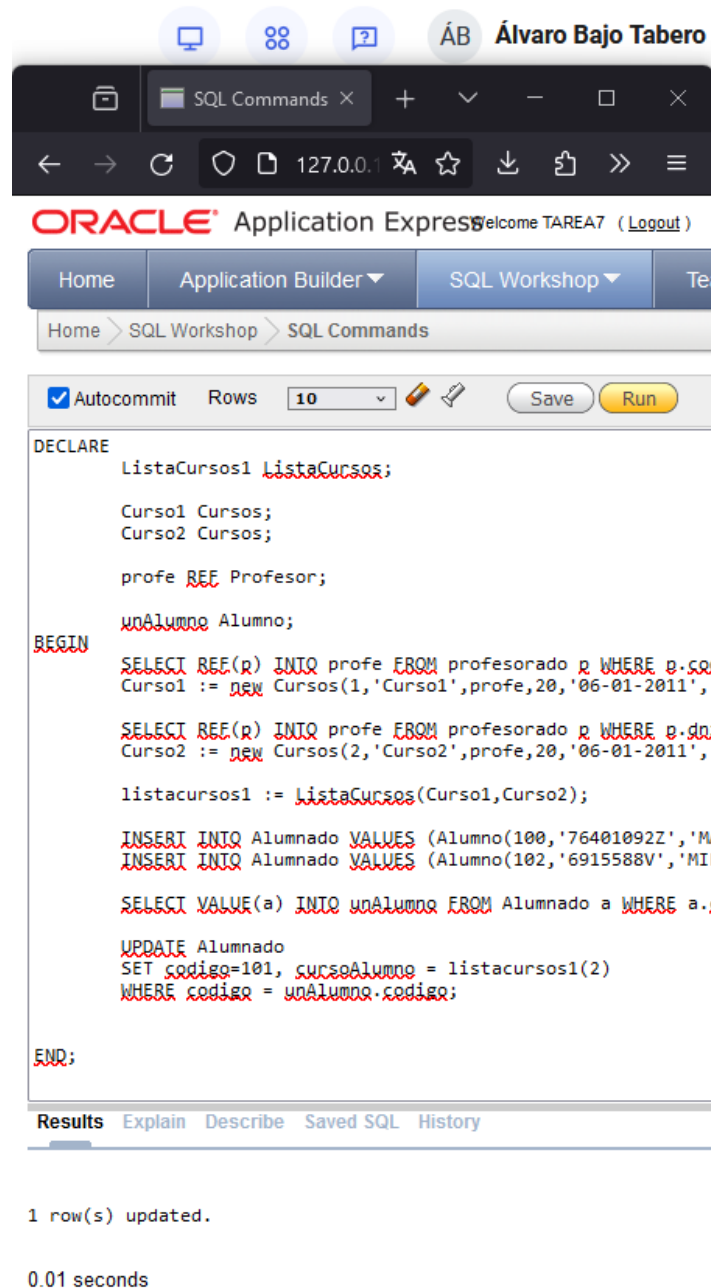
b) Inserta en la tabla "**Alumnado**" las siguientes filas:

```
codigo: 100
dni: 76401092Z
nombre: MANUEL
apellidos: SUAREZ IBAÑEZ
sexo: M
fecha_nac: 30/6/1990
cursoAlumno: objeto creado anteriormente para el primer curso
codigo: 102
dni: 6915588V
nombre: MILAGROSA
apellidos: DIAZ PEREZ
sexo: F
fecha_nac: 28/10/1984
cursoAlumno: objeto que se encuentre en la segunda posición de "listaCursos1"
```

(debe tomarse de la lista)

c) Obtener, de la tabla "Alumnado", el alumno que tiene el código 100, asignándoselo a una variable "unAlumno".

d) Modifica el código del alumno guardado en esa variable "unAlumno" asignando el valor 101, y su curso debe ser el segundo que se había creado anteriormente. Inserta ese alumno en la tabla "Alumnado".



The screenshot shows the Oracle Application Express interface. At the top, there's a user profile for 'ÁB Álvaro Bajo Tabero'. Below it is a navigation bar with 'Home', 'Application Builder', 'SQL Workshop', and 'Tools'. The 'SQL Workshop' tab is active, and the 'SQL Commands' page is open. The interface includes a toolbar with 'Autocommit' checked, 'Rows' set to 10, and 'Save' and 'Run' buttons. The main area contains a PL/SQL script with the following code:


```

DECLARE
    ListaCursos1 ListaCursos;
    Curso1 Cursos;
    Curso2 Cursos;
    profe REF Profesor;
    unAlumno Alumno;
BEGIN
    SELECT REF(r) INTO profe FROM profesorado p WHERE p.codigo = 100;
    Curso1 := new Cursos(1, 'Curso1', profe, 20, '06-01-2011', 'M');
    Curso2 := new Cursos(2, 'Curso2', profe, 20, '06-01-2011', 'M');
    listacursos1 := ListaCursos(Curso1, Curso2);
    INSERT INTO Alumnado VALUES (Alumno(100, '76401092Z', 'M', '06-01-2011', 'M'));
    INSERT INTO Alumnado VALUES (Alumno(102, '6915588V', 'M', '06-01-2011', 'M'));
    SELECT VALUE(a) INTO unAlumno FROM Alumnado a WHERE a.codigo = 100;
    UPDATE Alumnado
    SET codigo=101, cursoAlumno = listacursos1(2)
    WHERE codigo = unAlumno.codigo;
END;

```

Below the script, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the output: '1 row(s) updated.' and '0.01 seconds'.

10. Realiza una consulta de la tabla "Alumnado" ordenada por "cursoAlumno" para comprobar el funcionamiento del método MAP



```
1  -- Consulta de la tabla "Alumnado" ordenada por  
   "cursoAlumno"  
2  --Álvaro Bajo Tabero  
3  SELECT * FROM Alumnado ORDER BY cursoAlumno;
```

Este comando no me funciona en la aplicación. Pero creo que este comando esta bien.