AD

Álvaro Manuel Navarro Cruz

2º DAM

10/12/2024

Práctica 3

HERRAMIENTAS DE MAPEO OBJETO-RELACIONAL. HIBERNATE

Índice

[Introducción (↑) 3](#_Toc184078223)

[Repositorio en GitHub 3](#_Toc184078224)

[1.- Fichero pom.xml (↑) 4](#_Toc184078225)

[2.- Fichero de configuración de Hibernate (↑) 5](#_Toc184078226)

[3.- Clase ClienteANC (↑) 6](#_Toc184078227)

[4.- Clase HotelManager (↑) 8](#_Toc184078228)

[5.- Clase TestHotel (↑) 9](#_Toc184078229)

[6.- Ejecución (↑) 10](#_Toc184078230)

[6.1.- Crear y Leer clientes 10](#_Toc184078231)

[6.2.- Modificar y Leer cliente 13](#_Toc184078232)

[6.3.- Eliminar y Leer cliente 15](#_Toc184078233)

[7.- Valoración Personal (↑) 17](#_Toc184078234)

[8.-Bibliografía 18](#_Toc184078235)

Introducción [(↑)](#indice)

Para el desarrollo de esta práctica, se nos pide realizar el código de persistencia necesario para un programa de gestión de hotel, concretamente el apartado del almacenamiento de clientes.

Debemos utilizar un proyecto de tipo Maven e implementar los métodos necesarios para las operaciones CRUD de la tabla que se creará a partir de nuestra clase POJO “ClienteANC”.

Para ello, debemos completar la clase POJO con anotaciones Hibernate directamente sobre la misma, sin uso de ficheros “.hbm.xml”.

También debemos configurar el archivo hibernate.cfg.xml con los parámetros de conexión a la Base de Datos “hotelanc”.

Para la ejecución del programa, debemos implementar una clase TestHotel con método Main, donde realizaremos las operaciones de alta, baja, modificación y consulta.

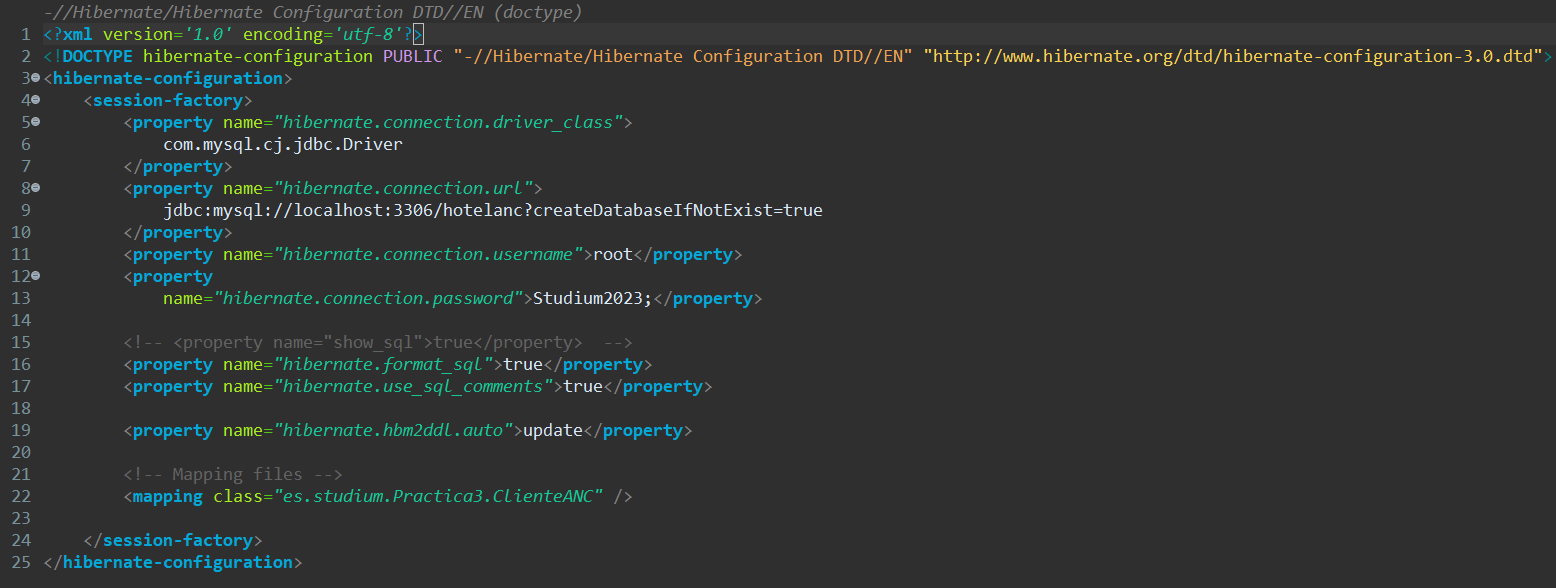
## Repositorio en GitHub

<https://github.com/AlvaroMfco/AD_Practica3.git>

1.- Fichero pom.xml [(↑)](#indice)



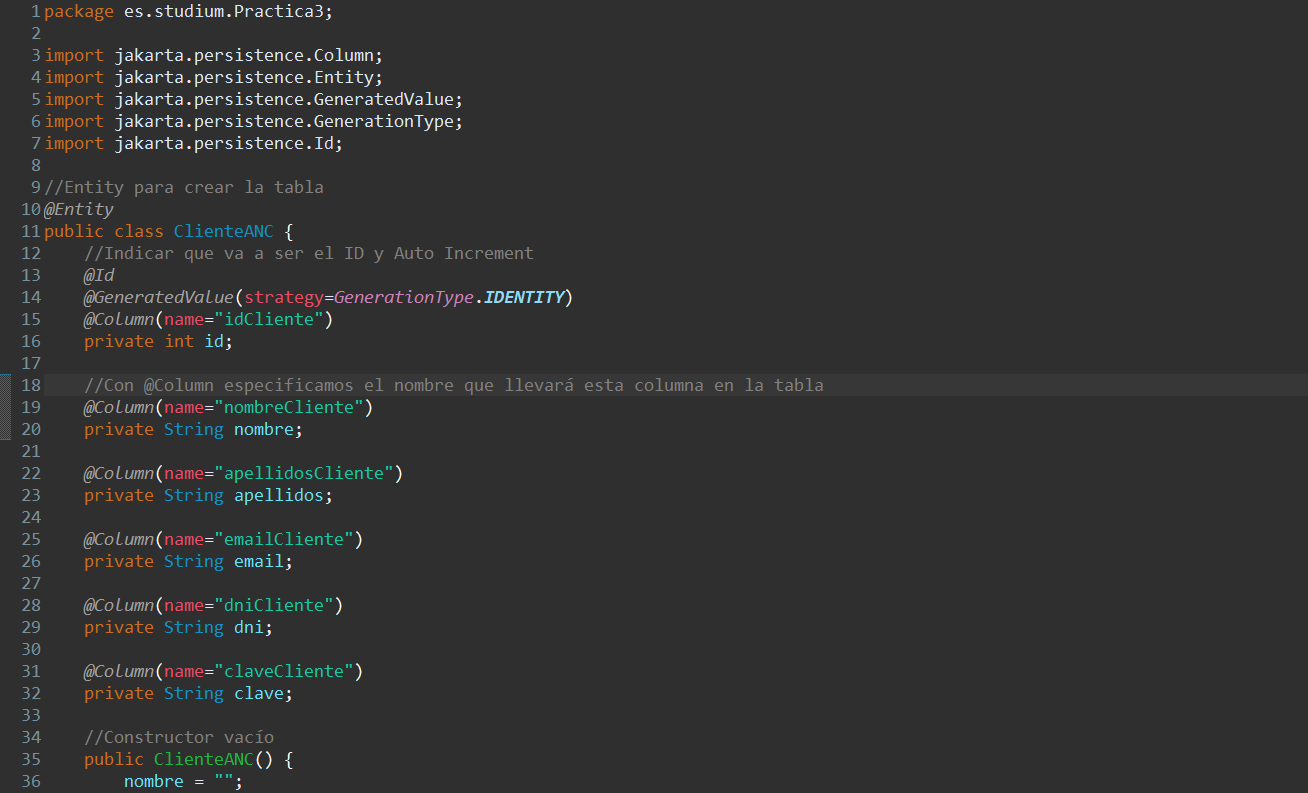
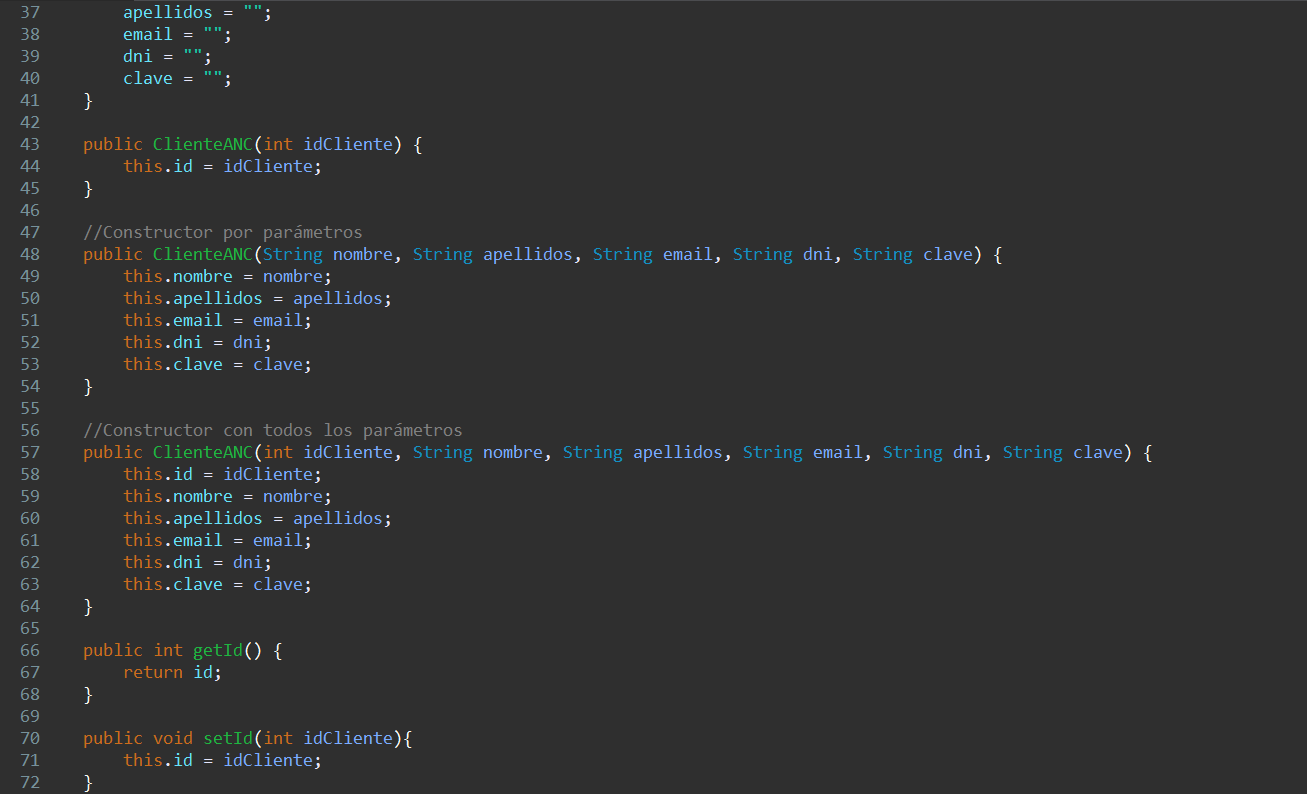
2.- Fichero de configuración de Hibernate [(↑)](#indice)

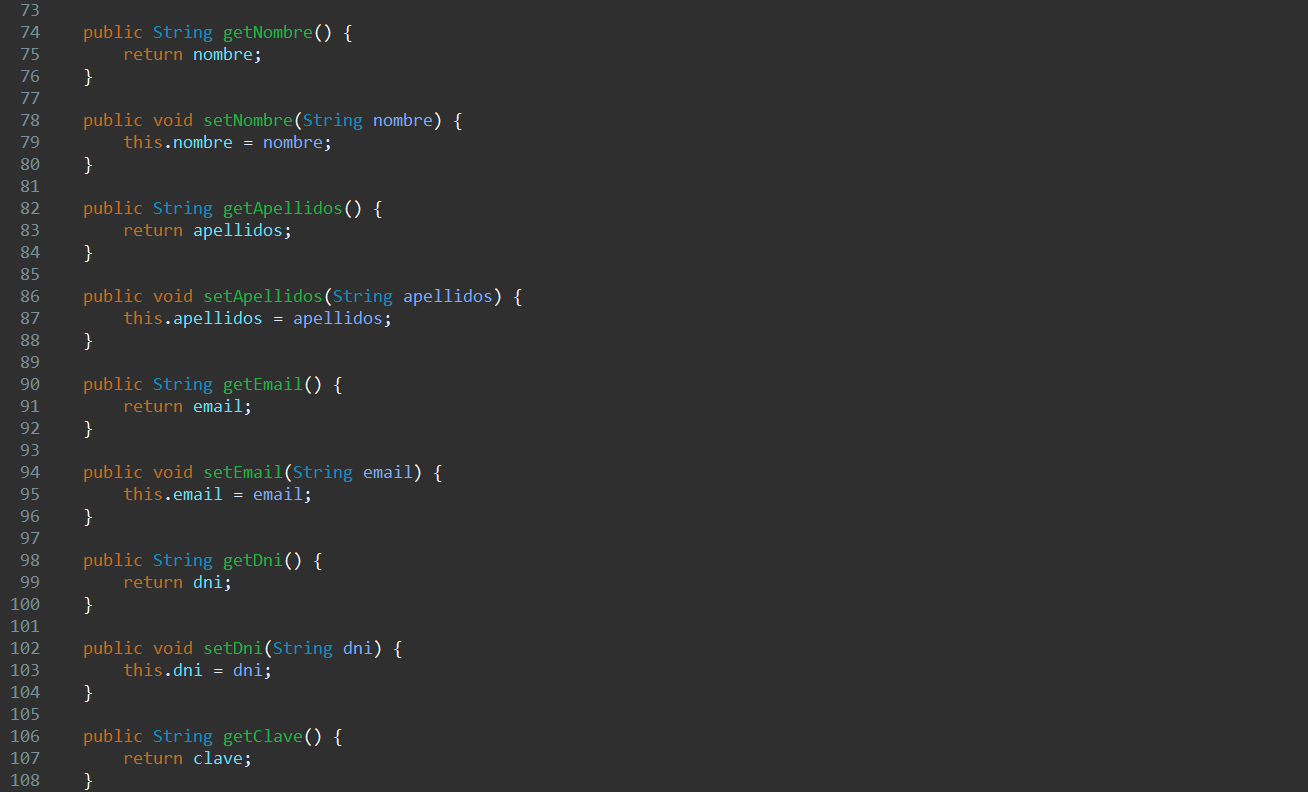


En este archivo se establece el conector JDBC, la URL de conexión a la base de datos, el usuario y la contraseña.

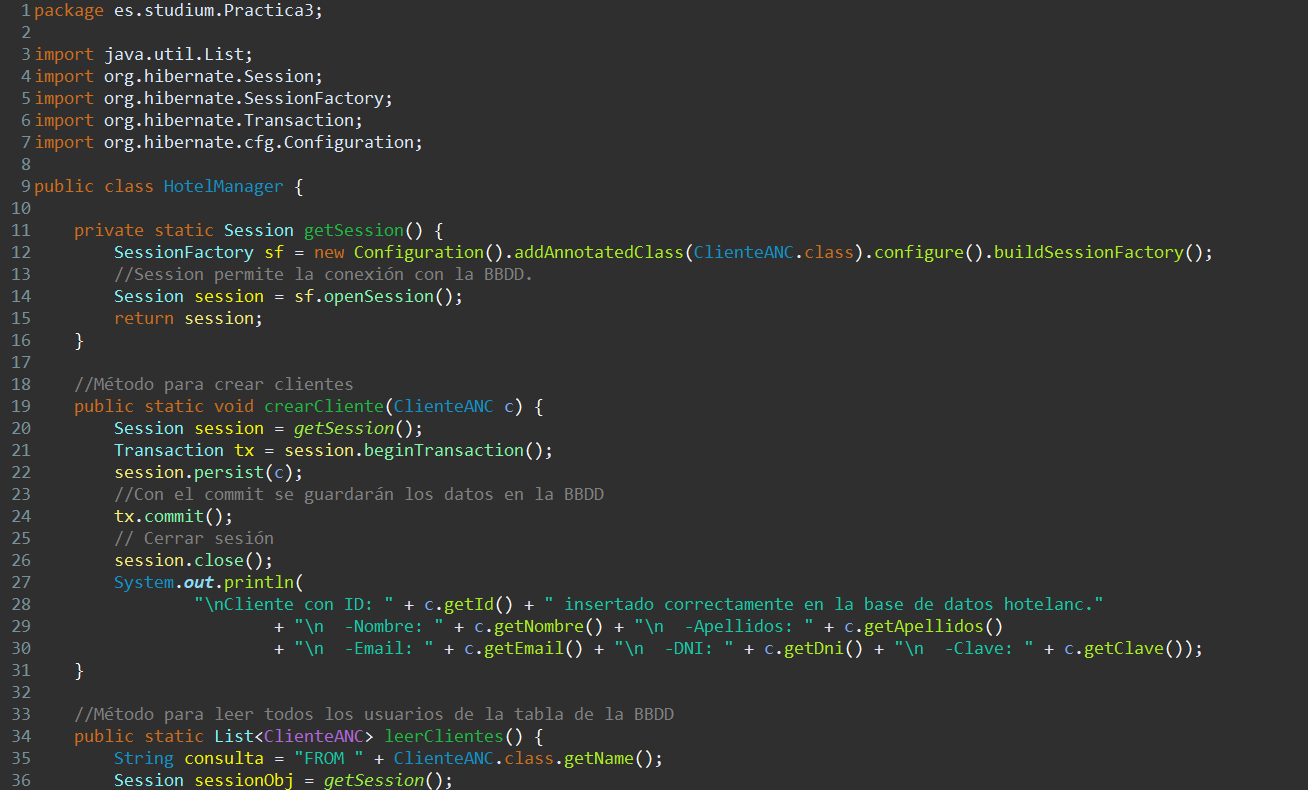
Se ha comentado la línea 15 para que no aparezcan las sentencias SQL por consola.

3.- Clase ClienteANC [(↑)](#indice)





4.- Clase HotelManager [(↑)](#indice)



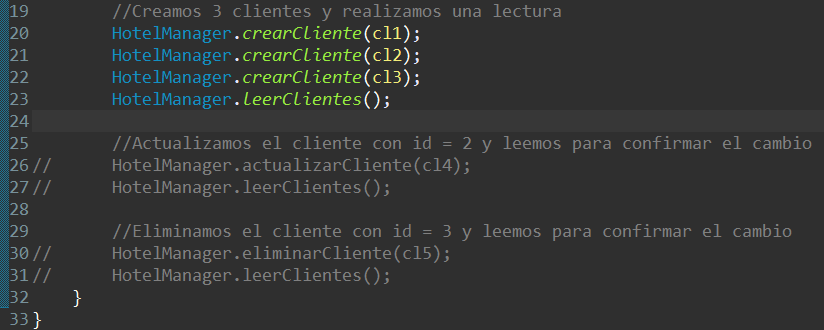
5.- Clase TestHotel [(↑)](#indice)



6.- Ejecución [(↑)](#indice)

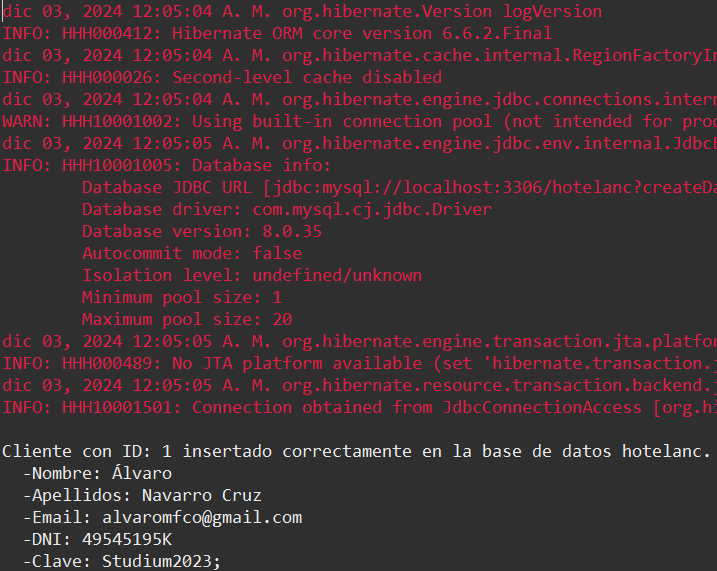
## 6.1.- Crear y Leer clientes

En primer lugar, vamos a comentar las líneas de Update y de Delete para crear los clientes y mostrarlos por consola.

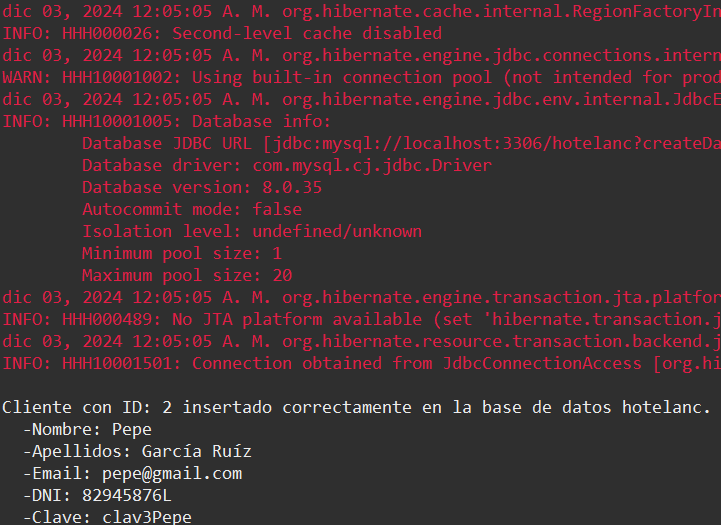


Al ejecutar el programa, por consola aparecerán los clientes al ser creados y, finalmente, todos los clientes encontrados en la BBDD.

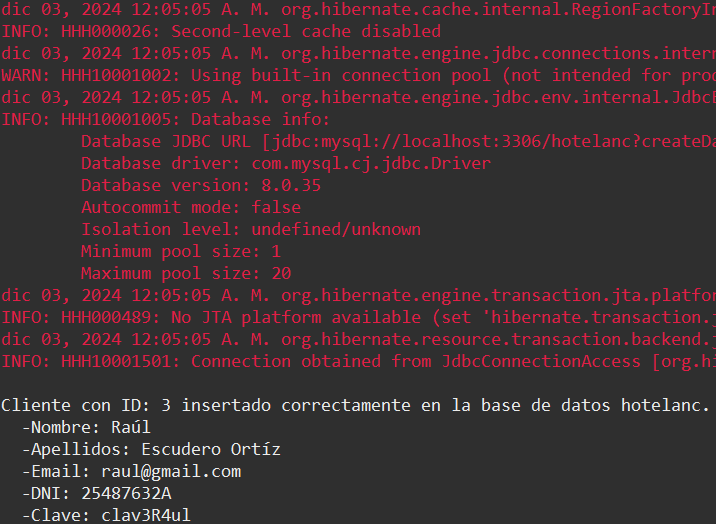
Cliente1:



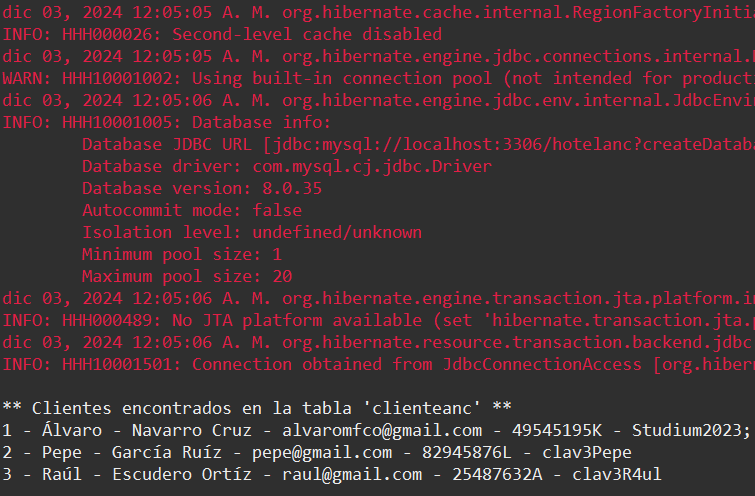
Cliente2:



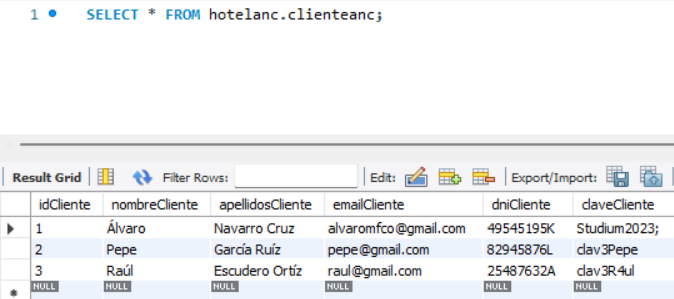
Cliente3:



Resultado de la consulta:

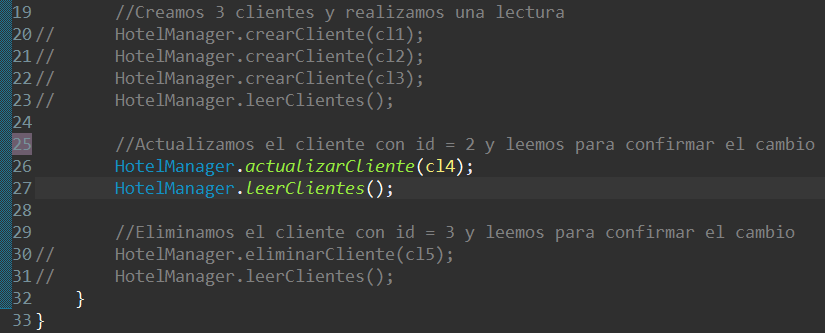


Si realizamos un SELECT en la BBDD, aparecerá la siguiente tabla con los datos introducidos anteriormente:

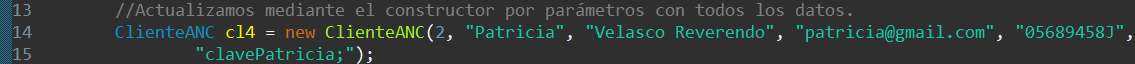


## 6.2.- Modificar y Leer cliente

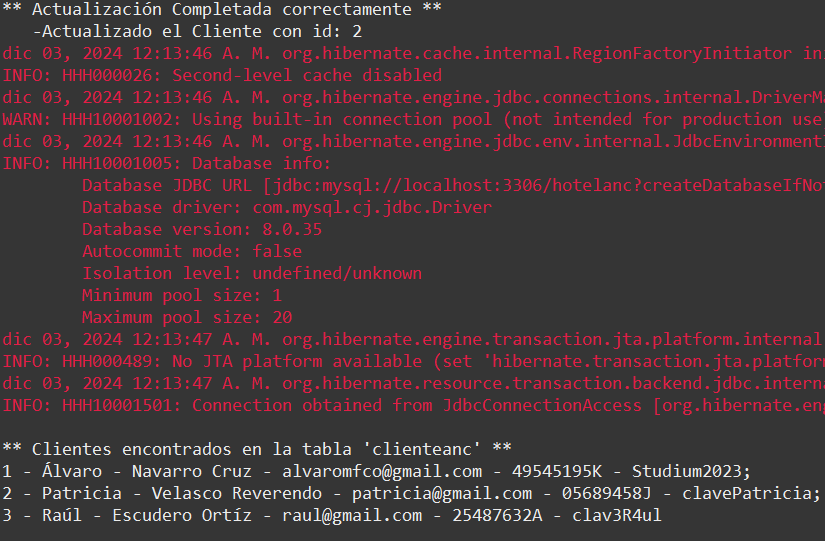
Para este paso, vamos a comentar las líneas de creación de usuarios y habilitamos las de modificación:



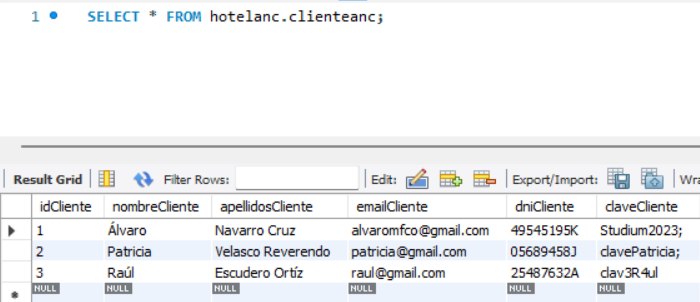
Esto debería actualizar los datos del cliente con id = 2 de la siguiente manera:



Al ejecutar el programa, se nos mostrará por consola el cliente con id 2 modificado con los valores pasados:

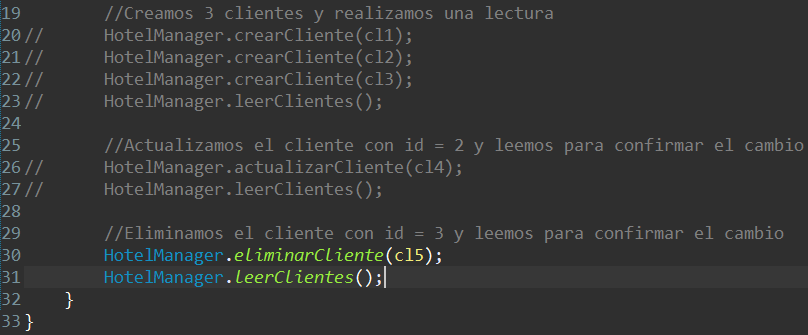


Si volvemos a realizar un SELECT en la BBDD, deberíamos confirmar que el cliente con id = 2 ahora tiene los datos modificados:



## 6.3.- Eliminar y Leer cliente

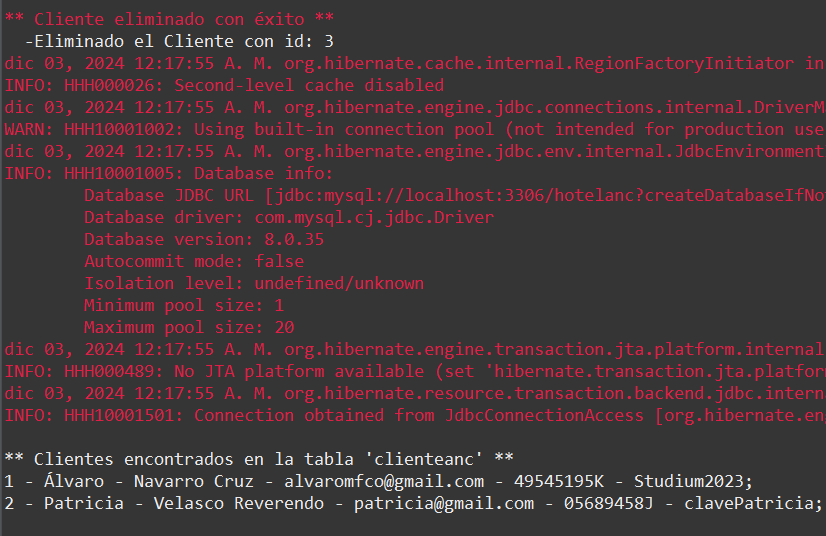
Para finalizar, vamos a comentar las líneas de modificación y habilitaremos las de eliminación:



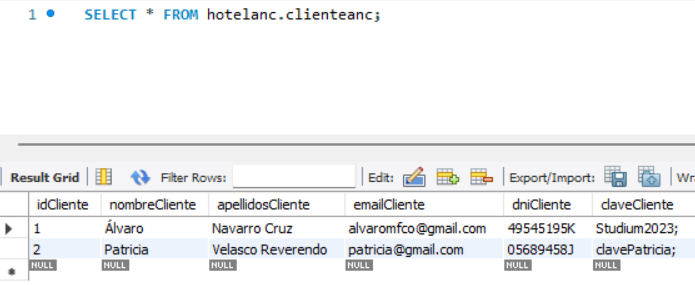
Esto debería eliminar el cliente con id = 3, según esta declaración:



Si ejecutamos el programa, nos aparecerá por consola un aviso de éxito y se nos mostrará la lista completa de los clientes en la tabla actualizada.



Si realizamos un SELECT en la BBDD, deberíamos comprobar que el cliente con id = 3 ha sido eliminado:



7.- Valoración Personal [(↑)](#indice)

Gracias al desarrollo de esta práctica, he conseguido afianzar los conocimientos vistos en la asignatura de Acceso a Datos, en particular, aquellos estudiados en el tema 3 de la asignatura.

Considero que el uso de Hibernate es más que satisfactorio, aportando rapidez y facilidad a la hora de realizar operaciones CRUD en Bases de Datos relacionales.

Tenía muchas ganas de probar este método por mi cuenta y, sinceramente, he salido más que satisfecho.

Todo lo estudiado en esta asignatura nos ha permitido ahorrar tiempo y recursos para realizar prácticas que ocuparían cientos de líneas de código en apenas 70 líneas como mucho, lo que me despierta curiosidad por explorar nuevas herramientas que faciliten el desarrollo de prácticas futuras.

8.-Bibliografía

Grupo Studium [(↑)](#indice)

1. *María José Martínez Navas, Acceso a Datos (Práctica Tema 3)*. Publicado en **Grupo Studium.**

Recuperado: <https://campustudium.com/pluginfile.php/15128/mod_resource/content/4/Tema%203%20-%20Herramientas%20de%20Mapeo%20Objeto-Relacional.%20Hibernate%20-%20Pr%C3%A1ctica_v2.pdf>

Último acceso (03/12/2024).