**Acceso a datos**

**Tarea Unidad 3**

Contenido

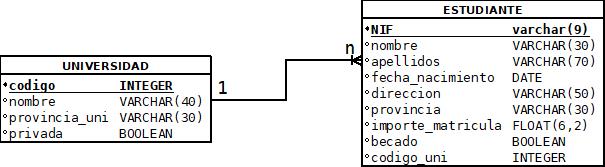
[**INTRODUCCIÓN** 2](#_Toc184669555)

[**EJERCICIO 1** 2](#_Toc184669556)

[**EJERCICIO 2** 2](#_Toc184669557)

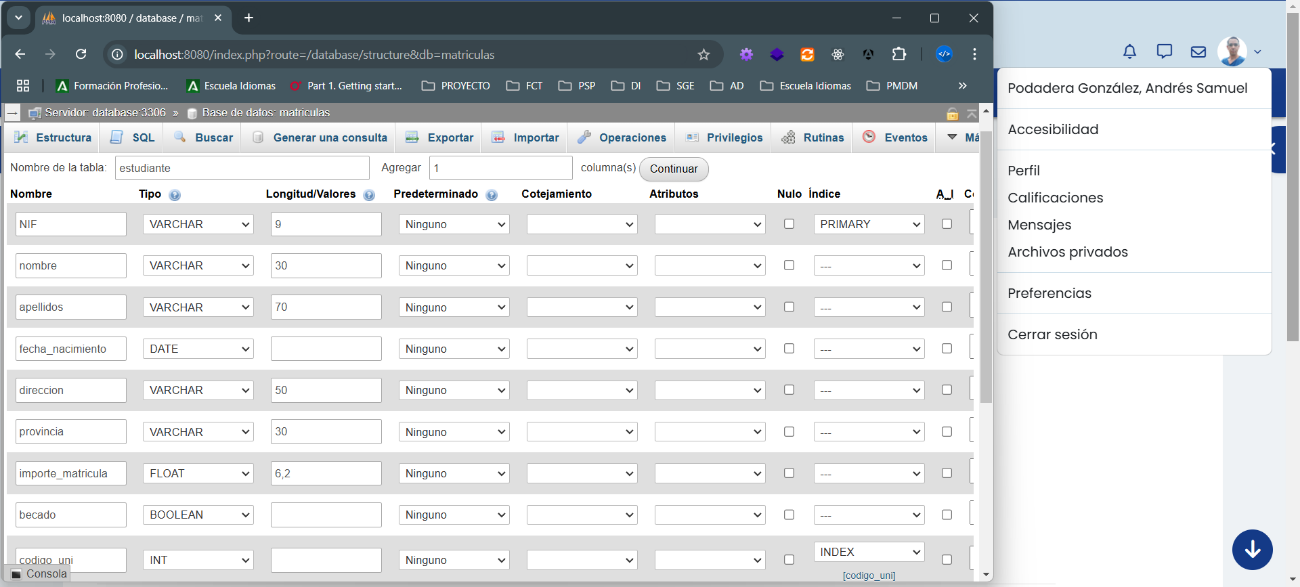
# ***INTRODUCCIÓN***

*Se pide crear una base de datos relacional llamada****matriculas****y formada por las tablas ESTUDIANTE y UNIVERSIDAD, con el siguiente esquema relacional. Se pide entregar el****script SQ****L que permita generar la base de datos.*

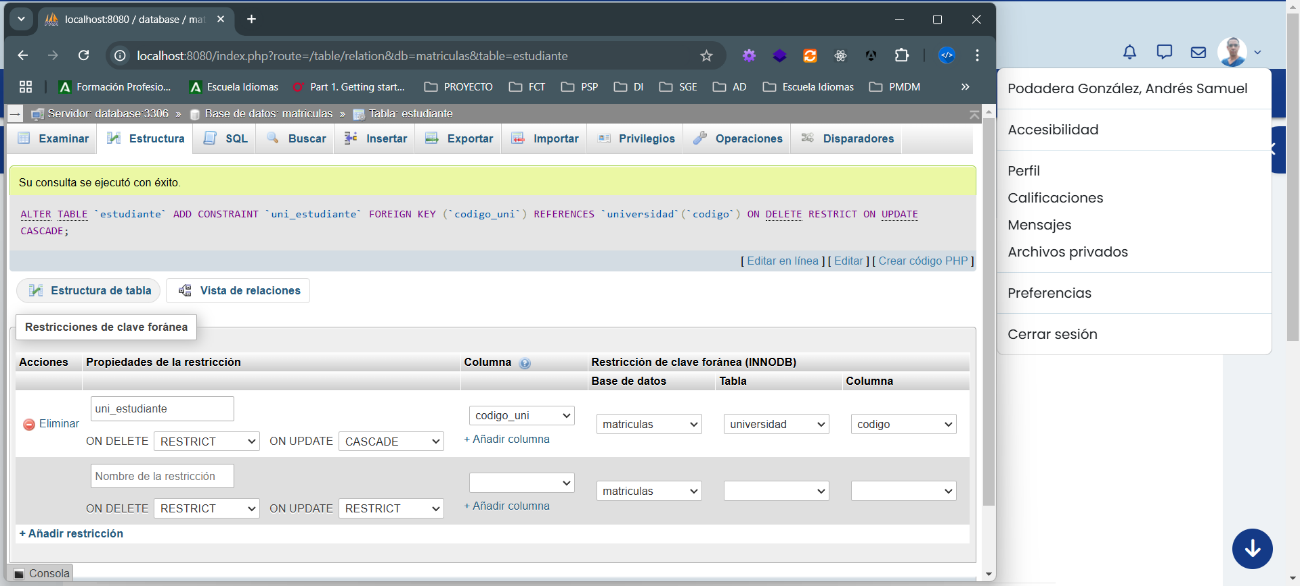
**

**Para crear la base de datos he usado phpmyadmin. Creando primero las tablas y sus campos:**

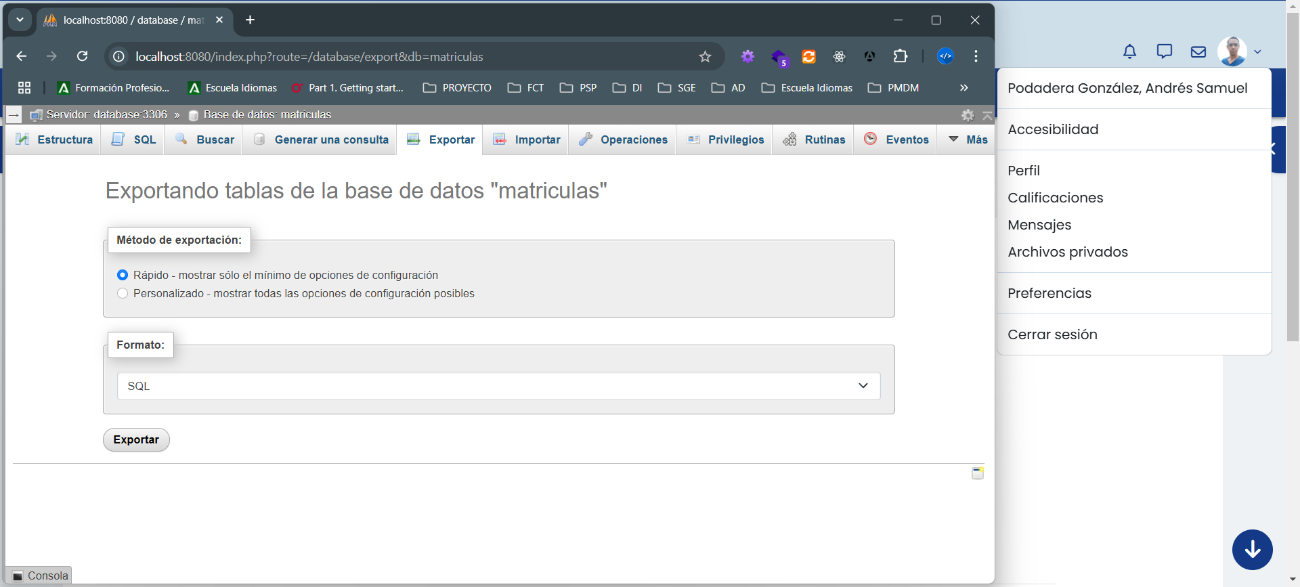
****

****

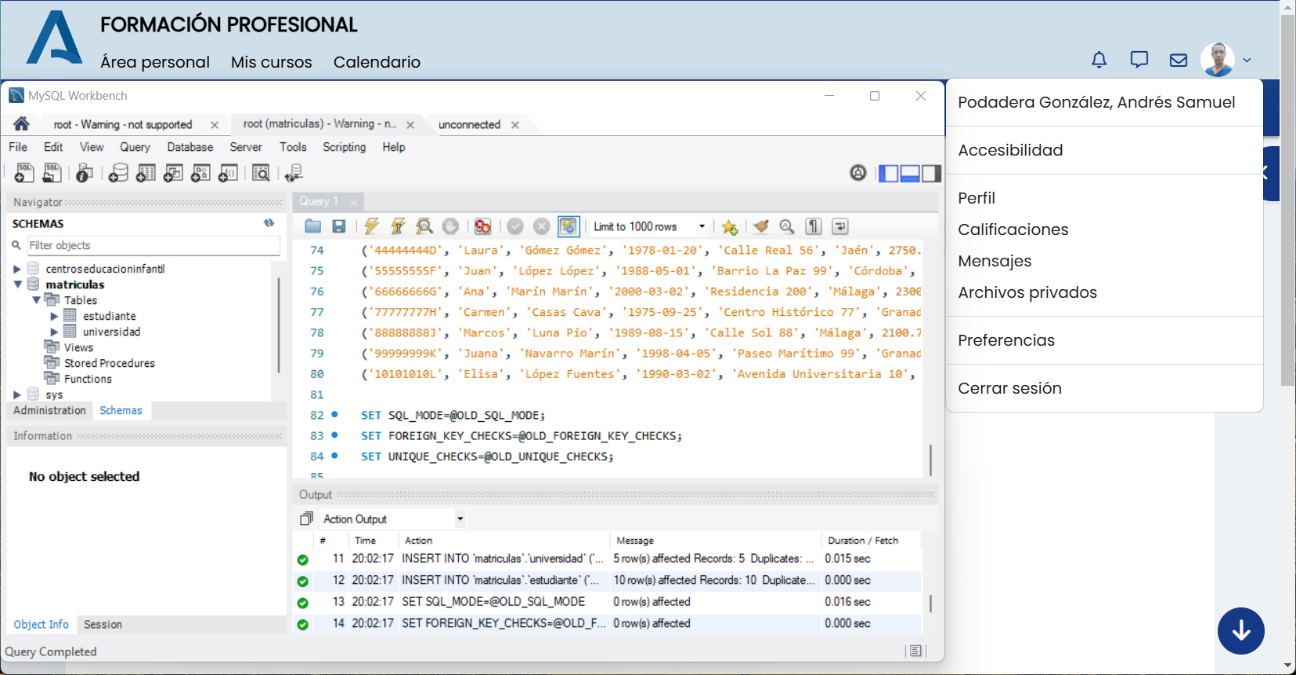
**Agregando la relación entre ambas tablas:**

**

**Me sirvo de la opción “Exportar” para obtener una primera versión de la BBDD:**

**

**Y por último modifico el script desde MySQL Workbench 8.0, agrego algunos datos de prueba y lo ejecuto:**

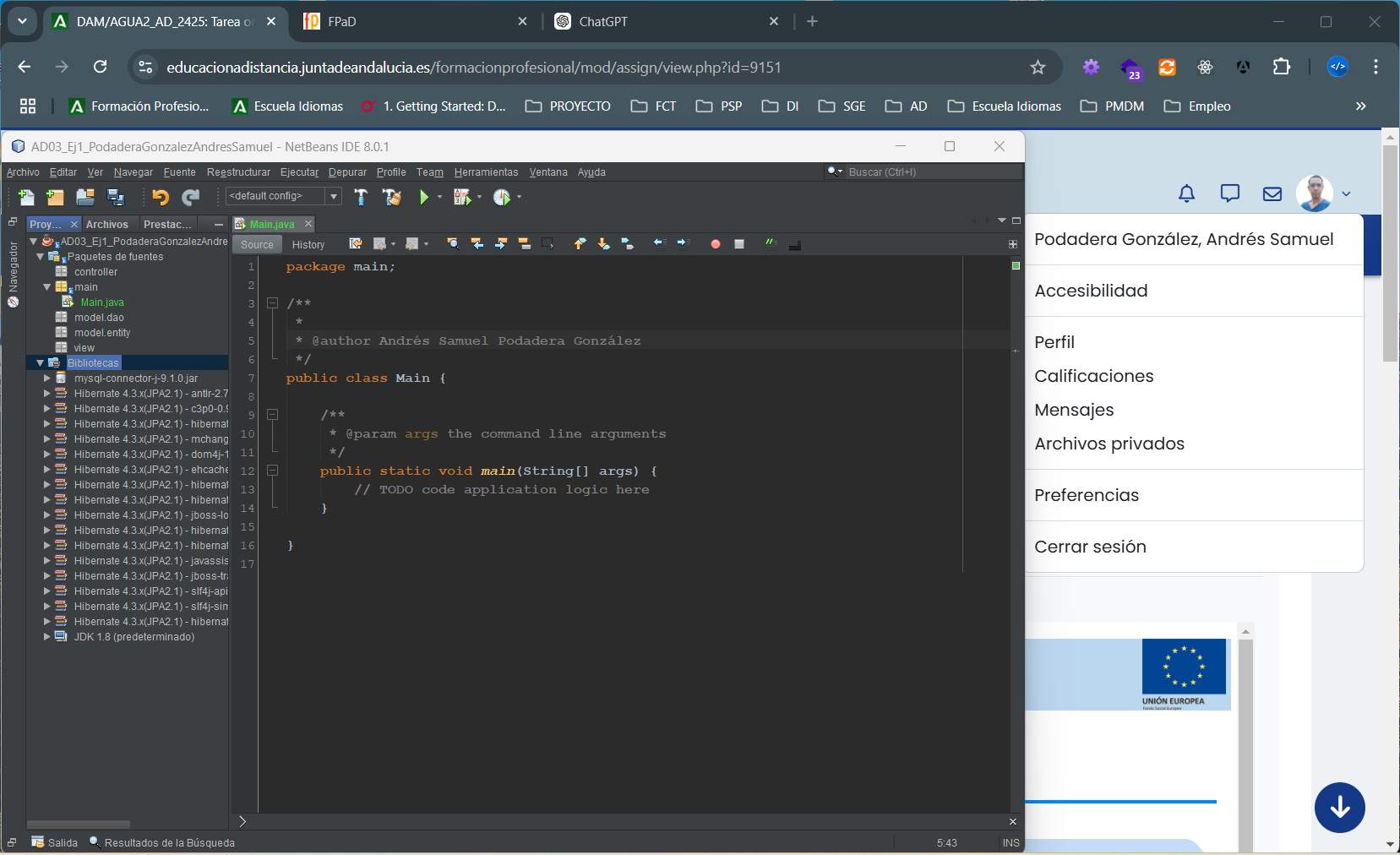
**

*Realiza las siguientes tareas:*

# ***EJERCICIO 1***

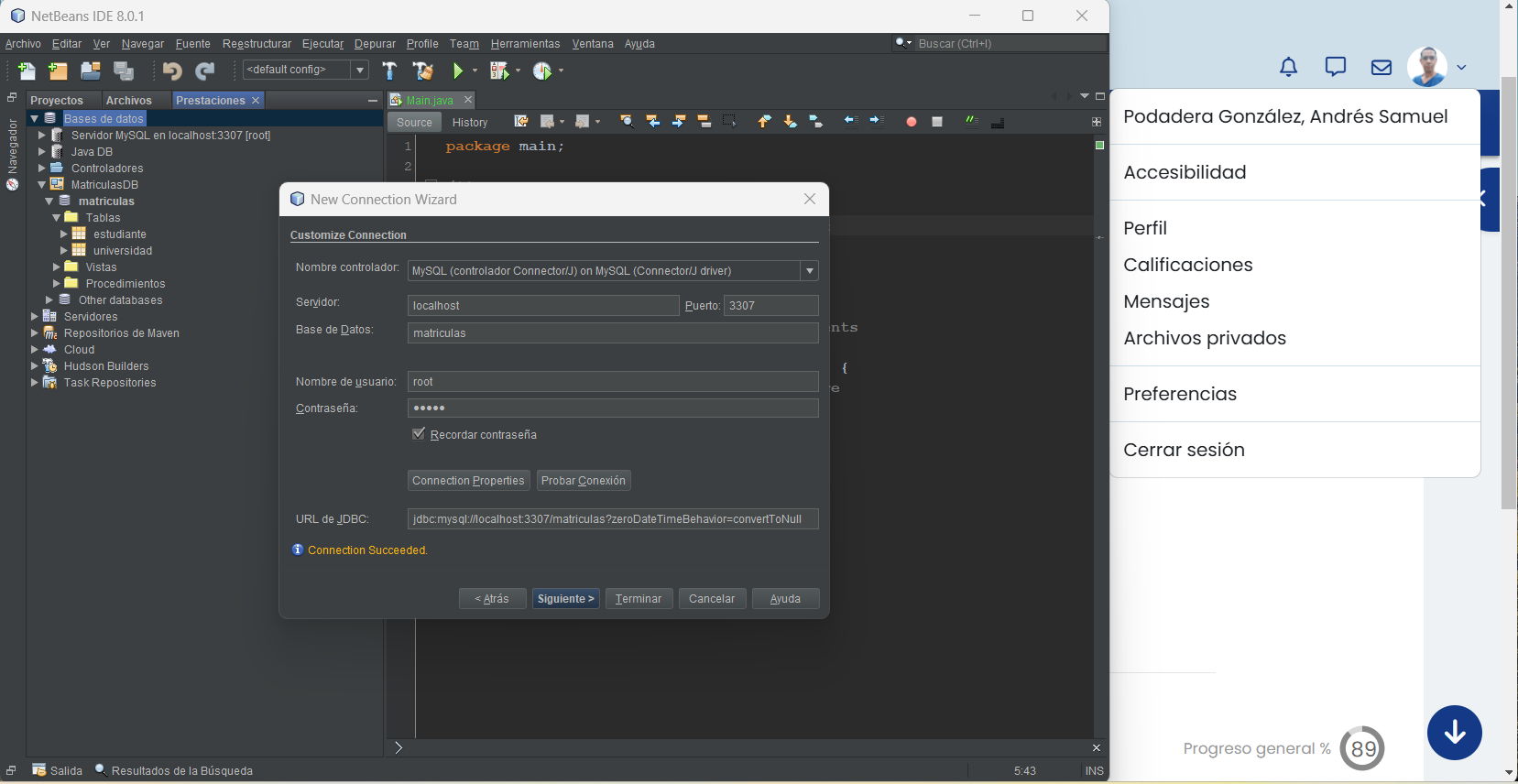
*Desarrolla un proyecto en Netbeans con nombre AD03\_Ej1\_ApellidosNombre que responda al patrón MVC y configura Hibernate para realizar el mapeo de esta base de datos relacional. Crea un****archivo pdf****donde documentes con capturas de pantalla los pasos que has seguido para la realización de las tres actividades siguientes:*

**Comienzo el proyecto creando los directorios, agregando el conector de BBDD MySQL y la librería de Hibernate 4.3:**

****

1. *Crea el fichero de configuración de Hibernate (hibernate.cfg.xml), con conexión a la base de datos ligafutbol con JDBC.*

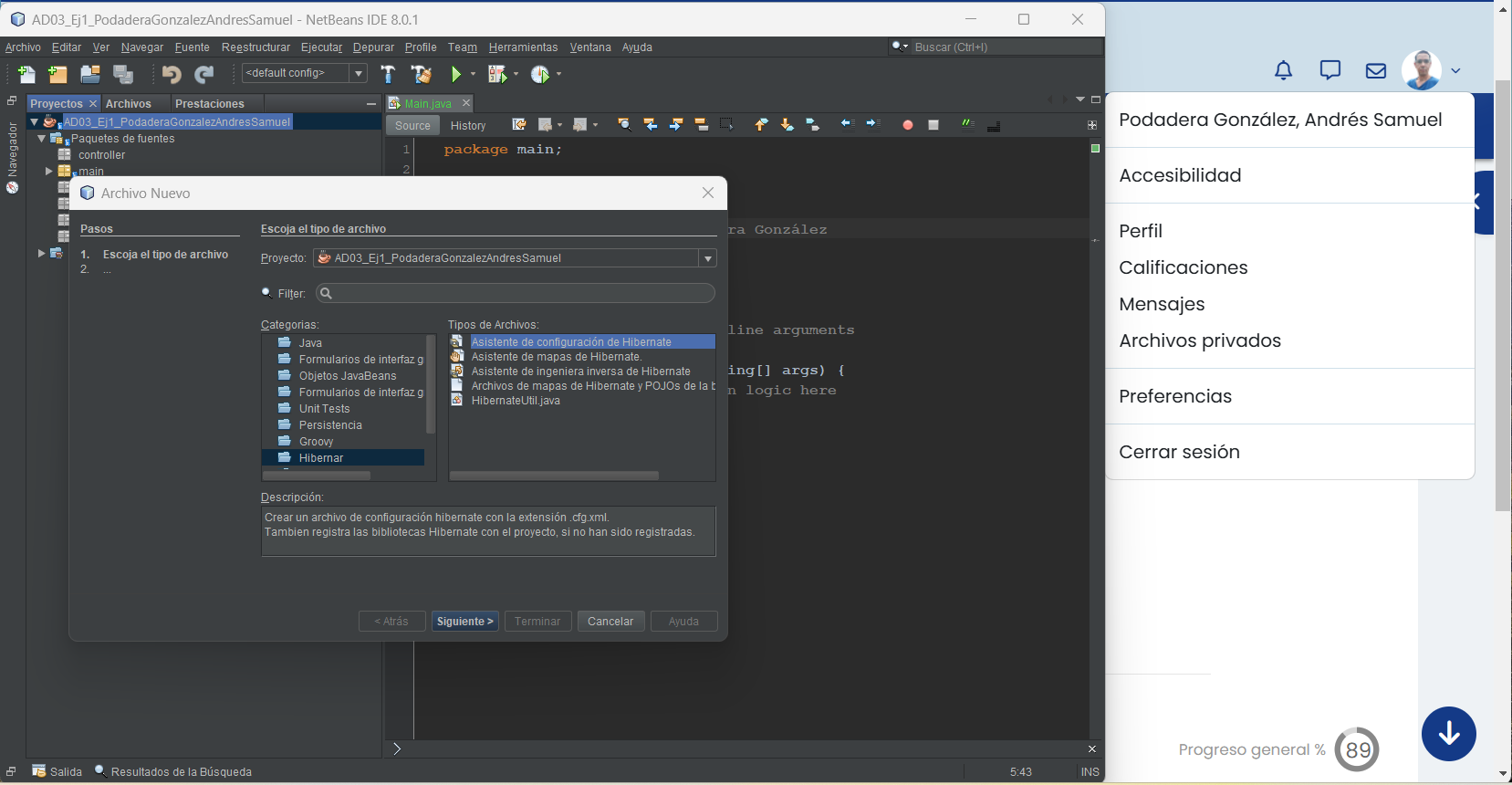
**Para facilitar la tarea primero me dispongo a crear la conexión a la base de datos matriculas, esto me permitirá indicar a Hibernate en su archivo de configuración, donde debe conectarse. A través del asistente configuro la url, driver, usuario y contraseña para realizar la conexión:**

****

**Una vez realizada la conexión en la pestaña “Prestaciones” del Netbeans, podremos ver las tablas de la BD y realizar consultas a la misma.**

**Ahora usaremos el asistente de Hibernate para configurar la conexión de este ORM a la BD matriculas:**

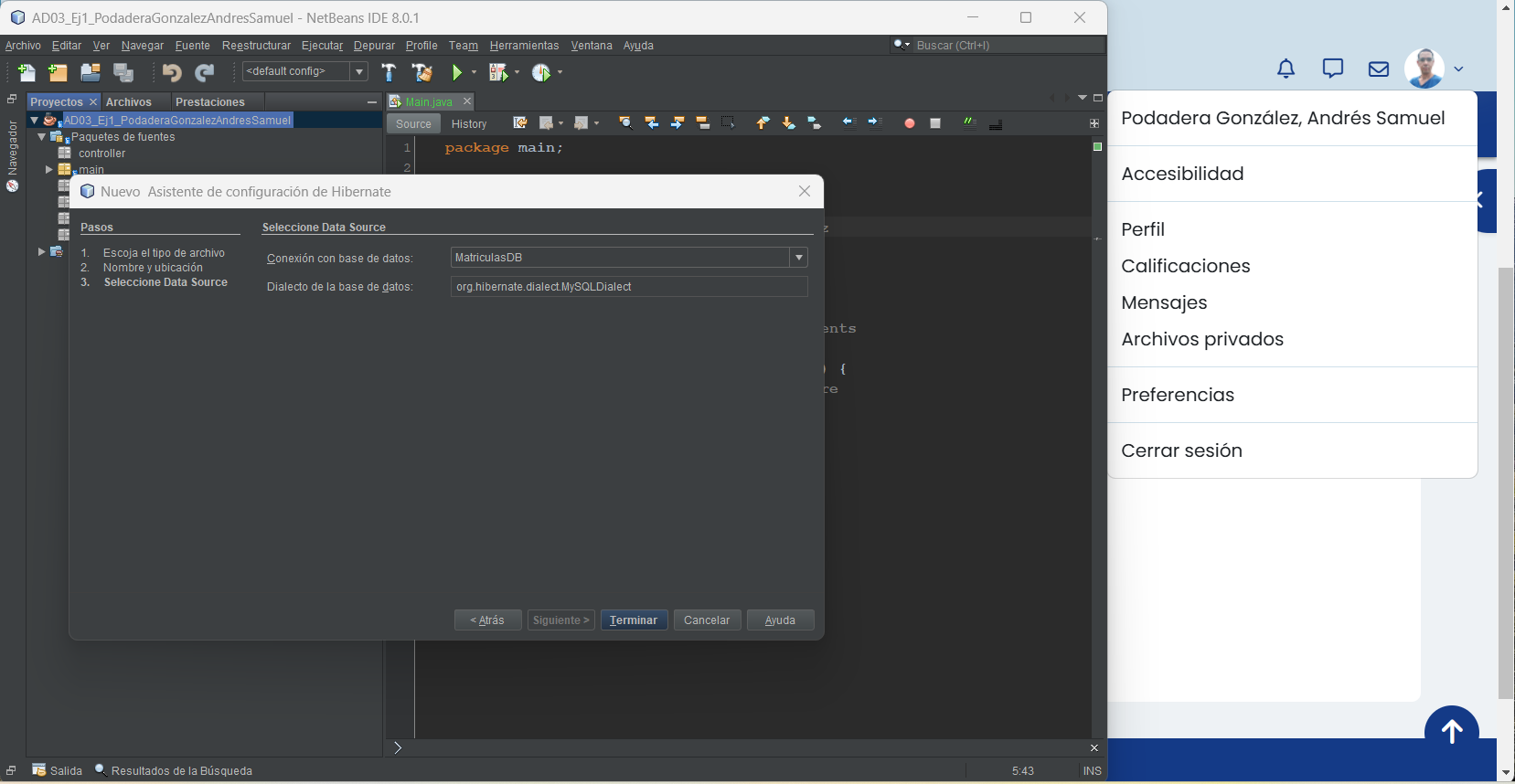
**Hacemos clic derecho sobre el proyecto, seleccionamos la opción “otro” e “Hibernar”. Dentro de este, elegimos crear el archivo de configuración hibernate.cfg.xml:**

****

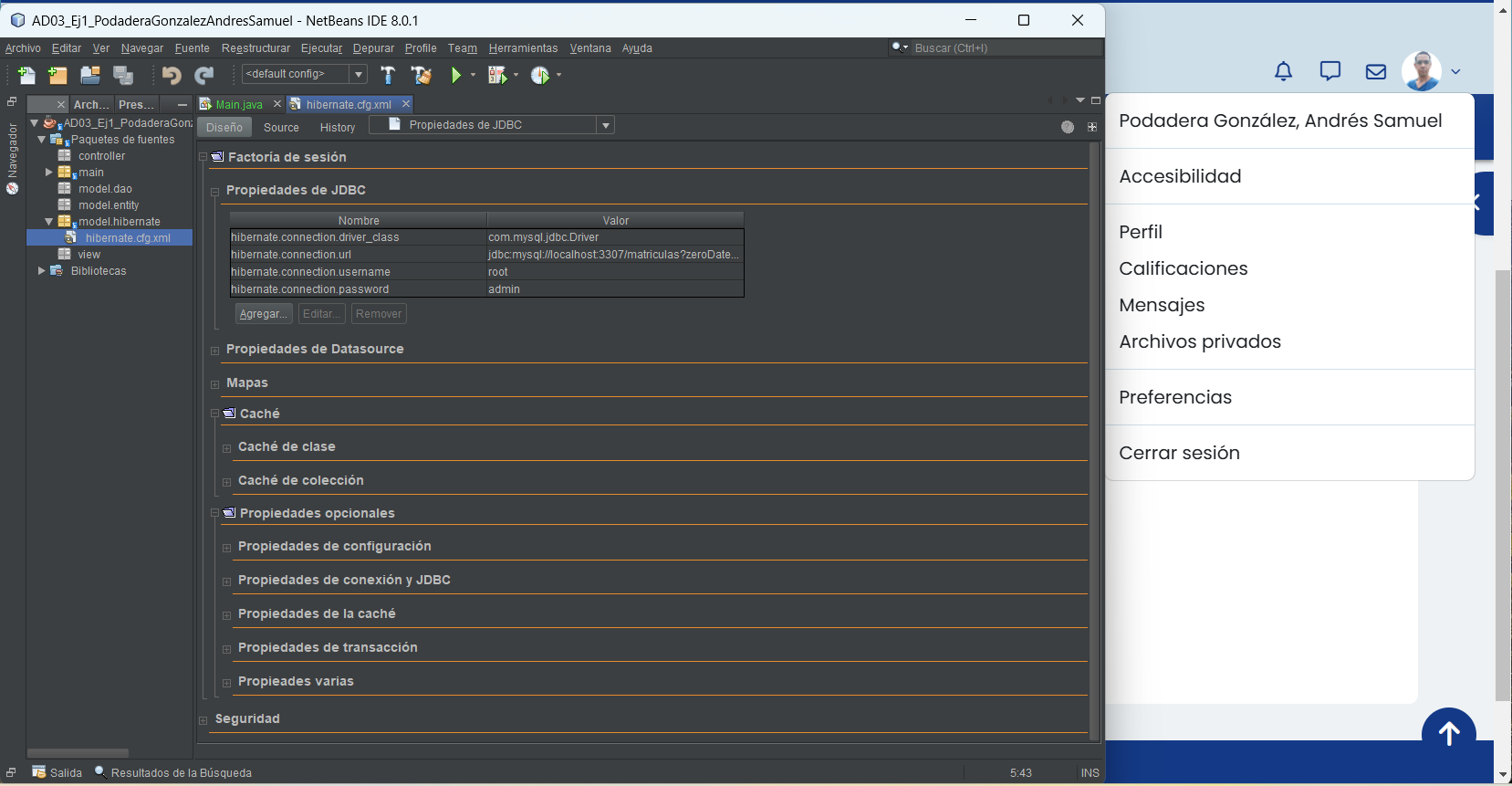
**Elegimos el directorio donde se almacenará el archivo:**

****

**Seleccionamos la base de datos que anteriormente hemos creado:**

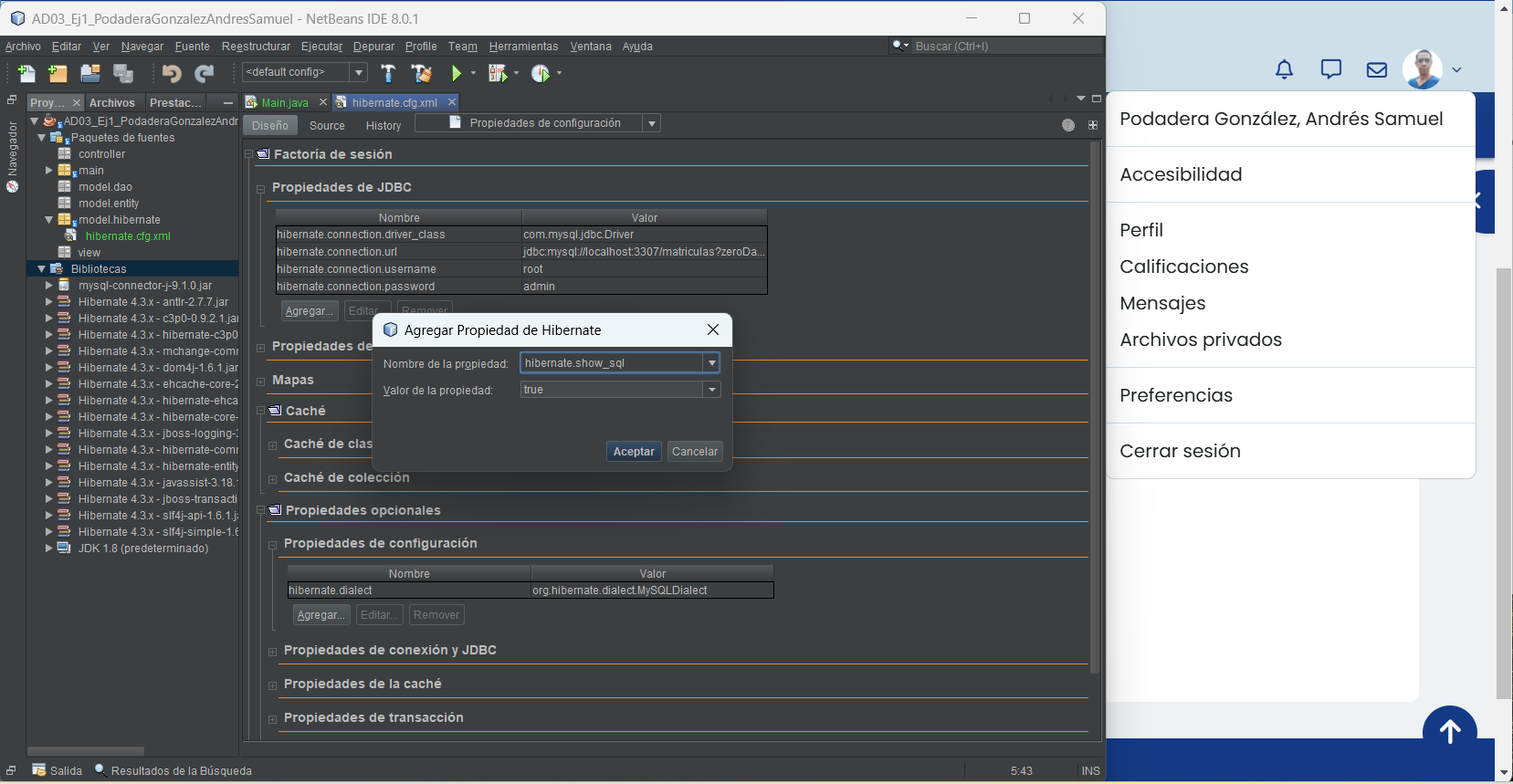
****

**Y haciendo clic en Terminar, nos muestra el archivo de configuración:**

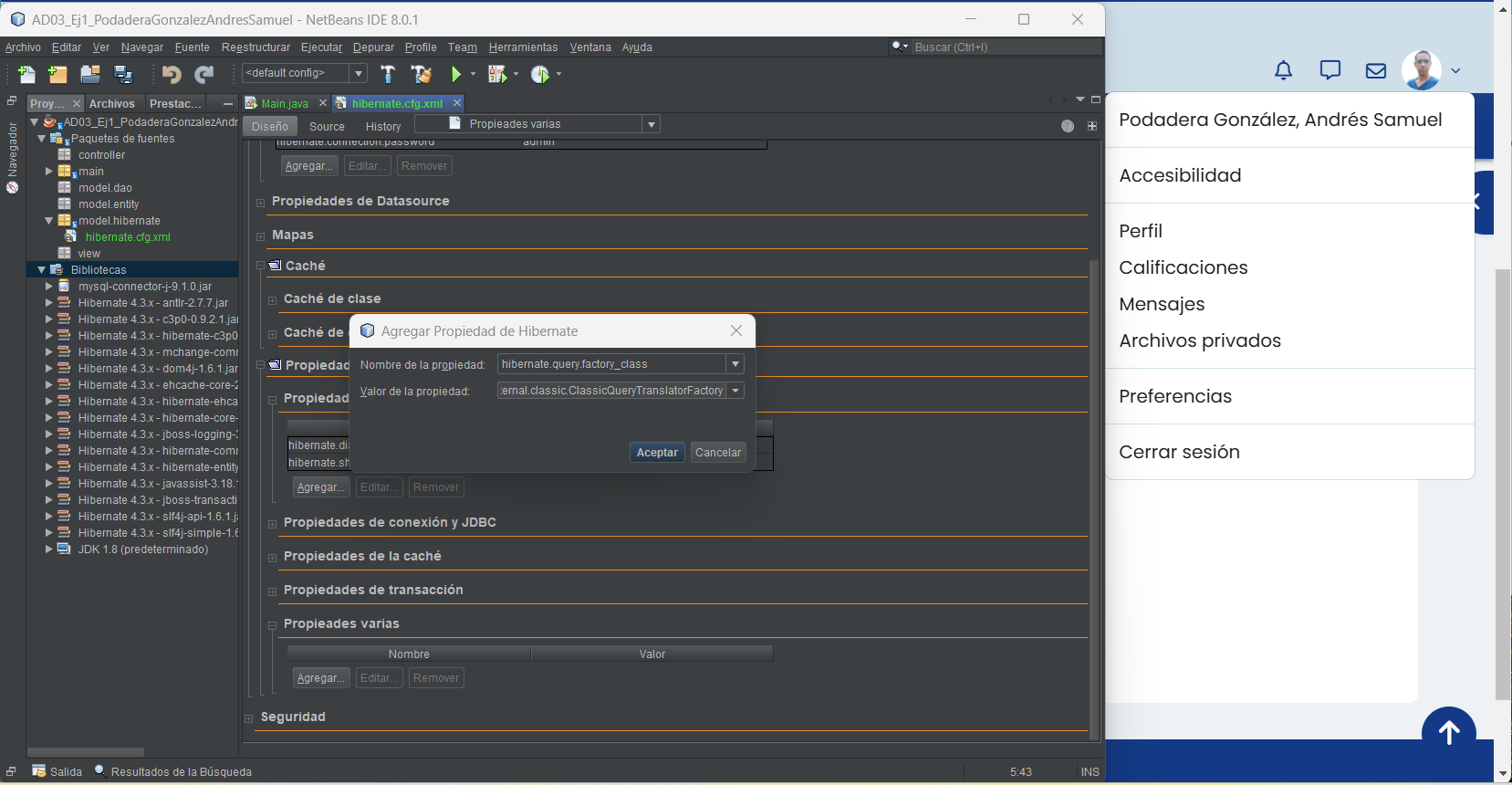
****

**También nos agrega la librería Hibernate automáticamente al proyecto en este último paso. Hecho esto debemos realizar la configuración de este archivo con algunos parámetros extra:**

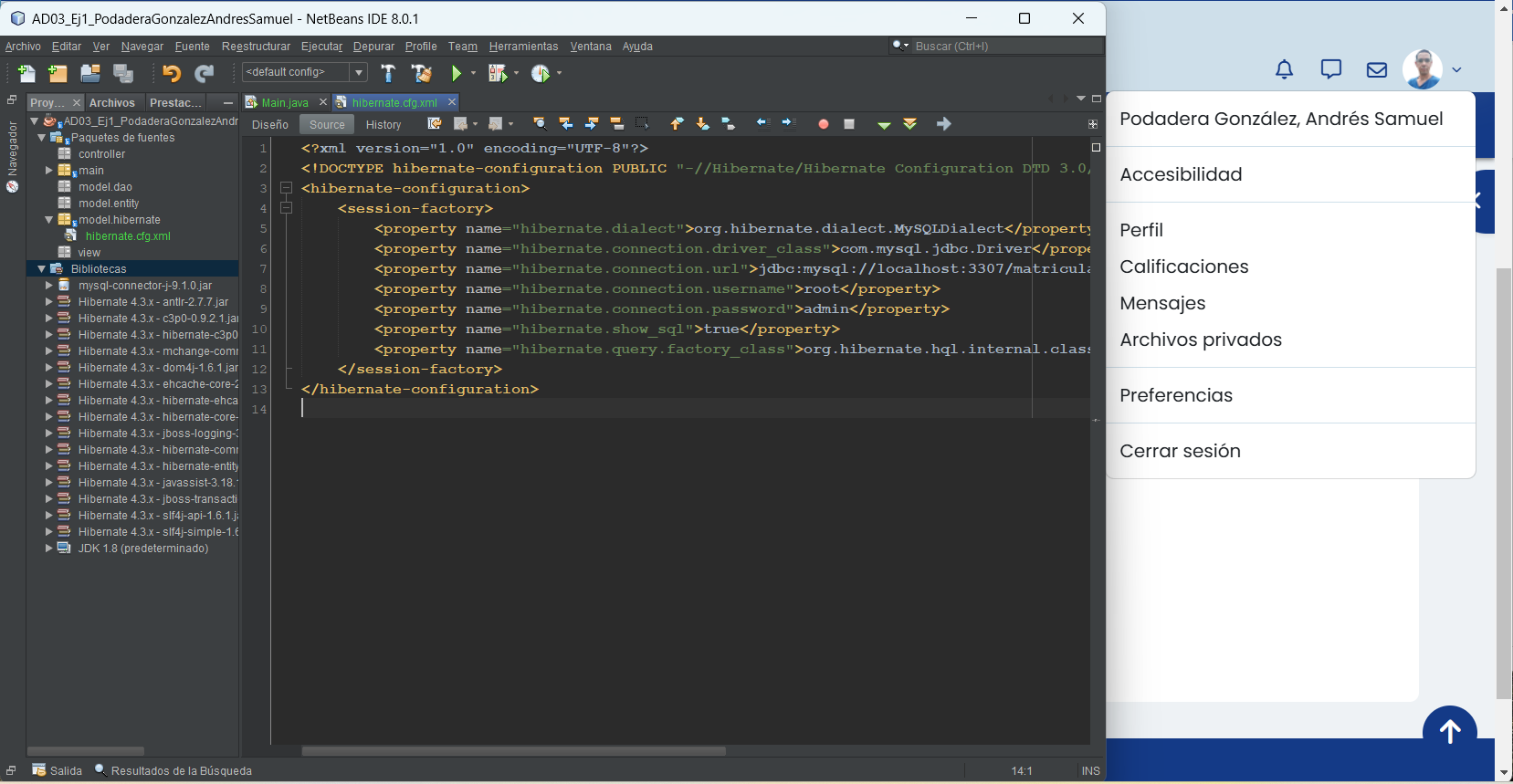
* **Habilitamos el seguimiento de la depuración de las consultas desde el apartado de Propiedades Opcionales:**

****

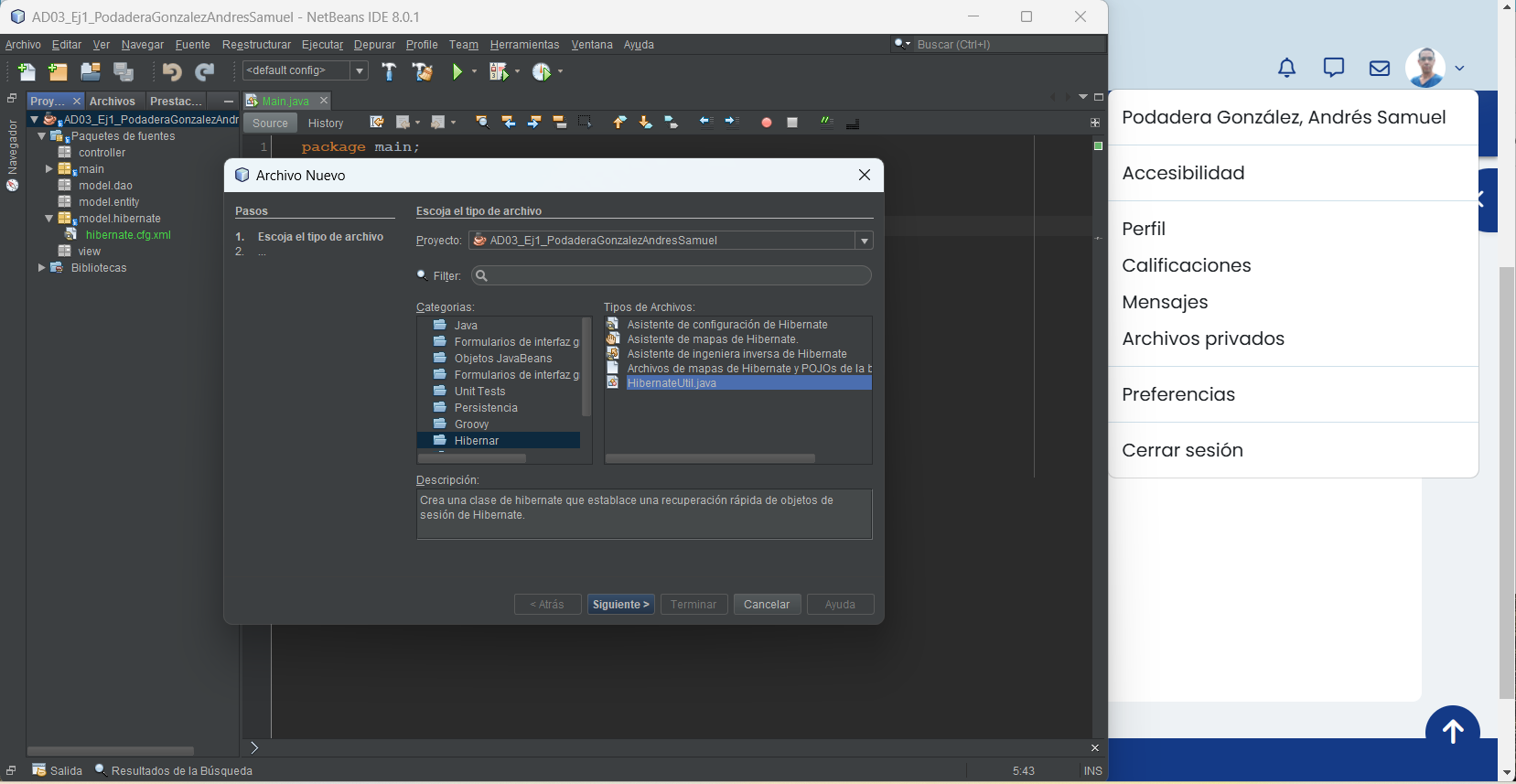
* **Especificar el motor de traducción de consultas HQL desde la opción de Propiedades Varias:**

****

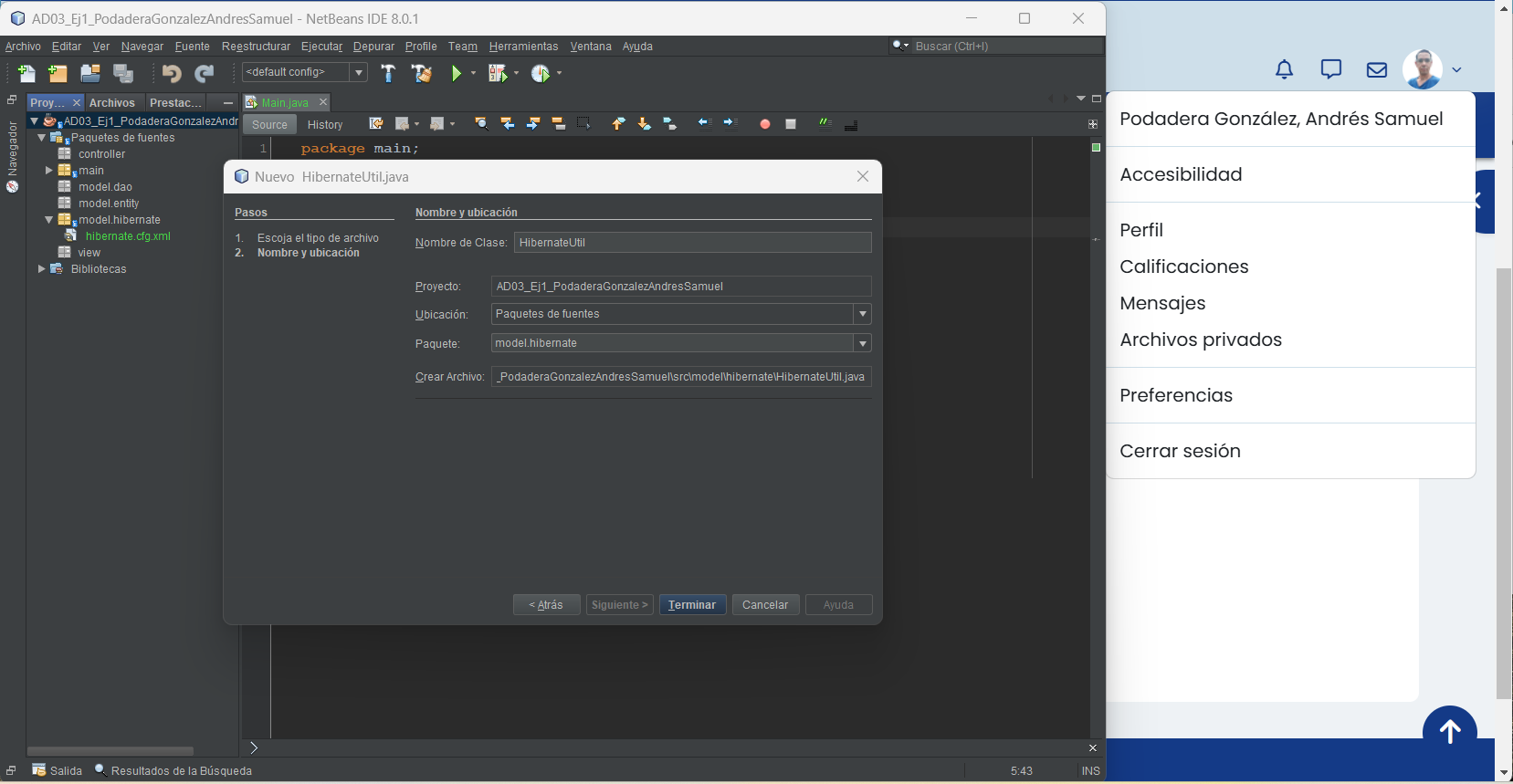
**Y guardamos la configuración realizada:**

****

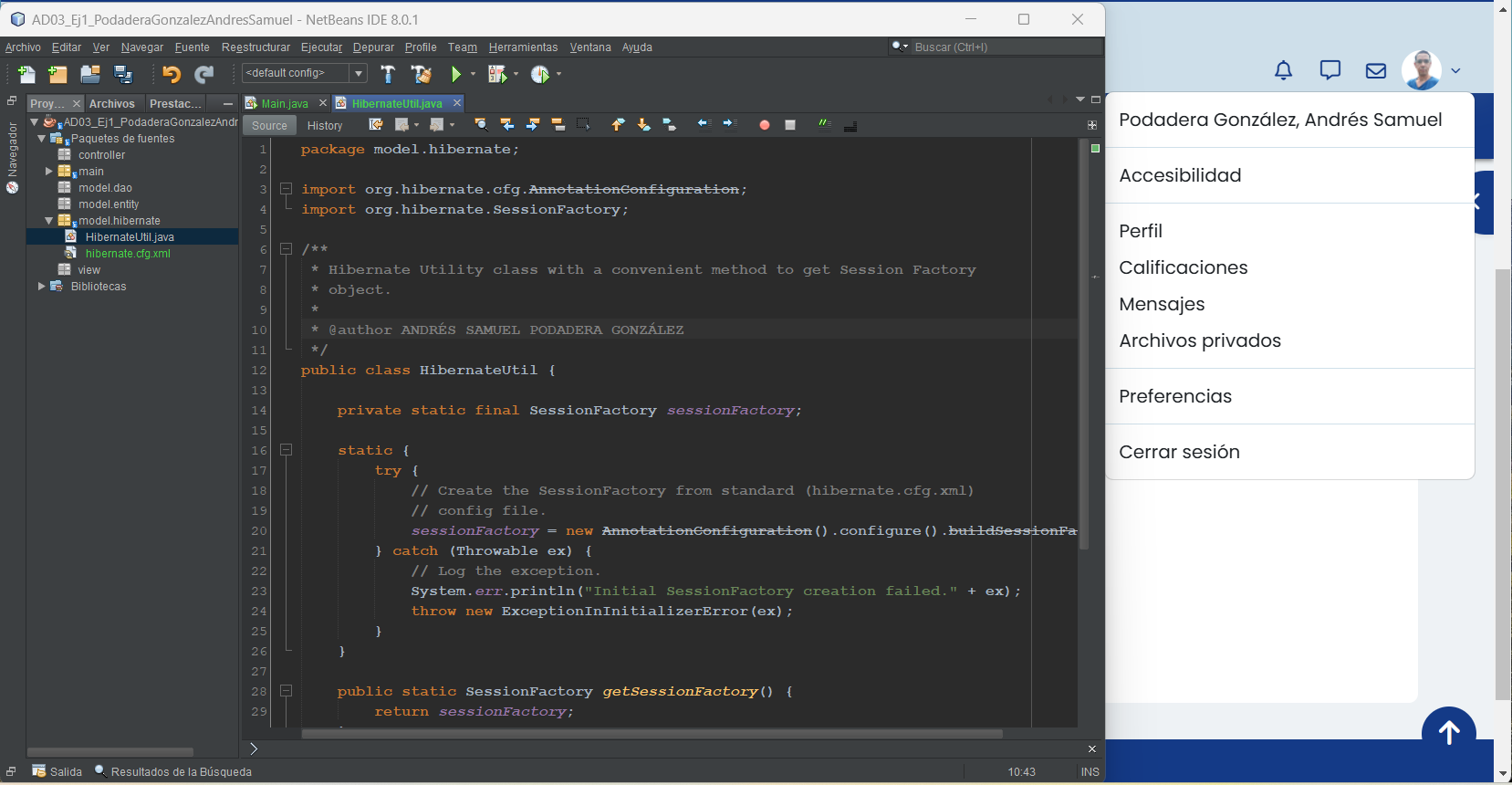
**Ahora necesitamos crear el archivo HibernateUtil.java, un archivo que nos permitirá crear a partir del archivo de configuración anterior, el SessionFactory y el objeto Session que usa Hibernate para las conexiones y operaciones con la base de datos. Para ello usamos el mismo asistente que en caso anterior, pero esta vez elegimos HeibernateUtil.java:**

****

**Elegimos el paquete donde se almacenará la clase y hacemos clic en Terminar:**

****

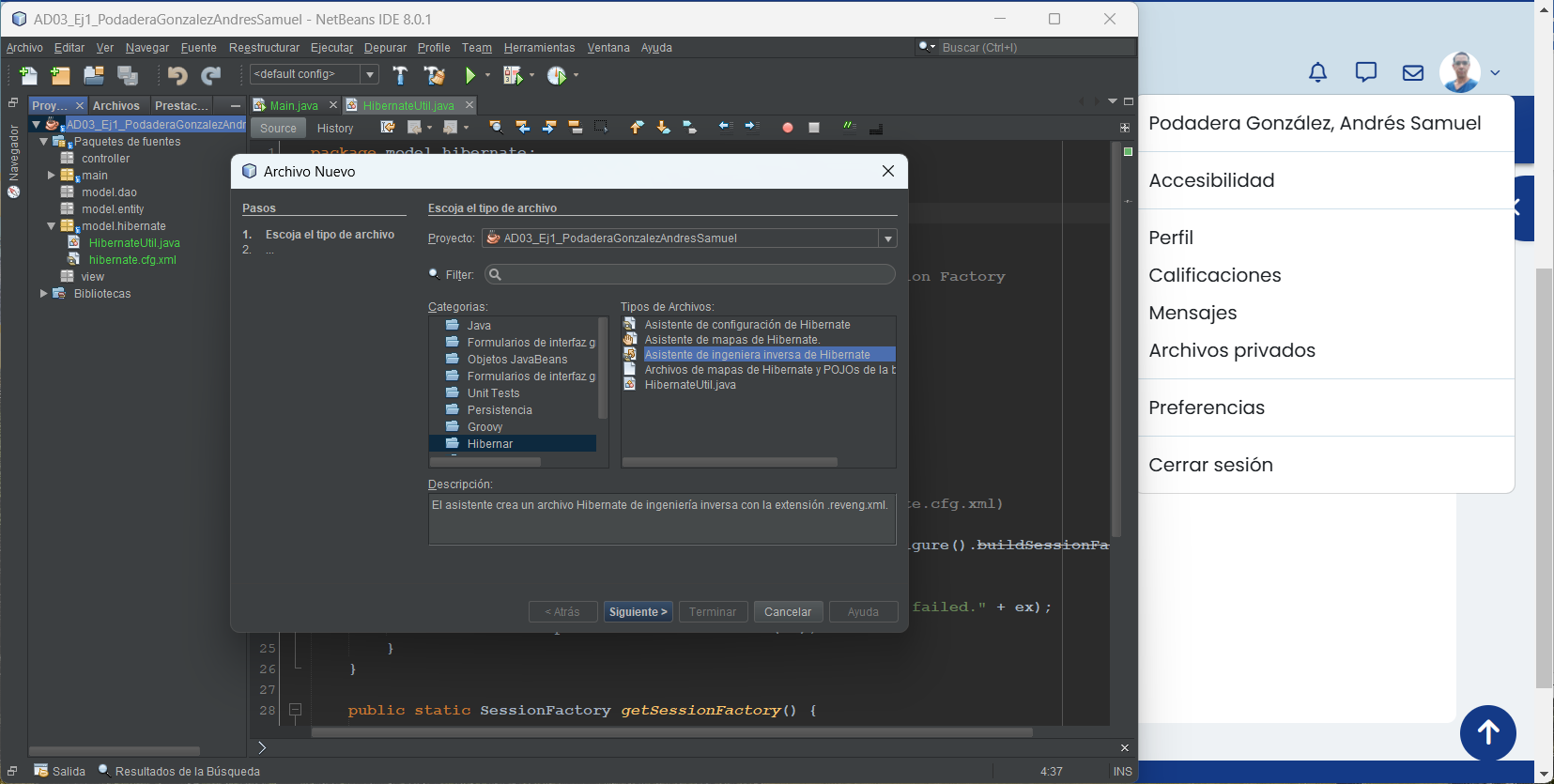
**Y acaba de crearse la clase:**

****

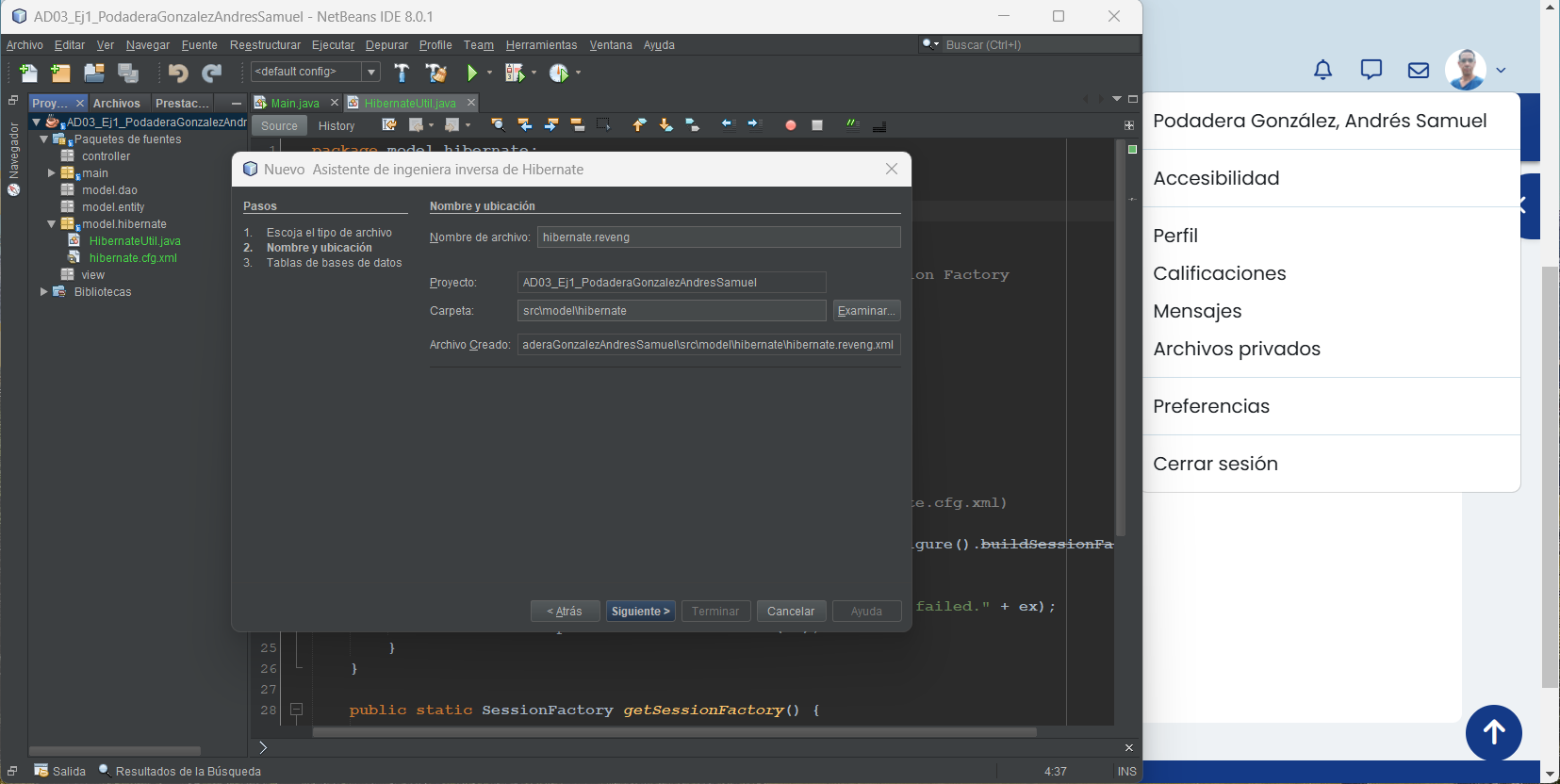
**Ya hemos realizado la configuración de la conexión a la base de datos de nuestro proyecto mediante el ORM Hibernate.**

1. *Crea el fichero de ingeniería inversa hibernate.reveng.xml e indica las tablas sobre las que se va a establecer correspondencia.*

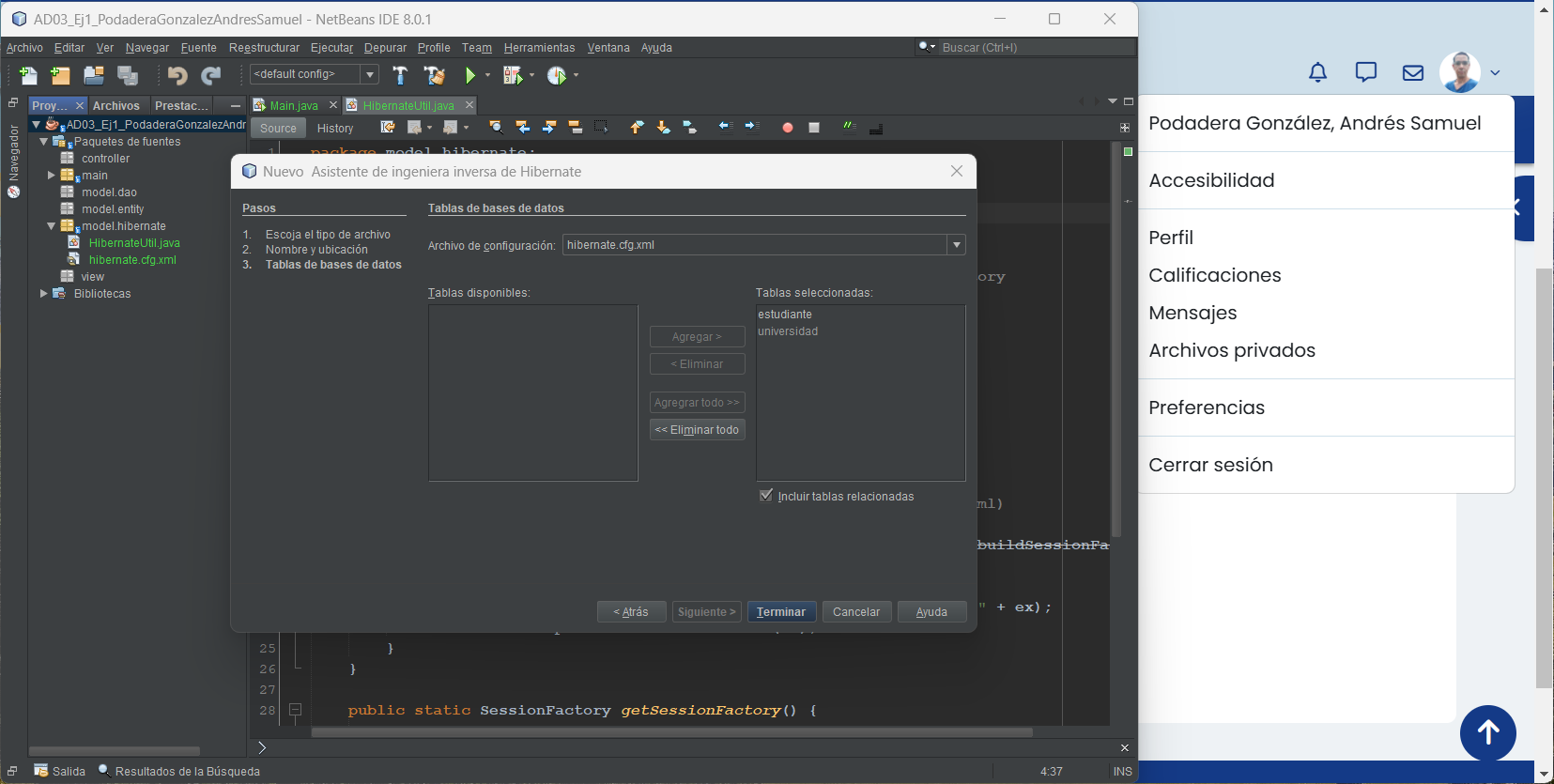
**Para crear este archivo hibernate.reveng.xml, nos dirigimos al mismo asistente de configuración de Hibernate que en los casos anteriores y elegimos la opción “Asistente de ingeniería inversa de Hibernate”:**

****

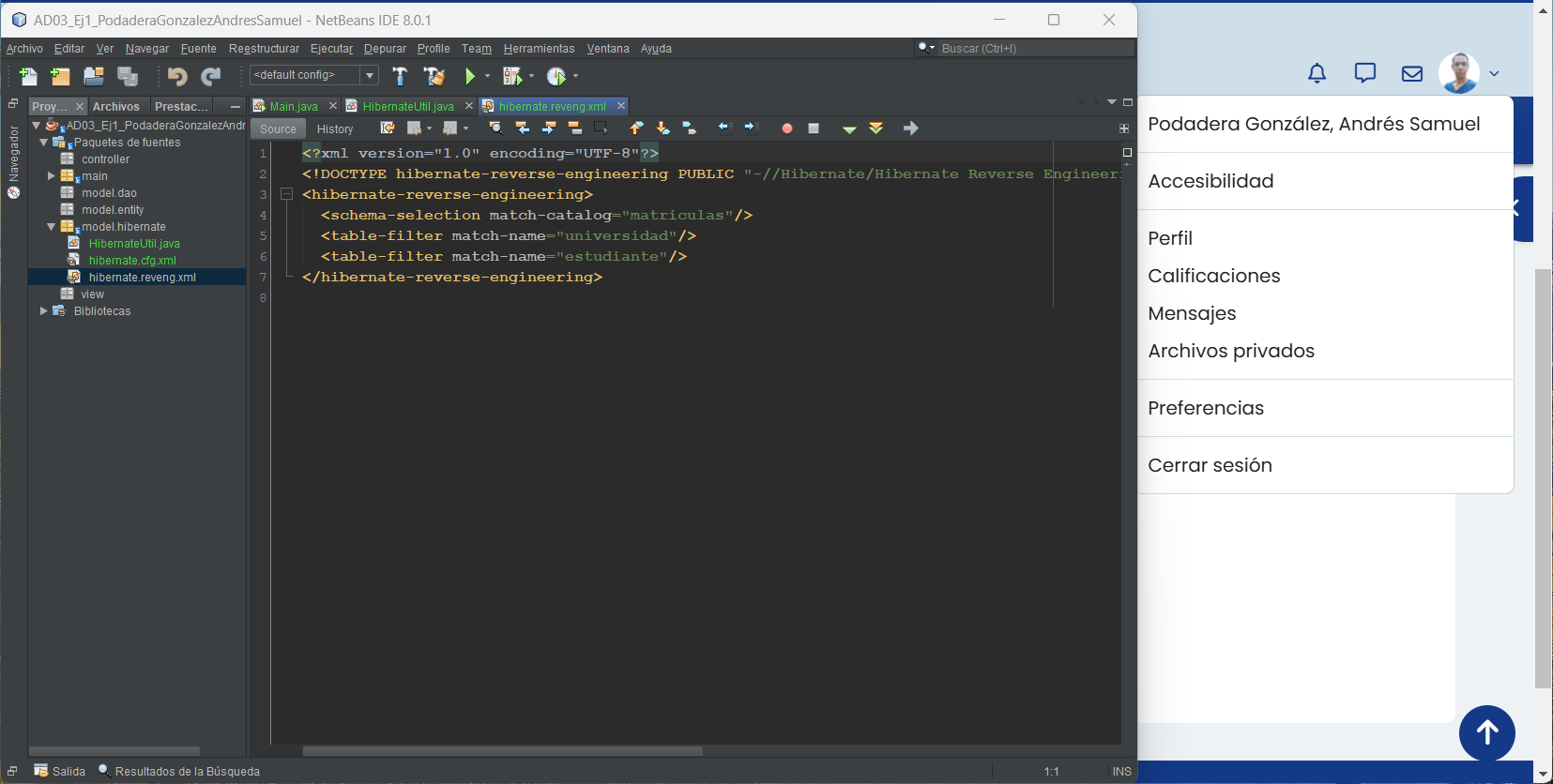
**Luego elegimos el paquete donde se almacenará el archivo:**

****

**Agregamos las tablas de BD a las que queremos hacer reversing(obtener la clase en POO a partir de la tabla en la BD realacional):**

****

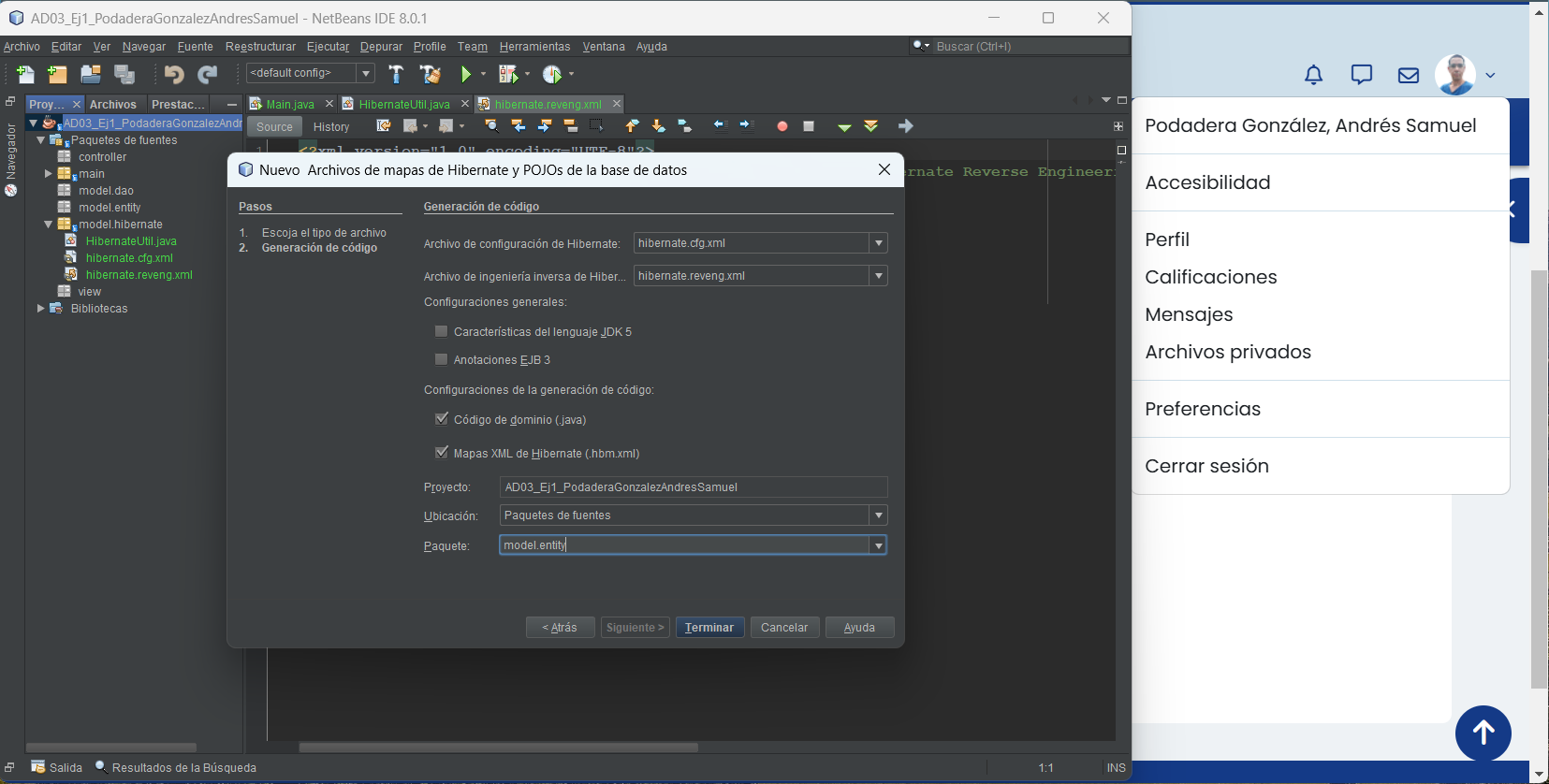
**Y ya tendríamos el archivo .xml creado correctamente:**

****

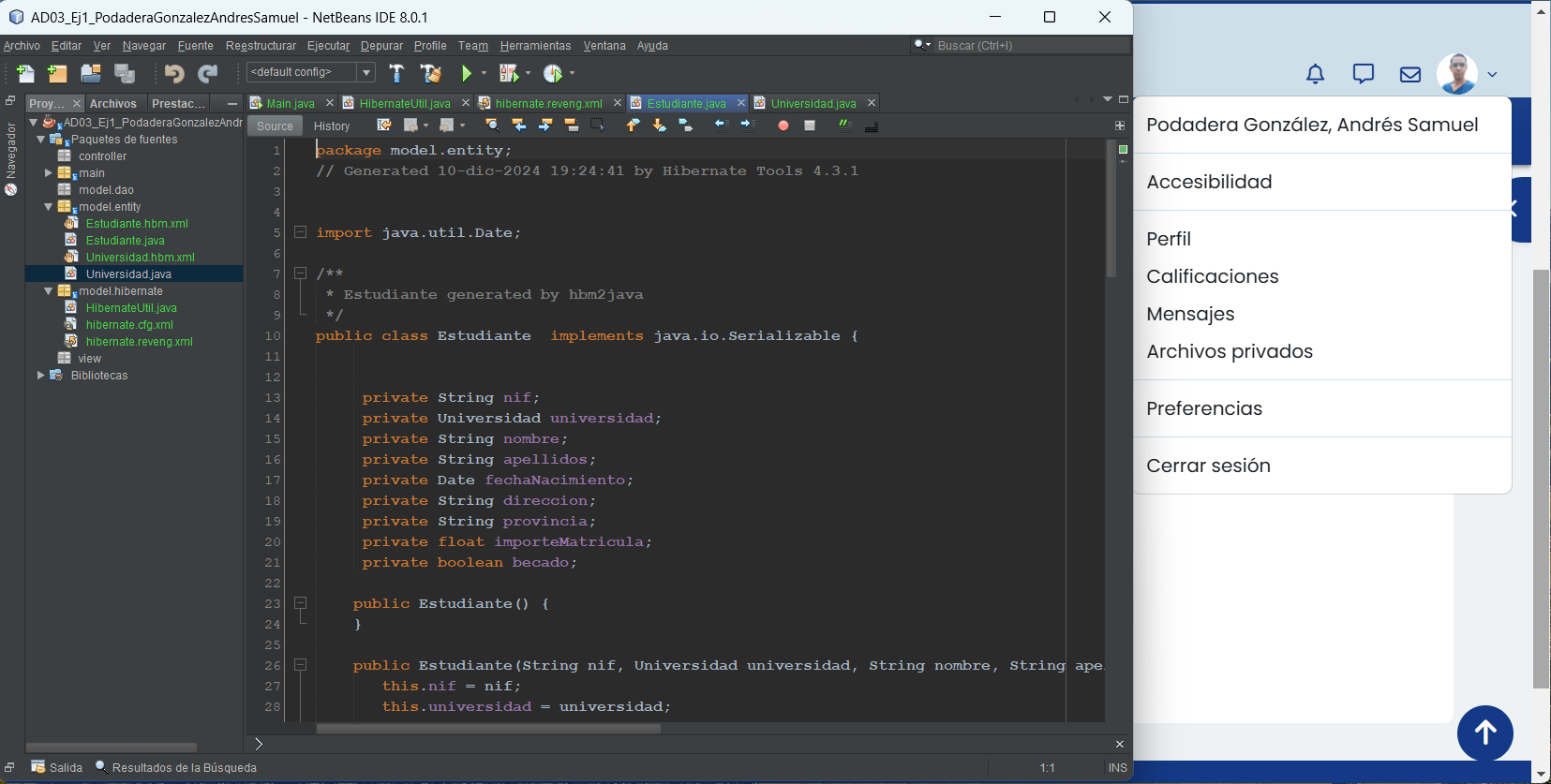
1. *Genera las clases y los ficheros de correspondencia (hbm).*

**Para generar los POJOs y los ficheros hbm nos dirigimos al asistente de Hibernate y elegimos la opción “Archivos de mapas Hibernate y POJOs de base de datos”:**

**Luego seleccionamos el archivo de configuración de Hibernate, el de ingeniería inversa y el paquete donde se guardarán los ficheros:**

****

**Por último, comprobamos que los archivos se han generado correctamente:**

****

# ***EJERCICIO 2***

*Desarrolla opciones en nuestro proyecto que den la posibilidad de realizar las siguientes acciones:*

1. *Añadir nuevos estudiantes y nuevas universidades a la base de datos.*
2. *Eliminar estudiantes y universidades de la base de datos (no se podrá eliminar una universidad que tenga algún alumno matriculado).*
3. *Mostrar el listado de universidades y alumnos de cada universidad.*
4. *Poder realizar las siguientes consultas:  
   - Nombre, apellidos, universidad e importe de matrícula de los estudiantes, ordenados por importe de la matrícula de menor a mayor.  
   - Nombre de universidad y el importe ingresado por matrículas de alumnos.  
   - Nombre, apellidos y dirección de todos los estudiantes de la provincia de 'ALMERIA' y que son becados.*
5. *Inserta un nuevo objeto estudiante en la base de datos con los siguientes datos: nif: 11110000B, nombre: "Javier", apellidos: "Pérez Pérez", fecha\_nacimiento: 23/11/2000, direccion: "Velázque, 33, Aguadulce", provincia: "GRANADA", importe\_matricula: 1550,20, becado: true, codigo\_uni: 1.*
6. *Modifica el importe de matrícula del objeto estudiante de la BD creado en el apartado anterior con un nuevo valor de 1200,00.*