

Matisse

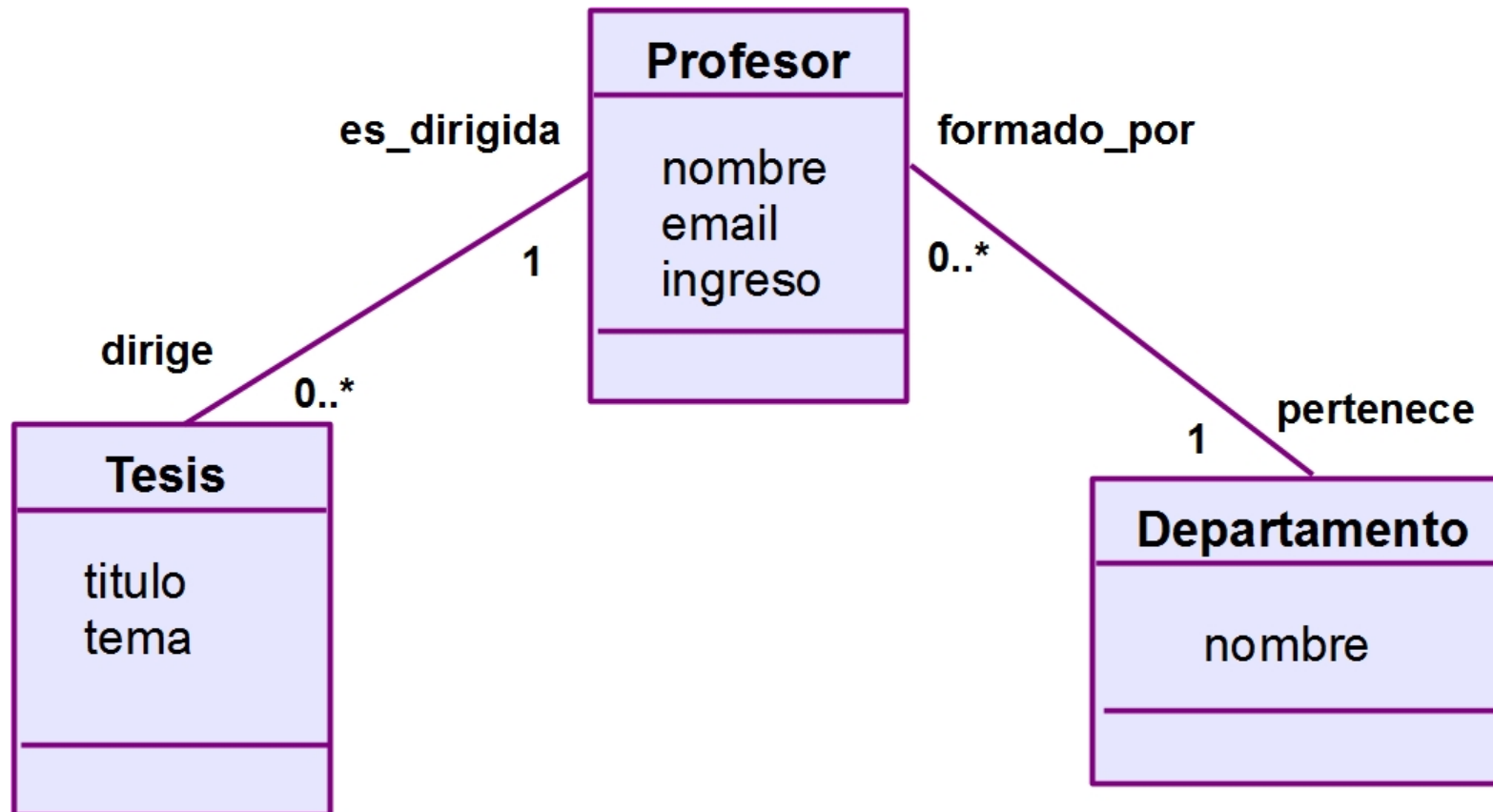
Creación de la BD00 doctorado

La base de objetos Doctorado

Doctorado será una **base de objetos** que permitirá almacenar información sobre profesores, su departamento y las tesis que dirigen.

- Un profesor puede dirigir varias tesis y pertenece a un solo departamento
- Una tesis es dirigida por un único profesor
- Un departamento está formado por varios profesores

Las clases necesarias son las siguientes: **Profesor**, **Tesis** y **Departamento**



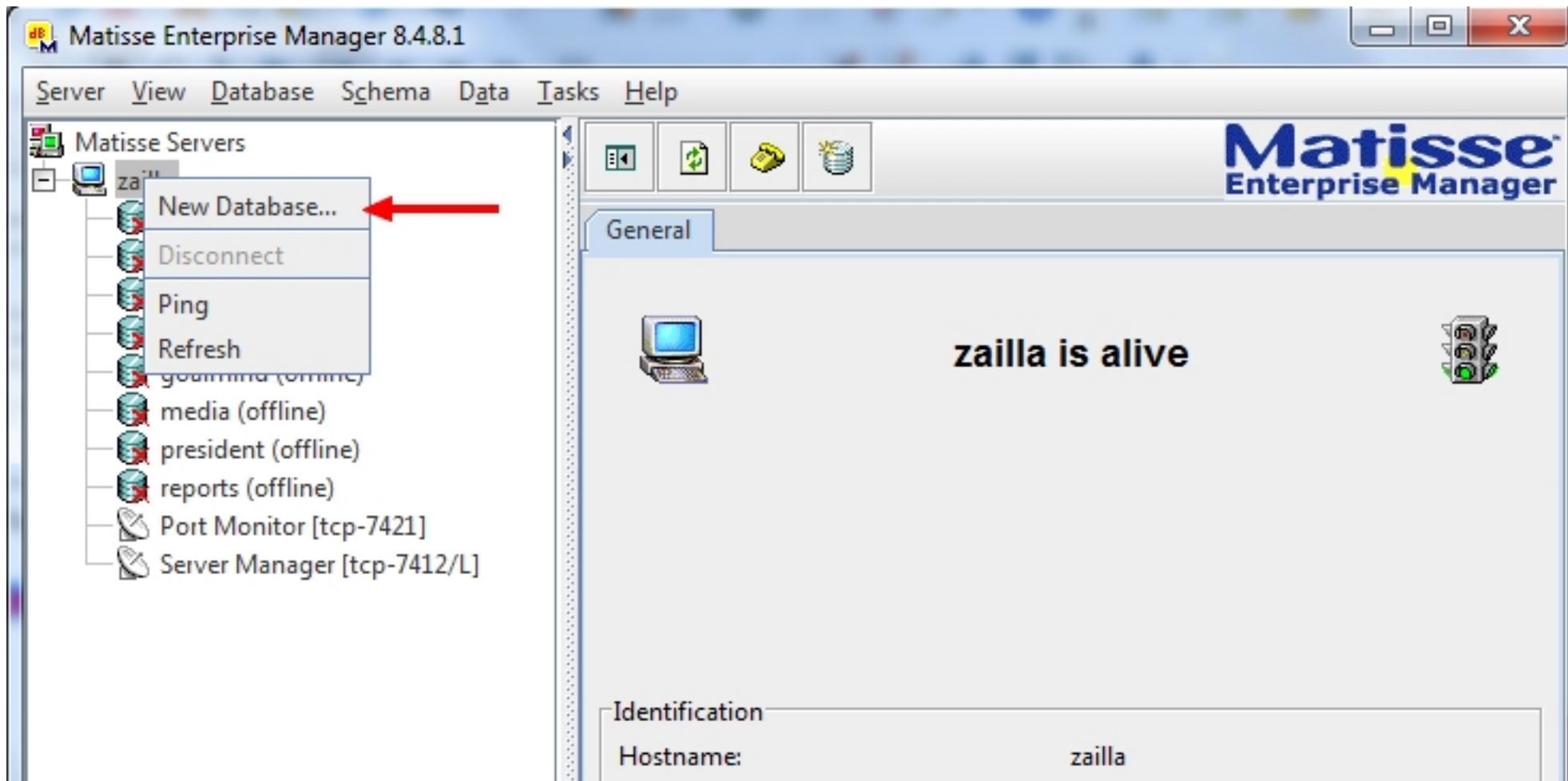
Lo que haremos

Seguiremos los siguientes pasos:

- En Matisse, crearemos una nueva **base de objetos** de nombre **doctorado**.
- En Matisse, **importaremos el esquema ODL** (fichero [doctorado.odl](#)) con las clases Profesor, Tesis y Departamento a la base de objetos doctorado. Este fichero te lo descargarás junto al proyecto y lo ubicarás en una carpeta de tu equipo.
- En NetBeans, crearemos el **proyecto DoctoradoMatisse**
- Agregaremos el **driver de matisse**, [matisse.jar](#), a las bibliotecas del proyecto.
- En el método main() indicaremos la **conexión a la base de objetos** doctorado.
- Desde Matisse **generaremos el código java que implementa las clases** Profesor, Tesis y Departamento en nuestro proyecto.
- En el método main() incluiremos llamadas a los métodos que permiten **insertar objetos en la base de objetos** y almacenarlos al ejecutar la aplicación.

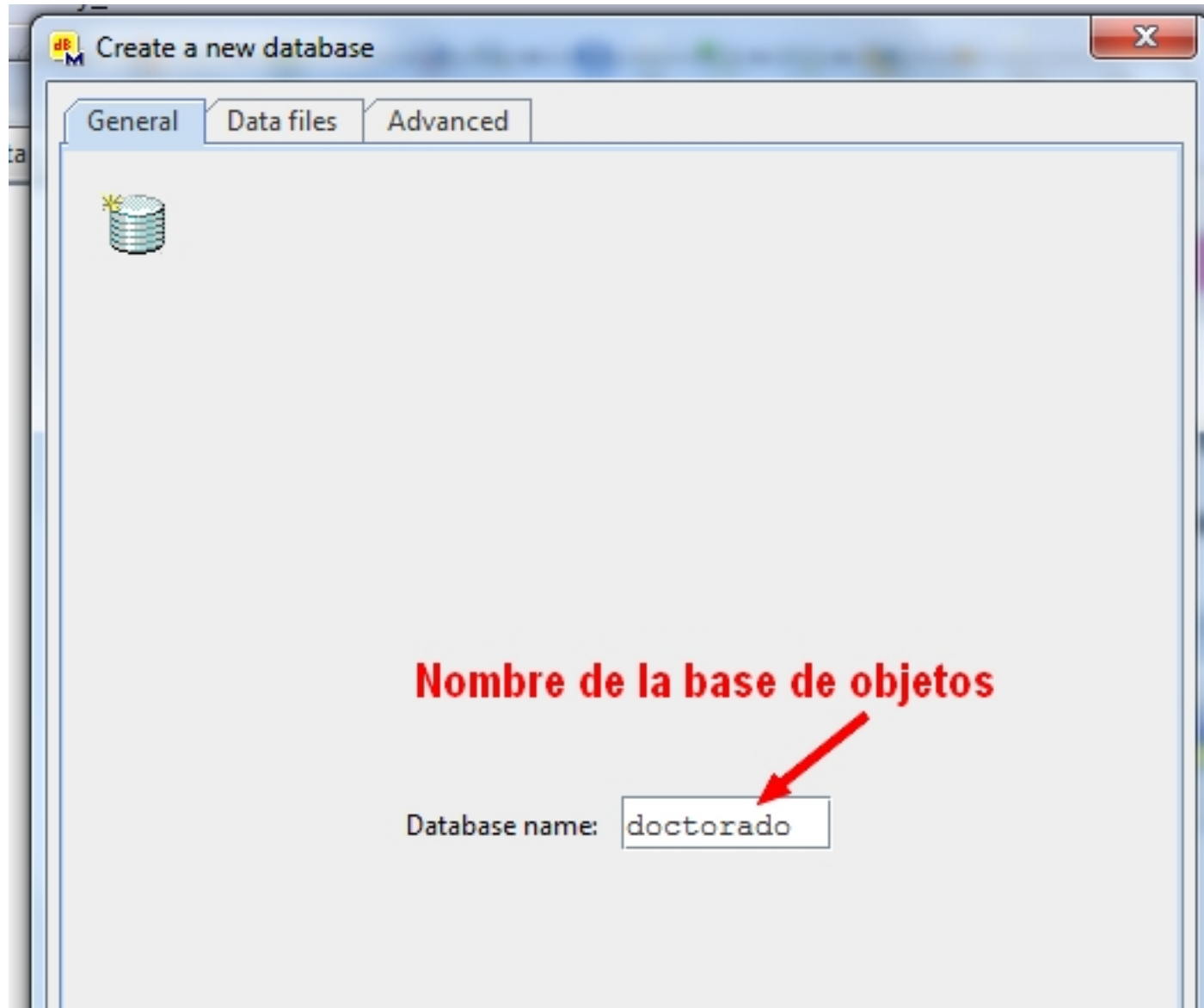
Desde Matisse crear base de objetos

- Desde **Inicio/Programas** seleccionamos **Matisse/Enterprise Manager**
- Nos situamos sobre el servidor en marcha (será nuestro equipo)
- Pulsamos el botón derecho del ratón y seleccionamos **New Database**



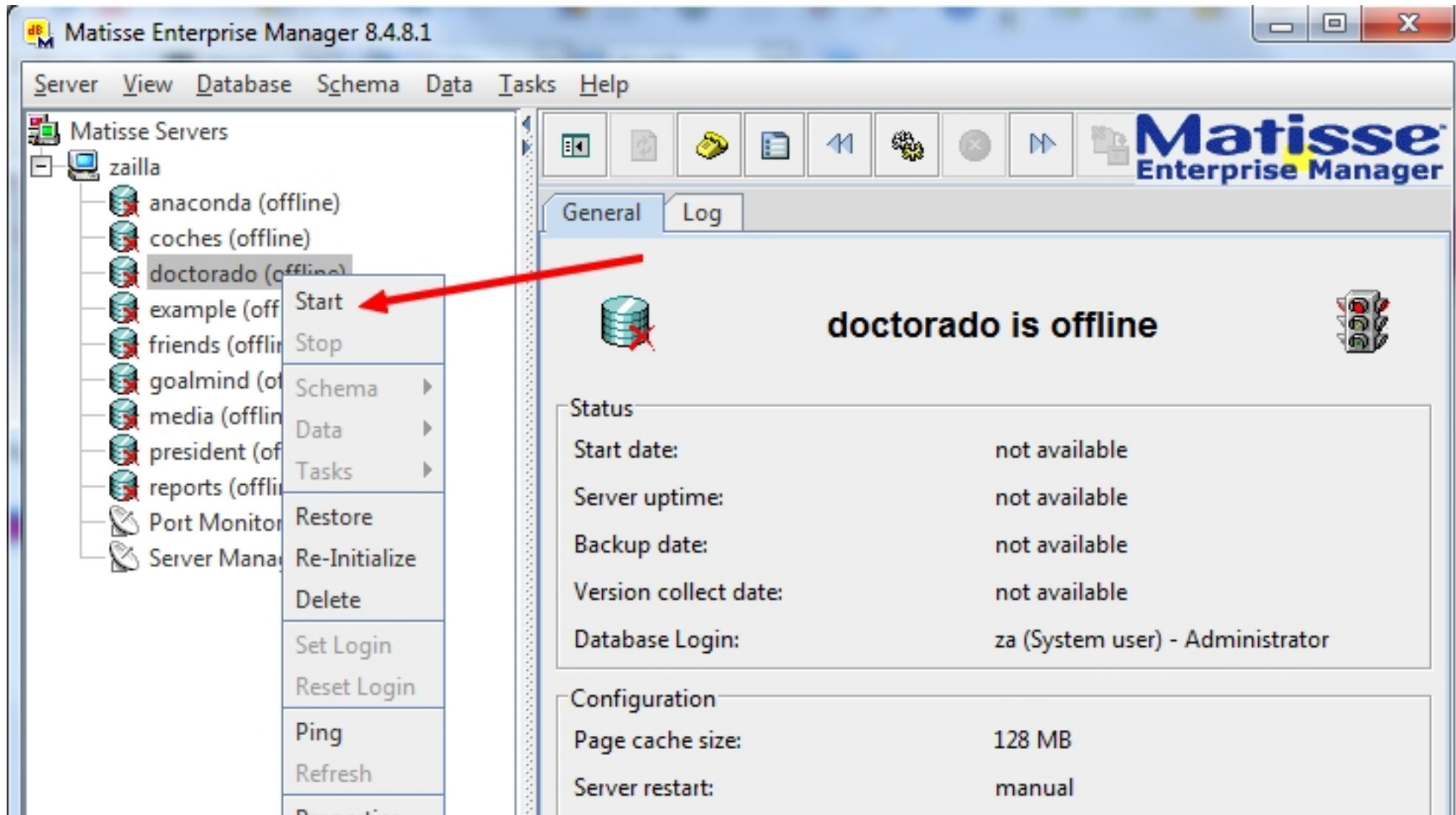
Damos nombre a la base de objetos

- Introducimos el nombre de la base de objetos **doctorado**
- Pulsamos el **botón OK**



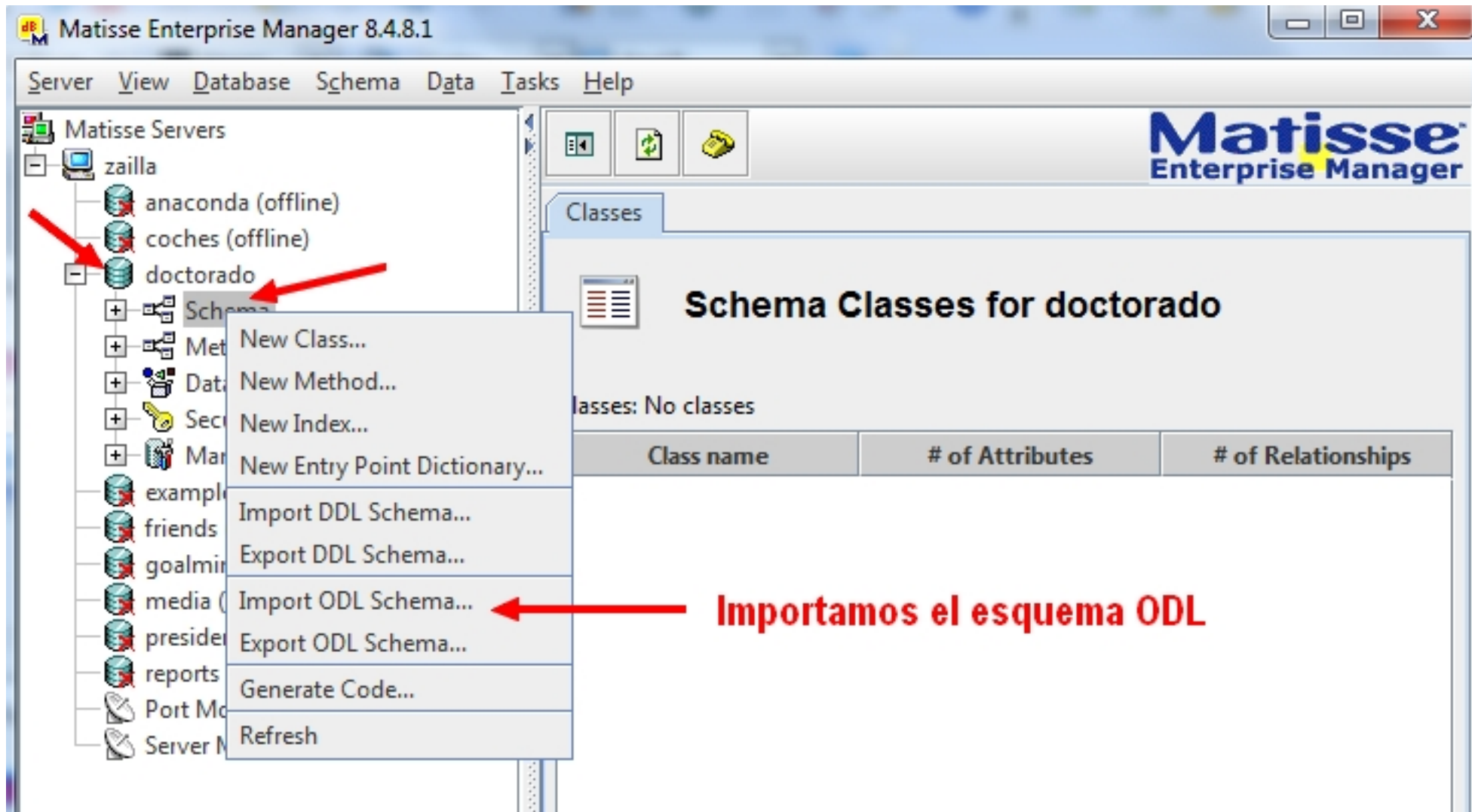
Iniciar la base de objetos en el servidor

- Desde el Enterprise Manager nos situamos sobre la base de objetos recién creada, doctorado
- Pulsamos el **botón derecho** del ratón y seleccionamos **start**. Esto iniciará o pondrá en línea la base de objetos doctorado.



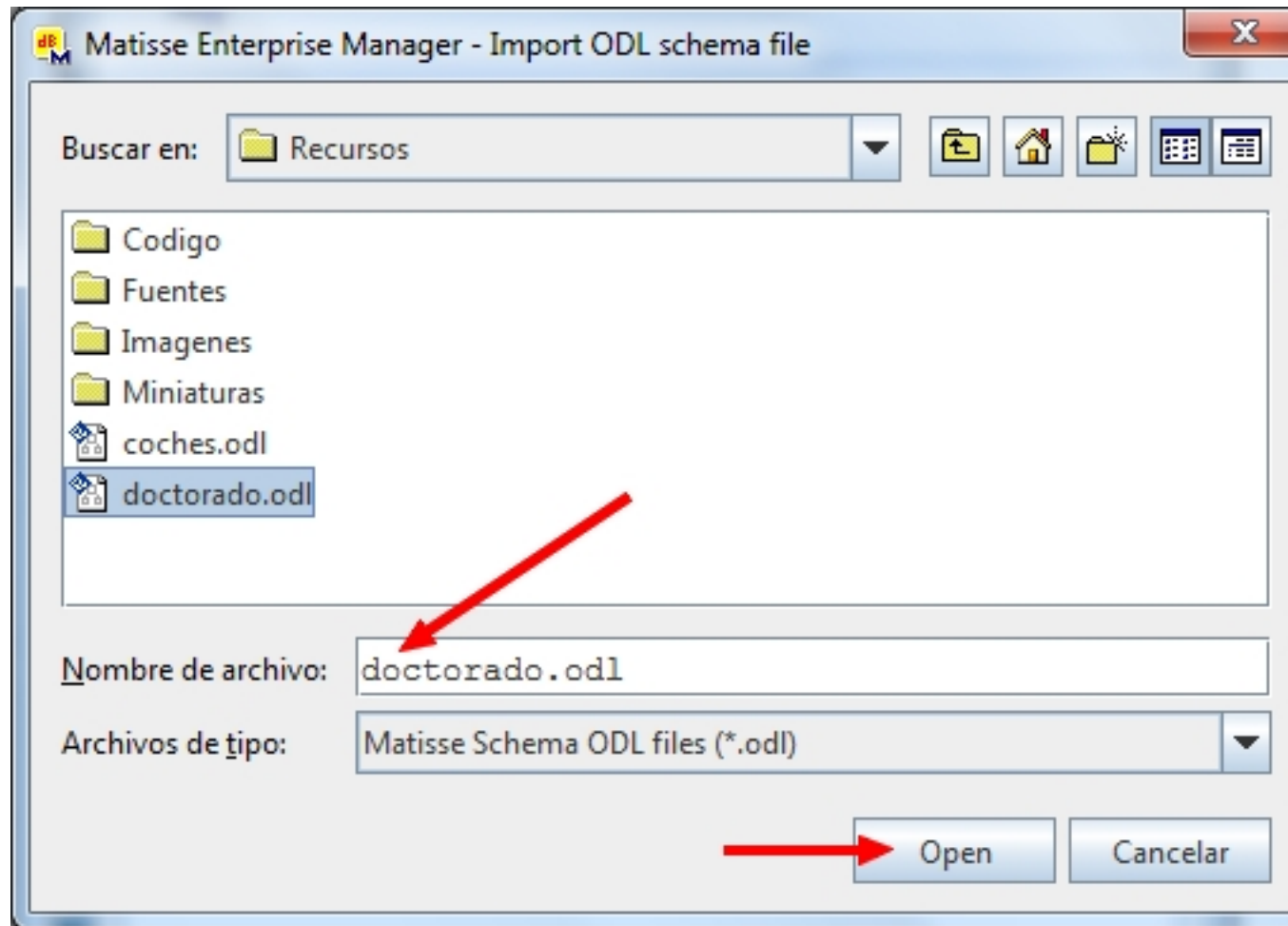
Importar esquema ODL

- Nos situamos sobre la base de objetos **doctorado** y hacemos **doble click**
- Nos situamos sobre **Schema**
- Pulsamos el **botón derecho** del ratón y seleccionamos **Import ODL Schema**.



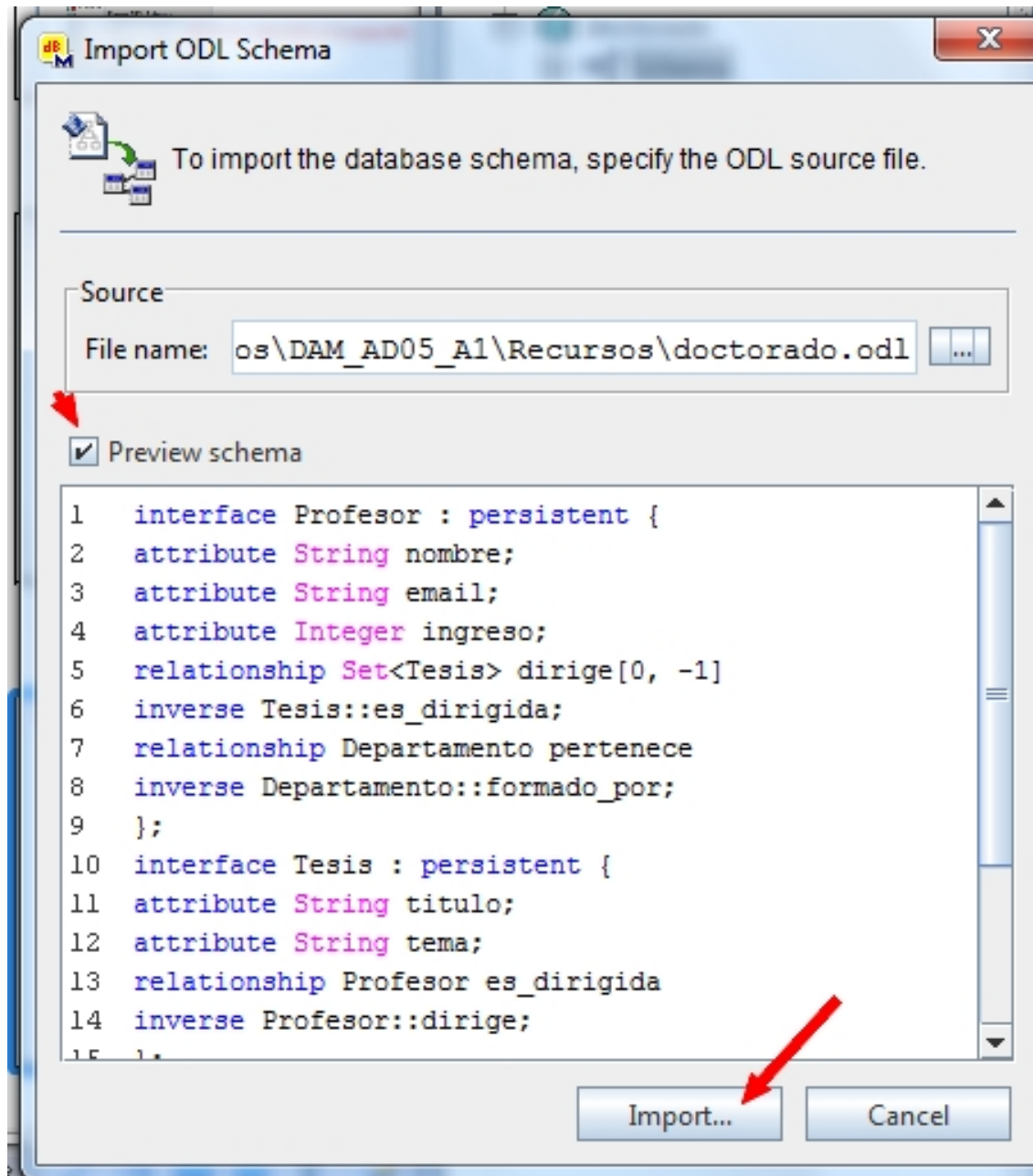
Importando esquema ODL , archivo .odl

- Buscamos el fichero **doctorado.odl** en nuestro equipo y pulsamos **Open**



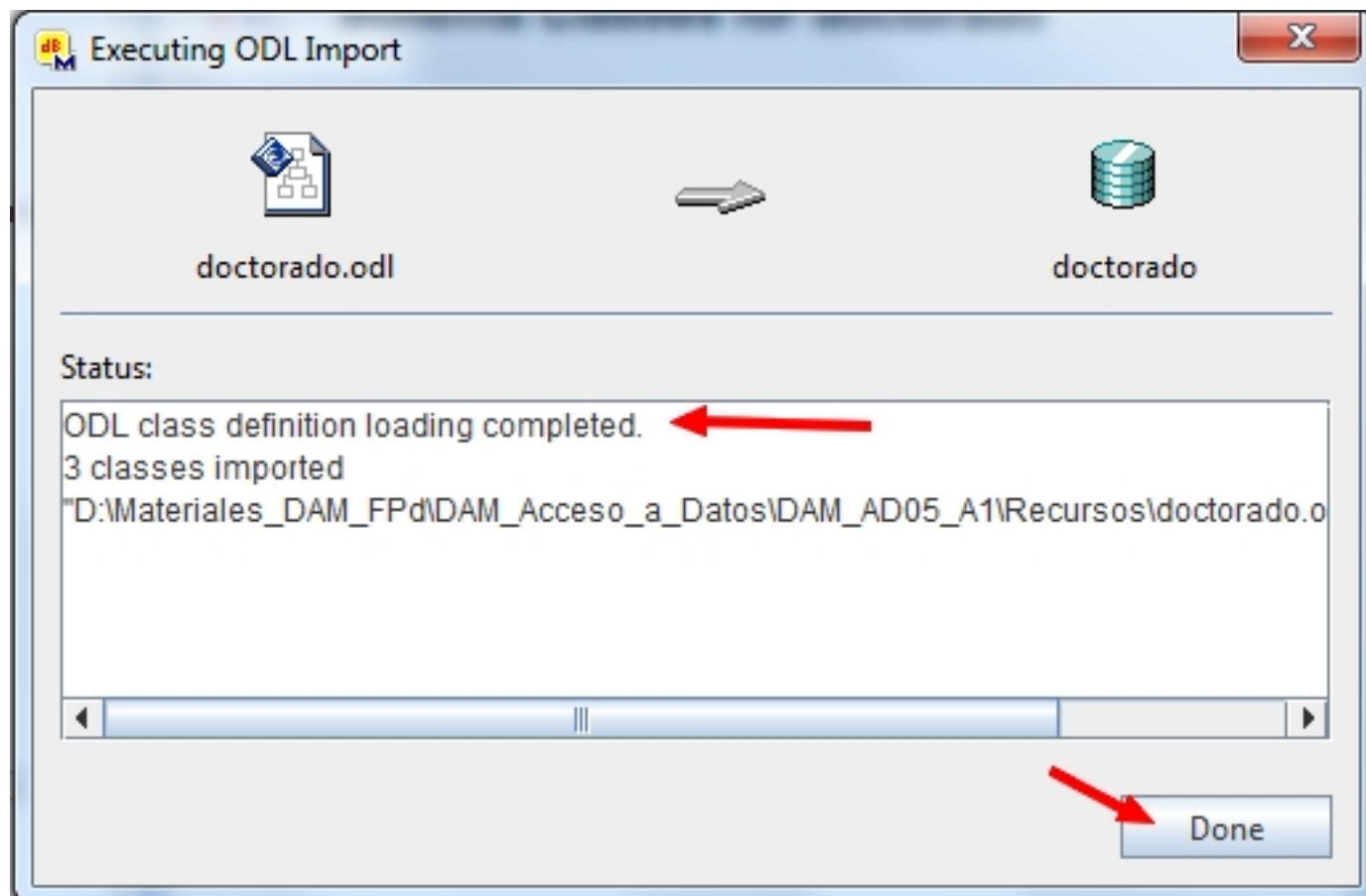
Confirmar importación esquema ODL

- En la siguiente pantalla, pulsamos **Import..** para confirmar la importación.



Importación ODL realizada

- La siguiente pantalla, nos informa del estado de la importación. Pulsamos **Done**.



Navegando por las clases

- Desde Matisse(Enterprise Manager) podemos navegar por las clases haciendo doble click sobre cada una de ellas y observar los atributos e interrelaciones.

The screenshot shows the Matisse Enterprise Manager 8.4.8.1 interface. On the left, a tree view displays the database structure under 'Matisse Servers' > 'zailla'. The tree includes 'coches (offline)', 'doctorado', 'Schema', 'Classes', 'Departamento', 'Profesor', and 'Tesis'. Red arrows point to the 'Classes' folder and its sub-entities: 'Departamento', 'Profesor', and 'Tesis'. The right pane shows the 'General' tab for the selected entity, displaying 'zailla is alive' with a computer icon and a traffic light icon. Below this, a table lists identification details for the 'zailla' host.

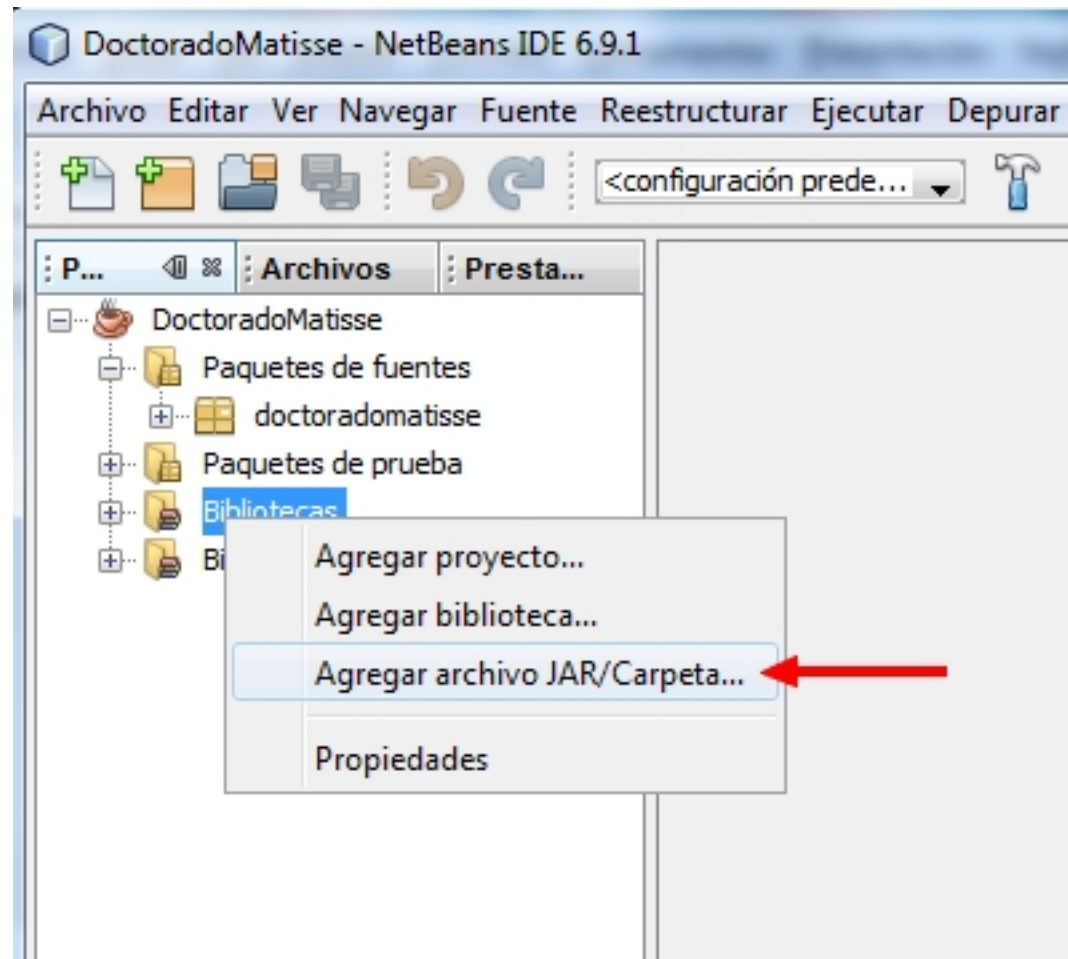
Identification	
Hostname:	zailla
IP address:	192.168.1.103
Login name:	za

Red arrows and text labels highlight specific relationships in the 'Profesor' entity details:

- Interrelación inversa formado_por**: Points to the 'formado_por (Profesor, [0..n])' relationship.
- Interrelación dirige**: Points to the 'dirige (Tesis, [0..n])' relationship.
- Interrelación pertenece**: Points to the 'pertenece (Departamento)' relationship.

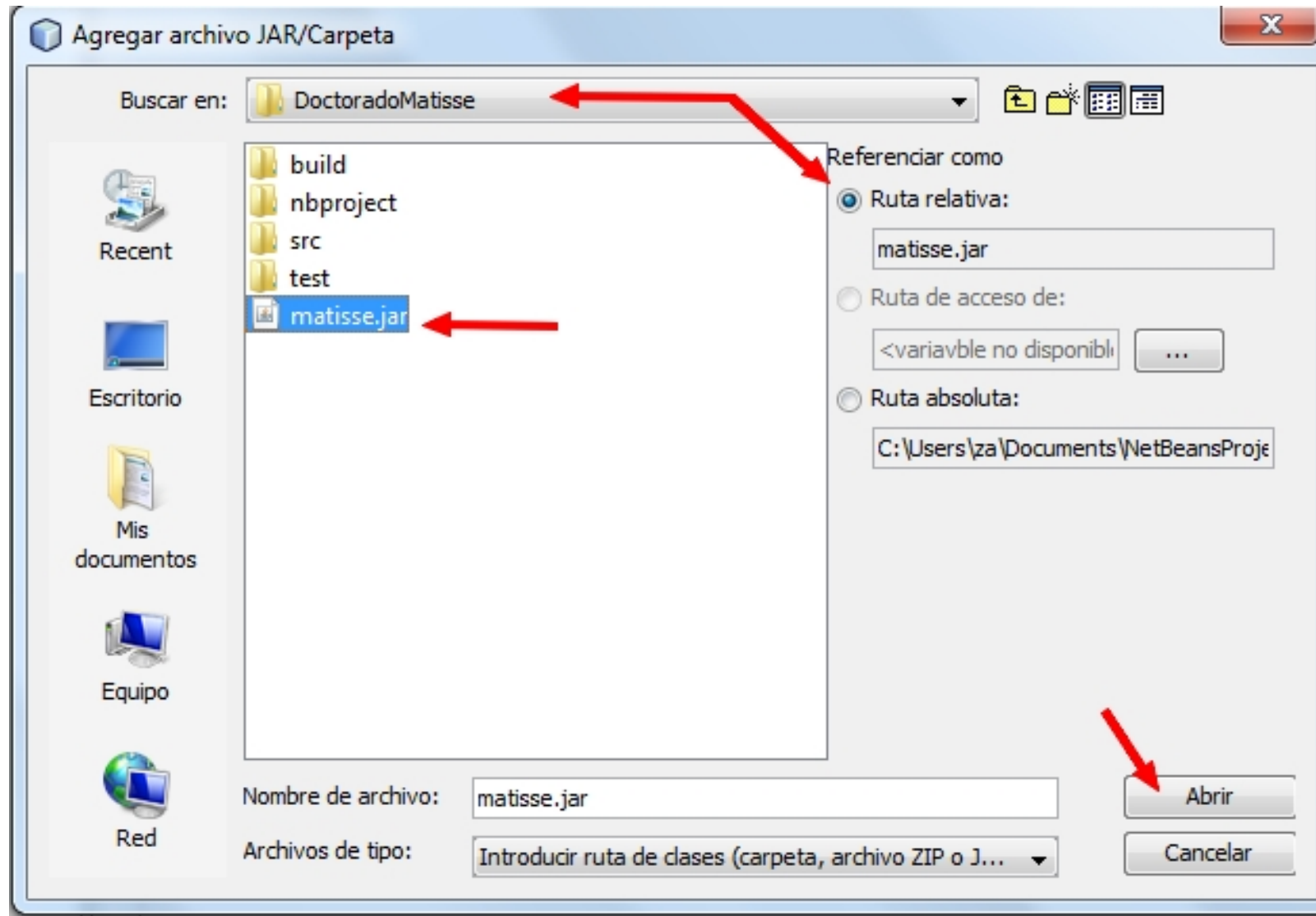
Crear el proyecto y agregar API matisse

- En NetBeans creamos el **proyecto de nombre DoctoradoMatisse**
- Nos situamos sobre el proyecto, en el **nodo Bibliotecas**.
- Hacemos un click derecho sobre Bibliotecas
- Ejecutamos la **opción 'Agregar archivo JAR/Carpeta...'**



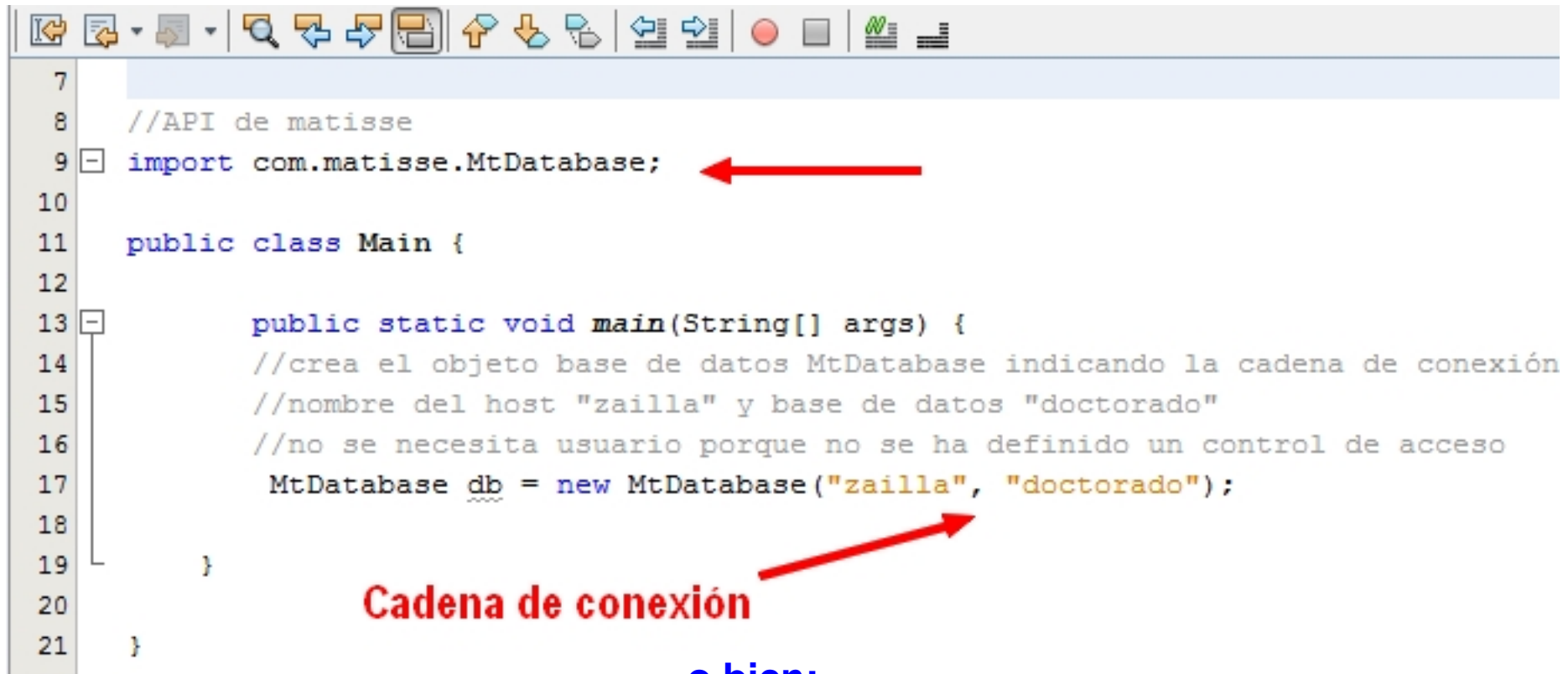
Agregando API de Matisse

- Seleccionamos el JAR de nuestra carpeta local, archivo **matisse.jar**
- Si **matisse.jar** lo hemos ubicado en el raíz de nuestro proyecto, seleccionamos Ruta relativa.
- Pulsamos el **botón Abrir**



Indicar cadena de conexión en main()

- En nuestra clase principal, la de main(), importamos la biblioteca de matisse **com.matisse. MtDatabase**
- En el método main() creamos un objeto **MtBasedata** indicando **la cadena de conexión** (“nombrehost”, “nombrebaseobjetos”)



```
7
8 //API de matisse
9 import com.matisse.MtDatabase;
10
11 public class Main {
12
13     public static void main(String[] args) {
14         //crea el objeto base de datos MtDatabase indicando la cadena de conexión
15         //nombre del host "zailla" y base de datos "doctorado"
16         //no se necesita usuario porque no se ha definido un control de acceso
17         MtDatabase db = new MtDatabase("zailla", "doctorado");
18     }
19 }
20
21 }
```

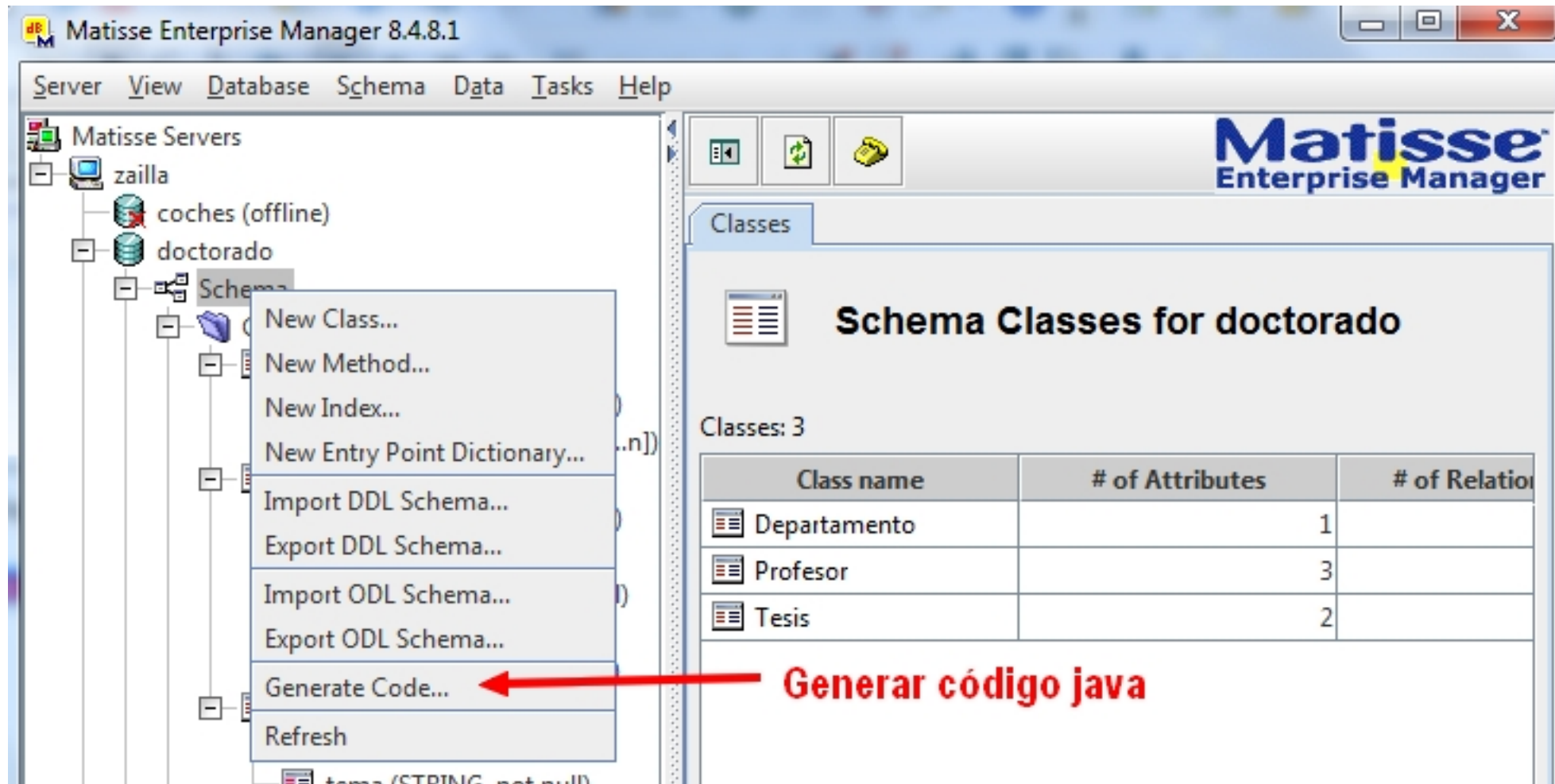
Cadena de conexión

o bien:

Cadena conexión (localhost, “docctorado”)

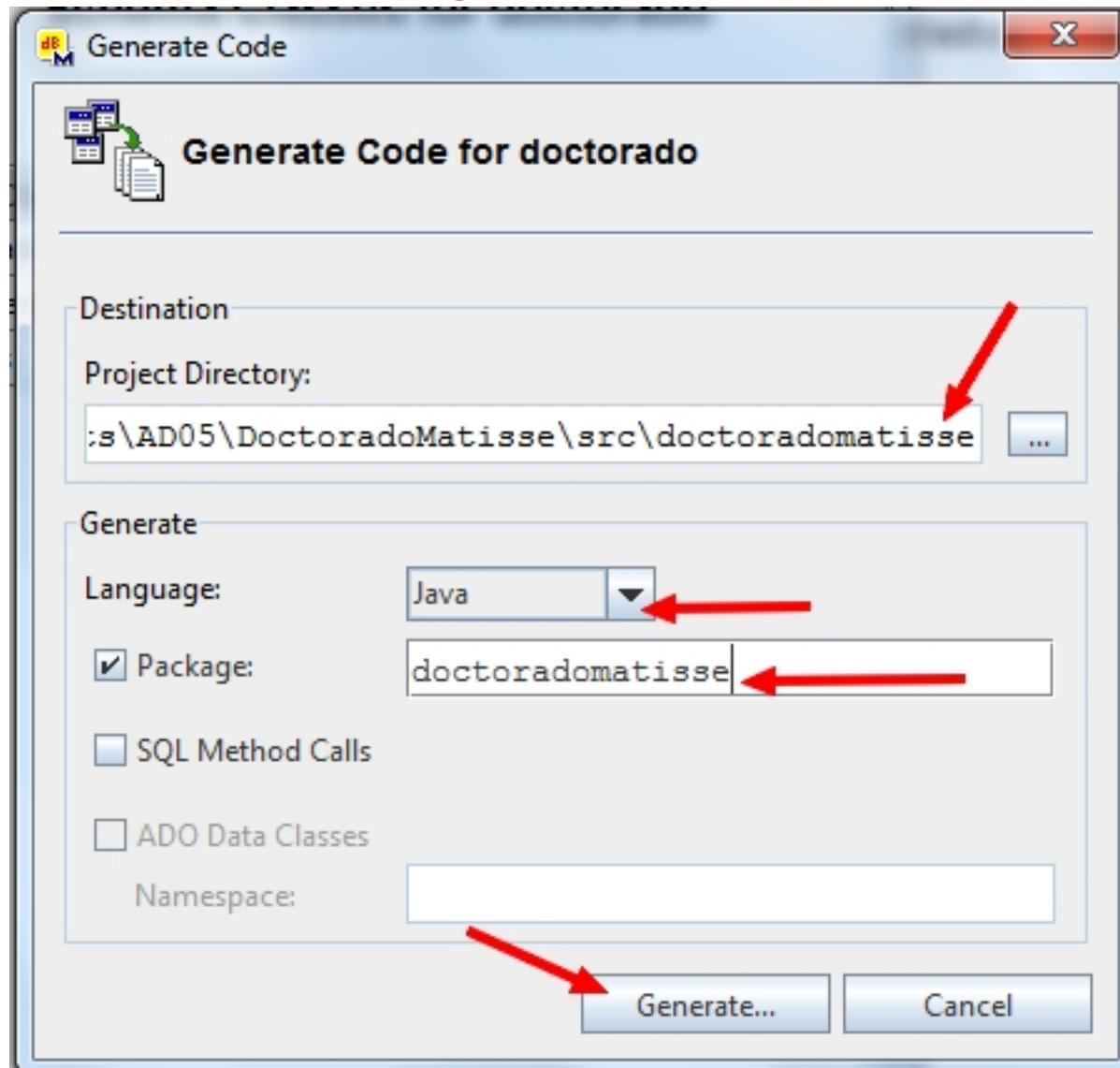
Desde Matisse generamos código java

- En **Matisse**, nos situamos sobre **Schema** de la base de objetos **doctorado**, y en su menú contextual seleccionamos **Generate Code...**



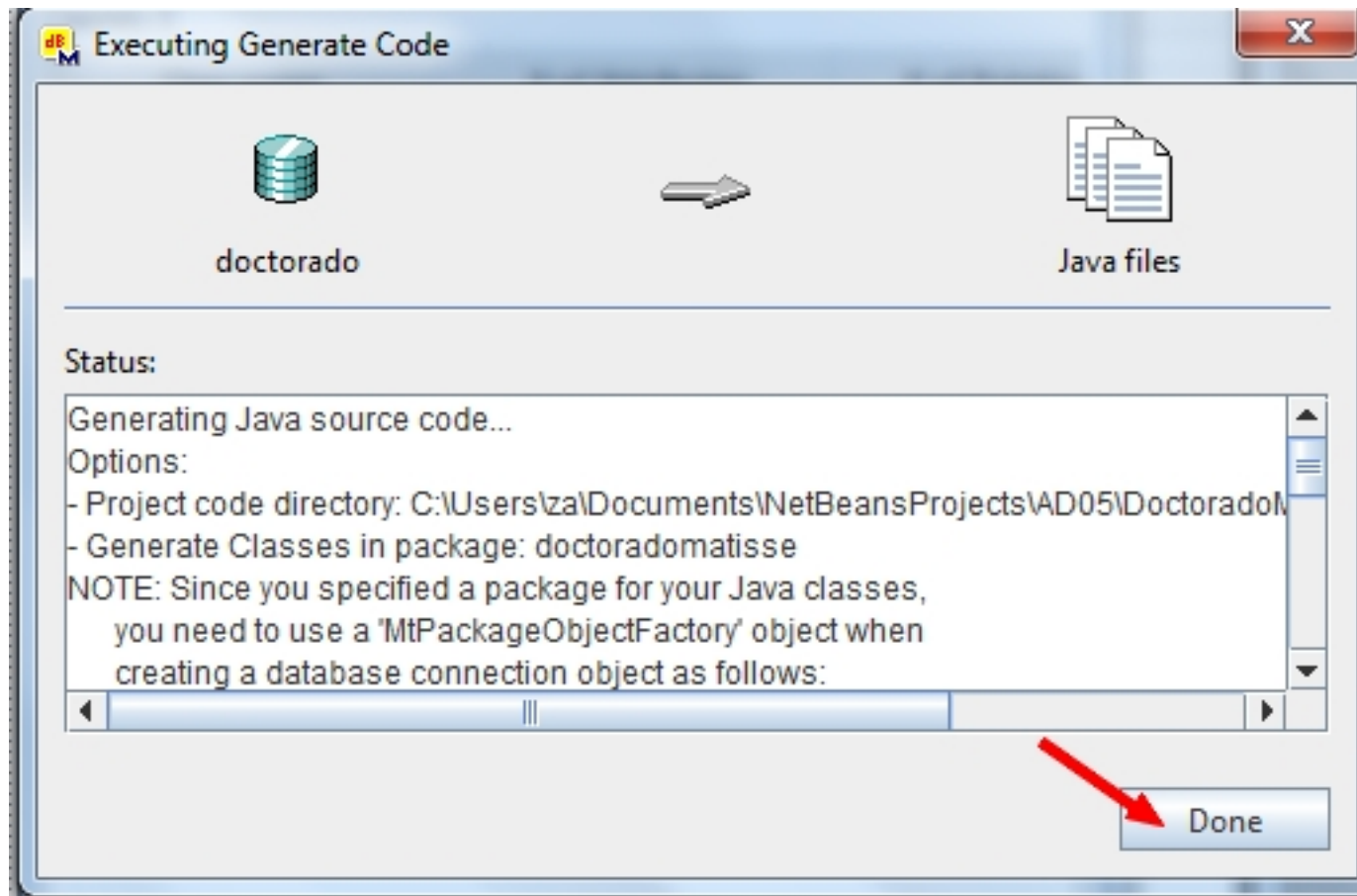
Indicamos carpeta destino y lenguaje java

- Buscamos el directorio de nuestro proyecto DoctoradoMatisse, carpeta **src\doctoradomatisse**
- Indicamos que el lenguaje es **Java**
- Indicamos el nombre del **package** que es **doctoradomatisse**.



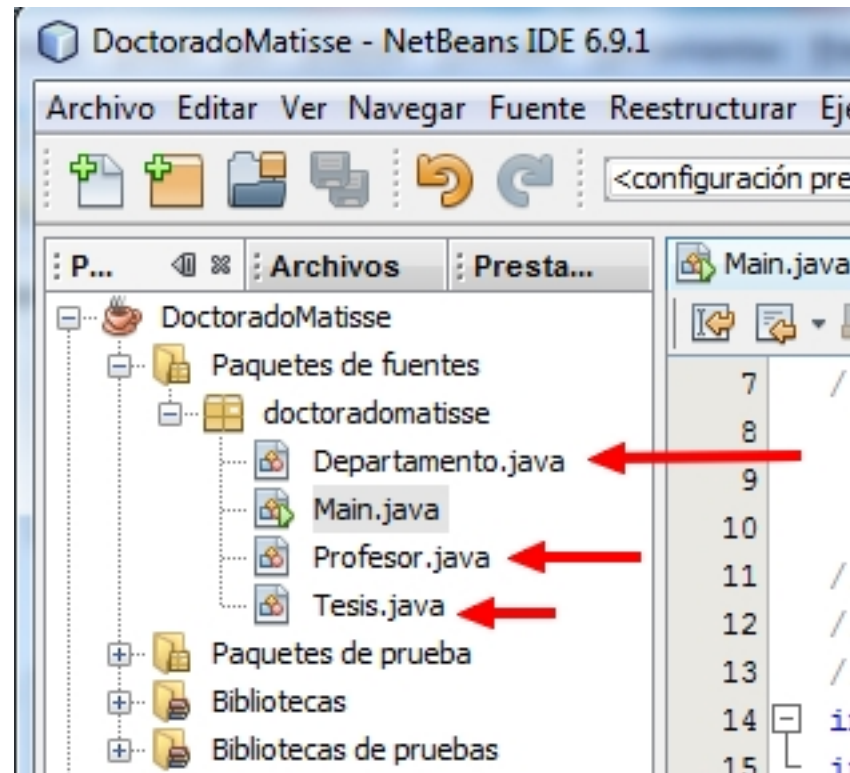
Indicamos carpeta destino y lenguaje java

- Una vez generado el código, pulsamos **Done**



Código generado en proyecto DoctoradoMatisse

- Desde NetBeans, podemos ver que se han generado las clases Profesor, Tesis y Departamento.



Operaciones desde main()

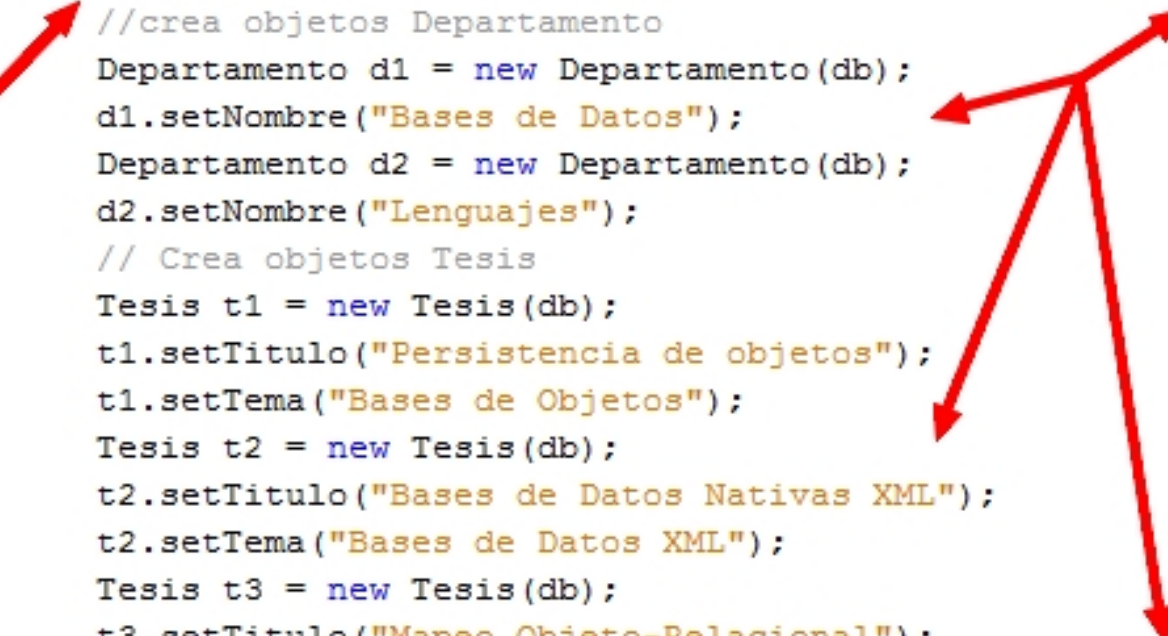
- En el método **main()**, tras la conexión a la base de objetos, en un **bloque try-catch**:
 - Abrimos conexión `db.open()`
 - Invocamos al método para insertar objetos `insertarObjetos()`
 - Cerramos conexión `db.close()`

```
MtDatabase db = new MtDatabase("zailla", "doctorado");
//mensaje para la Salida
System.out.println("===== connectFromMtDatabase =====\n");
try {
    //conecta con la base de datos
    db.open();
    //inicia una transacción (matisse gestiona todas las operaciones como transacciones)
    db.startTransaction();
    //insertar datos en registros
    insertarObjetos(db);
} catch (MtException mte) {
    //mensaje para la Salida
    System.out.println("MtException : " + mte.getMessage());
} finally {
    //confirma cualquier transacción en proceso
    if (db.isTransactionInProgress()) {
        db.commit();
    }
    //cierra la base de datos
    db.close();
}
```

Creación de objetos en clase Main

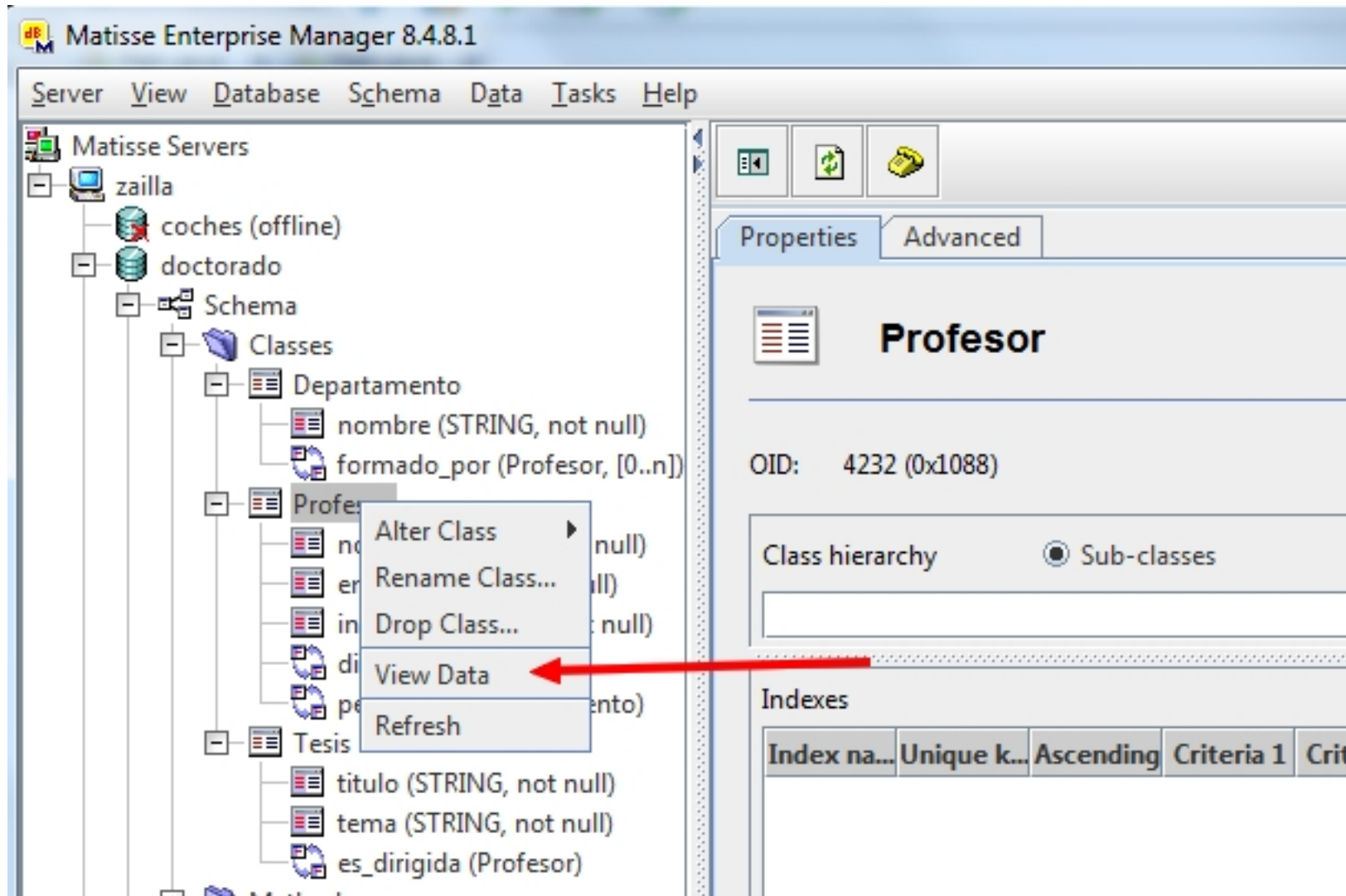
- En la clase Main, creamos el método **insertarObjetos()**, donde crearemos diversos objetos tipo Profesor, Tesis y Departamento.
- Relacionaremos los objetos mediante los **métodos asociados a las interrelaciones establecidas**.
- Todos estos métodos se han generado en la generación de código
- **EJECUTAMOS** y los objetos se almacenarán en la base de objetos

```
private static void insertarObjetos(MtDatabase db) {  
    //crea objetos Departamento  
    Departamento d1 = new Departamento(db);  
    d1.setNombre("Bases de Datos");  
    Departamento d2 = new Departamento(db);  
    d2.setNombre("Lenguajes");  
    // Crea objetos Tesis  
    Tesis t1 = new Tesis(db);  
    t1.setTitulo("Persistencia de objetos");  
    t1.setTema("Bases de Objetos");  
    Tesis t2 = new Tesis(db);  
    t2.setTitulo("Bases de Datos Nativas XML");  
    t2.setTema("Bases de Datos XML");  
    Tesis t3 = new Tesis(db);  
    t3.setTitulo("Mapeo Objeto-Relacional");  
    t3.setTema("Bases de Datos");  
    Tesis t4 = new Tesis(db);  
    t4.setTitulo("Multiproceso en Java");  
    t4.setTema("Lenguajes de Programación");  
  
    Profesor p1 = new Profesor(db);  
    p1.setNombre("Ana Martos Gil");  
    p1.setEmail("ana.martos@universi.es");  
    p1.setIngreso(1990);  
    Profesor p2 = new Profesor(db);  
    p2.setNombre("Isabel Ruz Granados");  
    p2.setEmail("isabel.ruz@universi.es");  
    p2.setIngreso(1986);  
    Profesor p3 = new Profesor(db);  
    p3.setNombre("Antonio Barea Navarro");  
    p3.setEmail("antonioa.barea@universi.es");  
    p3.setIngreso(1995);  
    //establece relaciones entre Profesor y Dep  
    //Al establecer una relación no hace falta  
    p1.setPertenece(d1);  
    p1.appendDirige(t1);  
    p2.setPertenece(d1);  
    p2.appendDirige(t2);  
    p2.appendDirige(t3);  
}
```



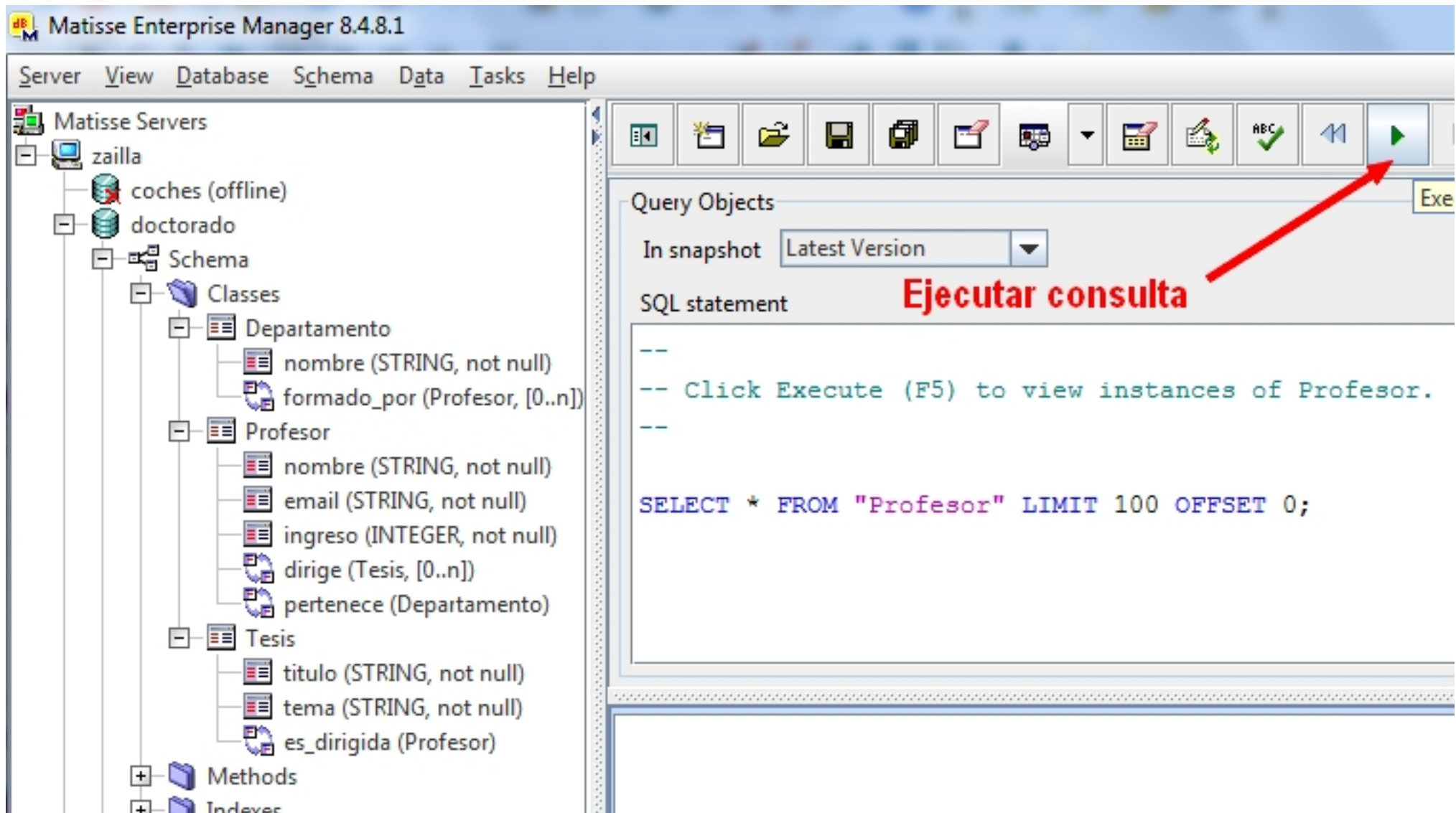
Visualizando objetos en Matisse

- Desde Matisse, nos situamos en una de las clases, por ejemplo **Profesor** y en su menú contextual seleccionamos **View Data**.



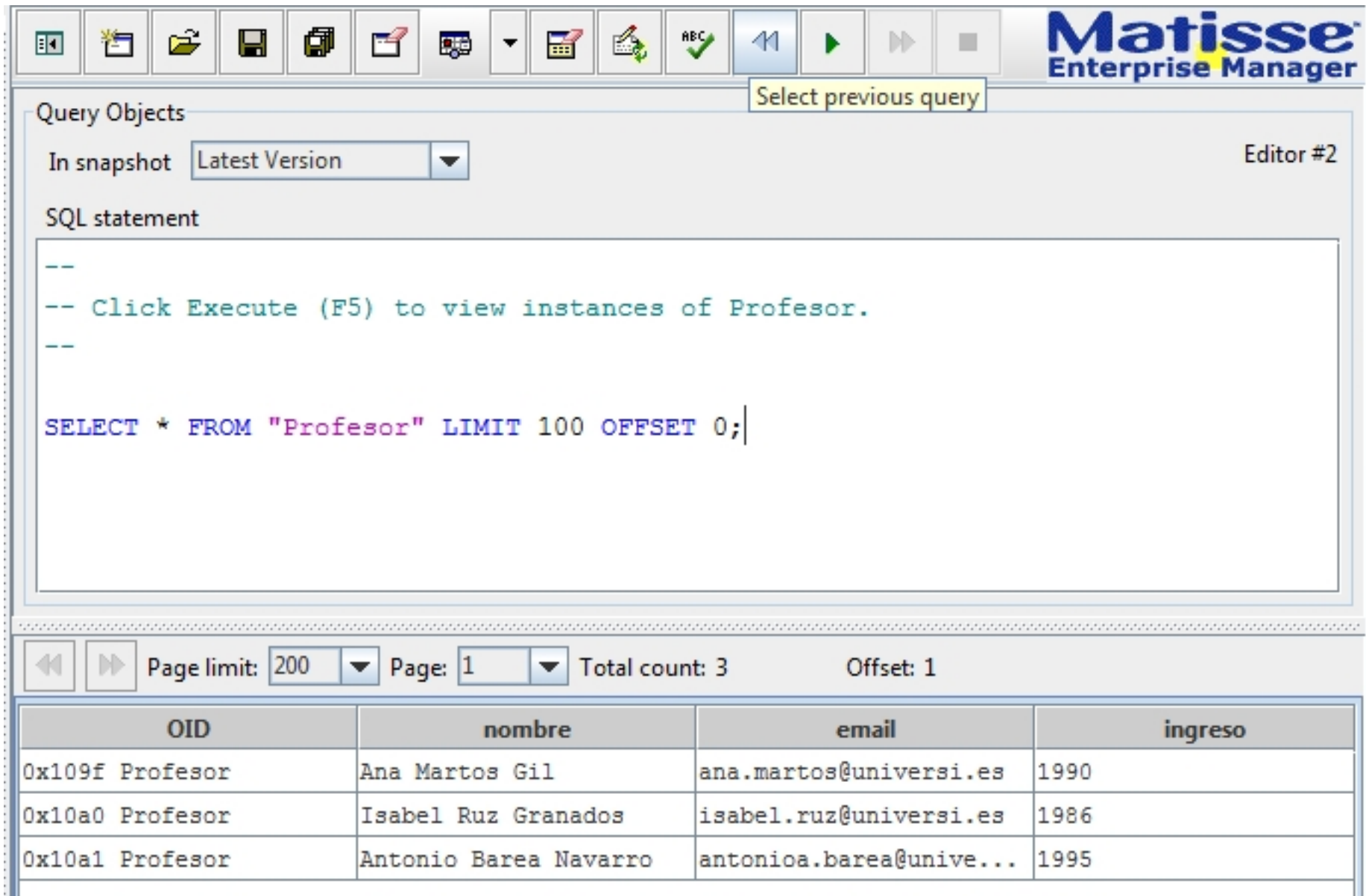
Ejecutando consulta en Matisse

- En la siguiente pantalla, ejecutamos la consulta mediante **F5** o bien botón Execute Query.



Visualizando consulta en Matisse

- En la siguiente pantalla se visualiza el resultado de la consulta



The screenshot displays the Matisse Enterprise Manager interface. At the top, there is a toolbar with various icons for file operations and navigation. The main window is titled "Query Objects" and contains a dropdown menu set to "Latest Version" and a label "Editor #2". Below this, the "SQL statement" section shows the following query:

```
--  
-- Click Execute (F5) to view instances of Profesor.  
--  
  
SELECT * FROM "Profesor" LIMIT 100 OFFSET 0;
```

Below the SQL statement, there is a pagination bar with the following information: "Page limit: 200", "Page: 1", "Total count: 3", and "Offset: 1". At the bottom, a table displays the results of the query:

OID	nombre	email	ingreso
0x109f Profesor	Ana Martos Gil	ana.martos@universi.es	1990
0x10a0 Profesor	Isabel Ruz Granados	isabel.ruz@universi.es	1986
0x10a1 Profesor	Antonio Barea Navarro	antonioa.barea@unive...	1995

Credenciales

Imagen	Datos licencia
Imagen de la diapositiva 2	Autoría: Isabel M. Cruz Granados Licencia: Uso educativo-no comercial. Procedencia: Dibujo realizado por la autora.
Capturas de pantalla de esta presentación EXCEPTO diapositivas , 2, 14, 18, 19 y 20.	Autoría:Matisse Software Inc. Tipo de licencia: Copyright (Cita). Procedencia: Instalación del gestor de objetos Matisse.

Credenciales

Imagen	Datos licencia
Capturas de pantalla de esta de las diapositivas 12, 18 de esta presentación	Autoría: Isabel M. Cruz Granados Licencia: Uso educativo-no comercial. Procedencia: Captura de pantalla del programa NetBeans, propiedad Sun Microsystems, bajo licencia GNU GPL v2
Capturas de pantalla de esta de las diapositivas 14,19, 20 de esta presentación	Autoría: Isabel M. Cruz Granados Licencia: Uso educativo-no comercial. Procedencia: Captura de pantalla del editor de código NetBeans, propiedad Sun Microsystems, bajo licencia GNU GPL v2