Sede Regional Brunca, Campus Coto.

Pruebas Unitarias del Segundo Proyecto Programado en Java.

programación II

Profesor:

Juan Gamboa Abarca

Ingeniería en Sistemas de la Información.

I Ciclo 2024

Estudiante:

Katherine Giselle Guatemala Barrientos

7 de junio 2024

**Tabla de Contenidos**

[Introducción 3](#_Toc169042106)

[Objetivos 3](#_Toc169042107)

[Casos De Prueba 4](#_Toc169042108)

[Oracle XE, MySQL Developer 4](#_Toc169042109)

[1. Crear Tabla e insertar Objetos 4](#_Toc169042110)

[2. Modificar o Editar Datos 5](#_Toc169042111)

[3. Búsqueda Por Campo (Leer) 6](#_Toc169042112)

[4. Seleccionar Todo 7](#_Toc169042113)

[5. Eliminación de Objetos en la Tabla 7](#_Toc169042114)

[6. Eliminación de una Tabla 8](#_Toc169042115)

[Mongo BD 9](#_Toc169042116)

[1. Crear Colecciones e Introducir Datos 9](#_Toc169042117)

[2. Editar Elementos de la Colección 10](#_Toc169042118)

[3. Leer Todos Los Objetos 11](#_Toc169042119)

[4. Eliminar un objeto 11](#_Toc169042120)

[My SQL 12](#_Toc169042121)

[1. Crear Tabla – Objetos 12](#_Toc169042122)

[2. Actualizar Datos 13](#_Toc169042123)

[3. Leer Todos los datos 14](#_Toc169042124)

[4. Buscar un objeto por campo (leer) 14](#_Toc169042125)

[5. Eliminar objetos 15](#_Toc169042126)

[Conclusión 16](#_Toc169042127)

# **Introducción**

Este documento presenta la implementación de pruebas unitarias a un modelo CRUD no relacional, que permite la integración y el uso de tres sistemas de bases de datos distintos simultáneamente, como lo son Oracle XE, MongoDB y MySQL. En este sentido, Oracle XE es una base de datos relacional que se distingue por su capacidad para soportar transacciones complejas. MongoDB es una base de datos NoSQL que se basa en un modelo de documentos destacando por su flexibilidad y escalabilidad, siendo una opción especialmente para grandes volúmenes de datos no estructurados. Por otro lado, MySQL es una de las bases de datos relacionales más populares del mundo, conocida por su facilidad de uso, rendimiento y compatibilidad.

Se ha desarrollado un modelo CRUD que puede funcionar de manera eficiente con tres sistemas de bases de datos, lo que permite a los desarrolladores realizar operaciones consistentes en esos sistemas de bases de datos, al mismo tiempo que brinda soporte para entornos de bases de datos como la de Oracle XE, que es Oracle SQL developer; MongoDB Compass, para MongoDB y MySQL Workbench, para MySQL.

Las pruebas unitarias realizadas garantizarán que cada componente funcione correctamente y que cumpla con los requisitos especificados ya que las pruebas unitarias son esenciales para garantizar la calidad y confiabilidad del software y para detectar y corregir errores tempranos.

Objetivos

1. **Integración en Entornos de Desarrollo**: Garantizar la compatibilidad del modelo con diferentes entornos de desarrollo, incluyendo MongoDB Compass para MongoDB y MySQL Workbench para MySQL.
2. **Implementar Pruebas Unitarias**: Diseñar y ejecutar pruebas unitarias para validar cada componente del modelo CRUD, asegurando su correcto funcionamiento y rendimiento.
3. **Evaluar la Interoperabilidad**: Probar y demostrar la capacidad del modelo para operar simultáneamente con las tres bases de datos, manteniendo la integridad y consistencia de los datos

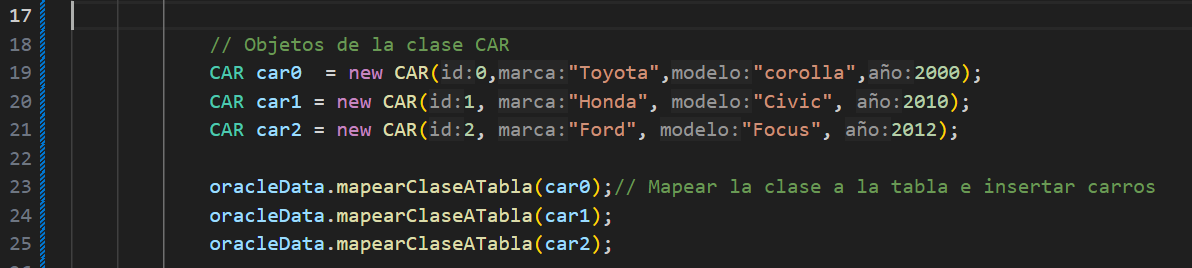
Casos De Prueba

Para este apartado se están probando las tres conexiones a bases de datos distintas como lo son, Oracle MySQL Developer, Mongo DB en su API Mongo Compass, y en MySQL con su respectiva API MySQL Workbench.

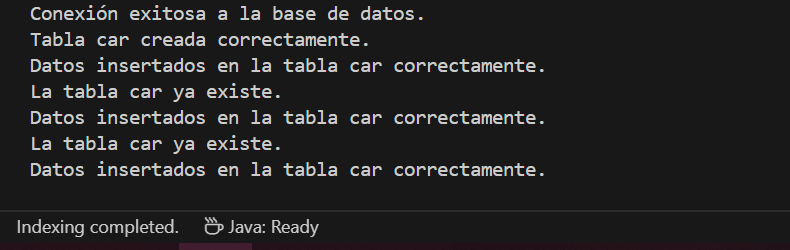
Oracle XE, MySQL Developer

1. Crear Tabla e insertar Objetos

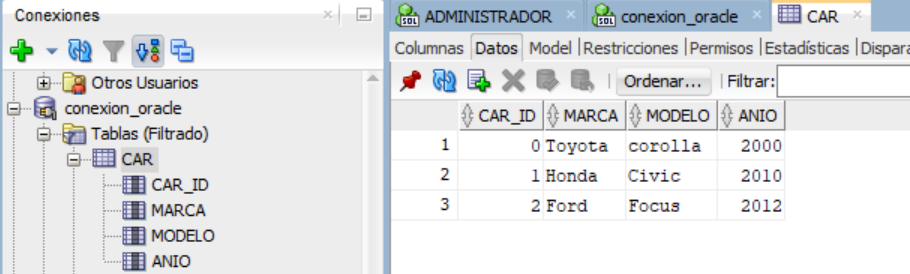
Verifica si la tabla correspondiente al objeto existe, y si no, crea la Tabla con el nombre de la clase correspondiente al objeto e inserta o mapea los datos del objeto en la tabla. Así se ve el código.



Compilación del proyecto en la terminal de VS code.

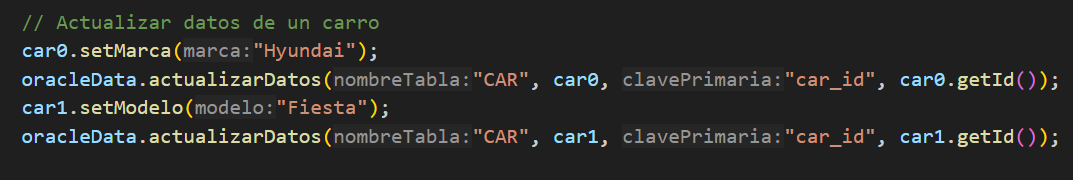


Este es el resultado de la ejecución de la prueba en el entorno de desarrollo de Oracle.

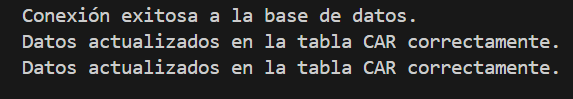


1. Modificar o Editar Datos

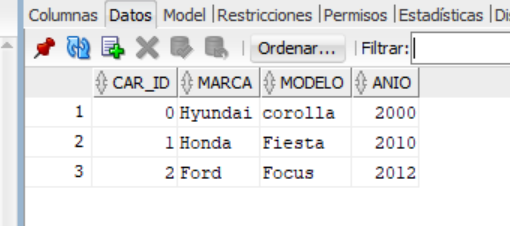
Se actualizan los objetos de un registro ya existente en la base de datos, necesita de los setters y getters de la clase y el objeto solo se modifican con su identificador único (Primary key).



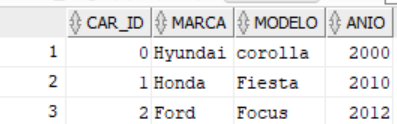
Salida en la terminal.



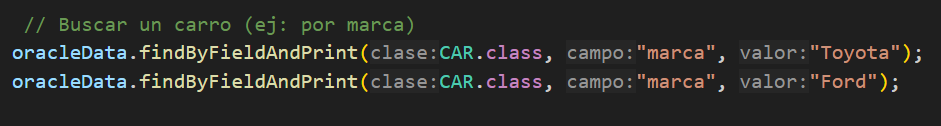
Resultado de la de la prueba en la Base de Datos, con los objetos CAR modificados.



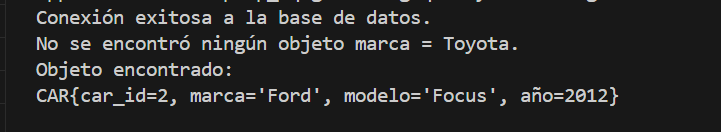
1. Búsqueda Por Campo (Leer)

Recupera un registro específico de la tabla, si existe; y esté identificado por cualquier campo del nombre de la Tabla creada. Esta es la Tabla que se usará para las pruebas:

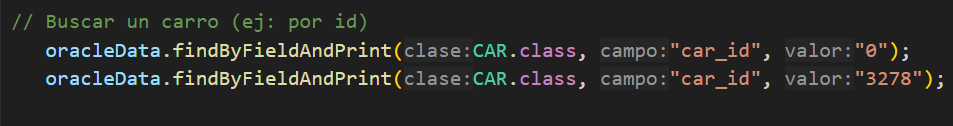
Primer ejemplo de búsqueda por el campo de tipo “marca”.



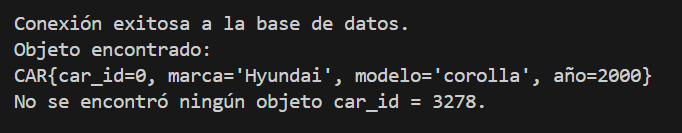
Salida por consola.



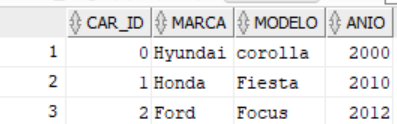
Segundo ejemplo de búsqueda por el campo de tipo “id”.



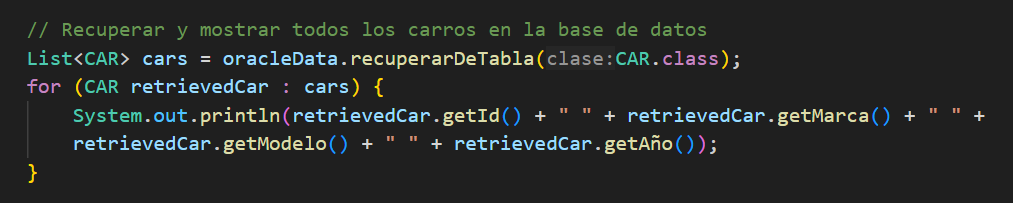
Salida por consola.



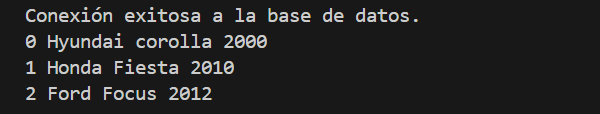
1. Seleccionar Todo

Recupera todos los registros de una tabla específica y muestra a cada uno por consola. Esta es la Tabla que se usará para la prueba.

Recuperación de los objetos de la tabla CAR, mediante una lista.

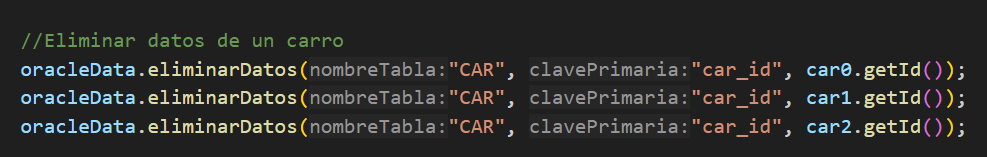


Salida en consola.

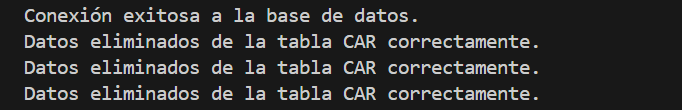


1. Eliminación de Objetos en la Tabla

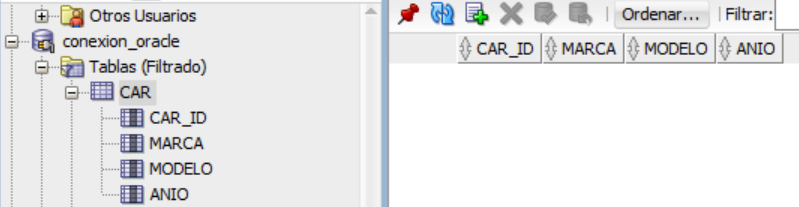
Se elimina un registro específico de la tabla, con su ID, el identificador único.



Salida en consola.

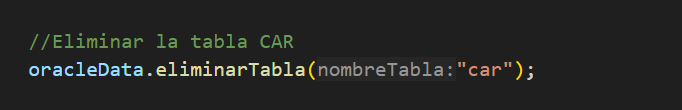


Resultado de la de la prueba en Oracle, con los objetos CAR eliminados.

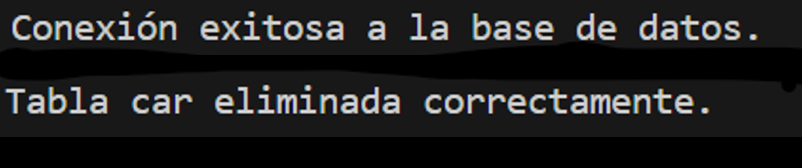


1. Eliminación de una Tabla

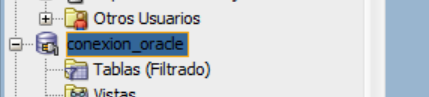
Se elimina la Tabla y todo el contenido interno que poseía la tabla. Solo se debe colocar el nombre de la tabla en minúsculas.



Salida en consola.



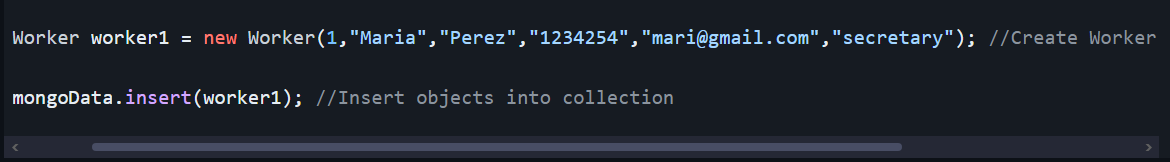
Prueba de eliminación completa de la Tabla CAR.



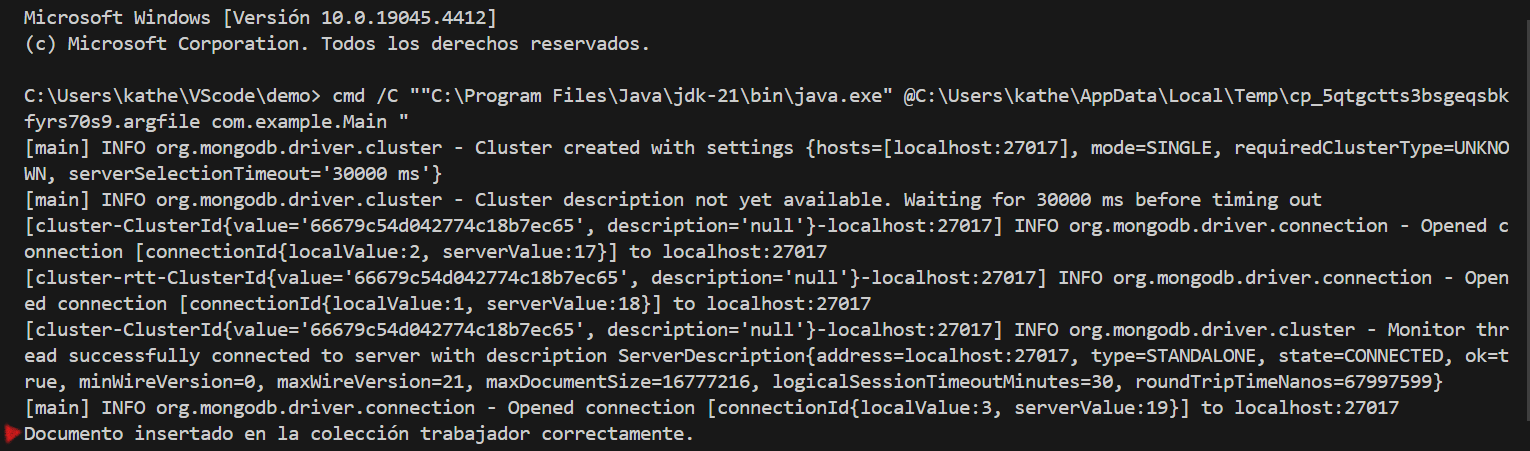
Mongo BD

1. Crear Colecciones e Introducir Datos

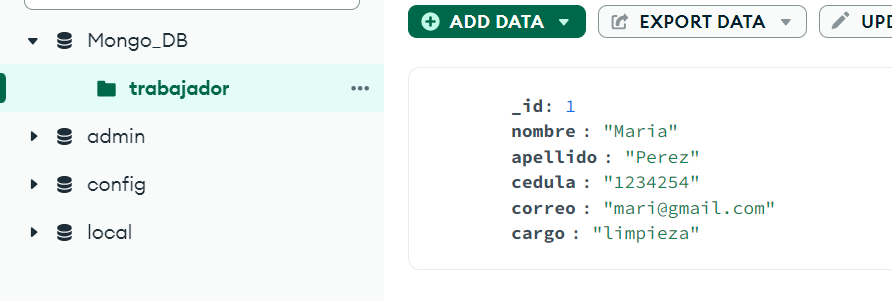
Verifica si la colección correspondiente al objeto existe, y si no, la crea con el nombre de la clase correspondiente al objeto y mapea los objetos a la misma. Así se ve el código.



En Consola, Mongo podría parecer que te está generando errores con este gran texto en la terminal, pero es totalmente normal, y siempre aparece.

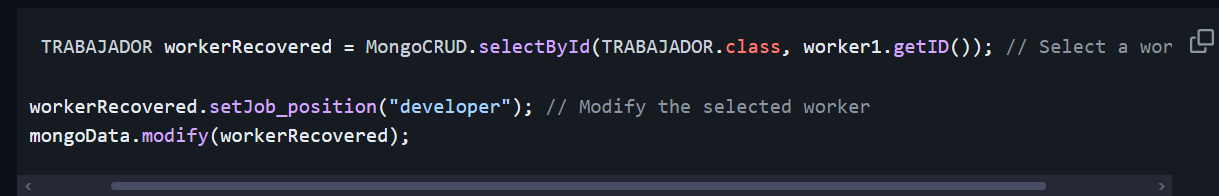


Este es el resultado de la ejecución de la prueba en el entorno de desarrollo de Mongo DB.

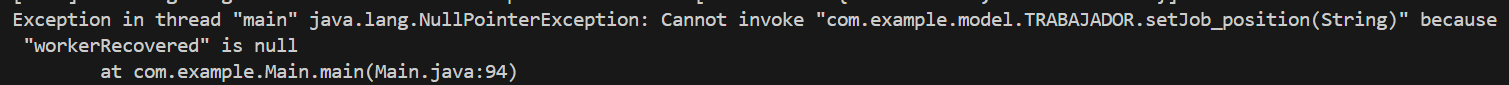


1. Editar Elementos de la Colección

Se actualizan los objetos de un registro ya existente en Mongo, necesita de los setters y getters de la clase (colección). Los objetos solo se modifican con su identificador único (Primary key).



Mensaje en consola de error, si el objeto que se desea actualizar no existe (basado en su ID).



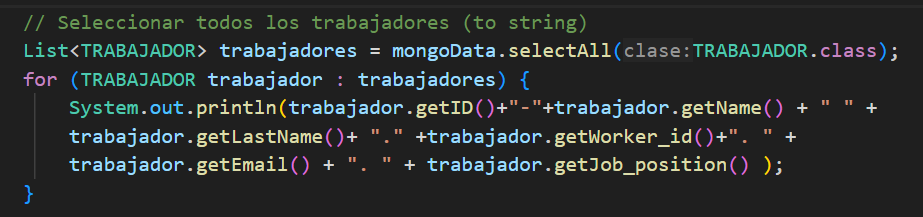
Consola: Mensaje exitoso.  


Objeto “Worker” modificado en la base de datos, se actualizó el cargo del objeto 

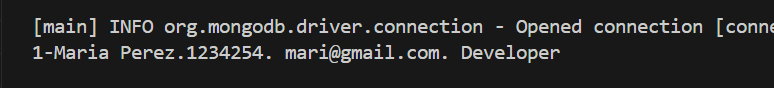
1. Leer Todos Los Objetos

Recupera el registro de toda la colección, y muestra en consola cada uno de los objetos. Esta es la Tabla que se usará para la prueba:



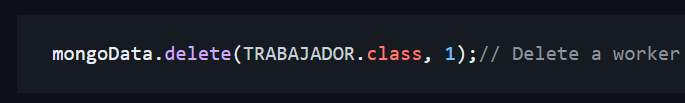


Consola.



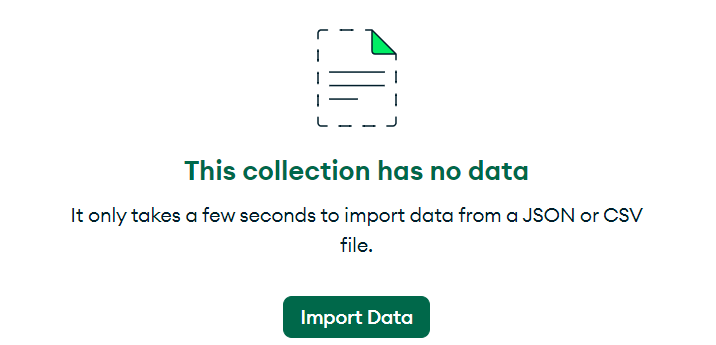
1. Eliminar un objeto

Se elimina un Documento específico de la colección, con su ID ( el identificador único).



Terminal de VS code.



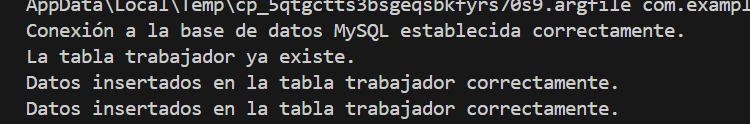
Eliminación reflejada en Mongo DB Compass  


My SQL

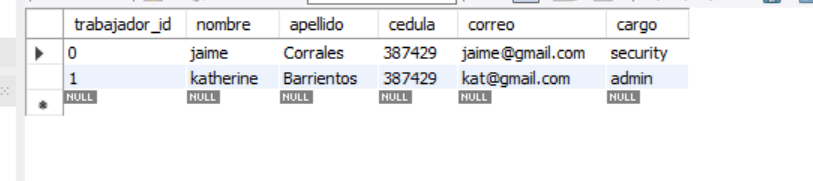
1. Crear Tabla – Objetos

Verifica si la Tabla correspondiente al objeto existe, y si no, crea con el nombre de la clase correspondiente al objeto y mapea los objetos a la misma. Así se ve el código.



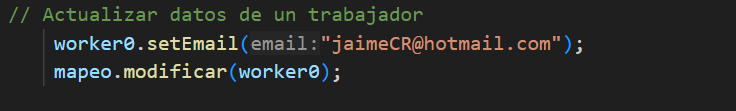
Consola: Ya la Tabla existía, solo se insertaron los dos objetos  


Este es el resultado de la ejecución de la prueba en el entorno de desarrollo de MySQL Workbench.



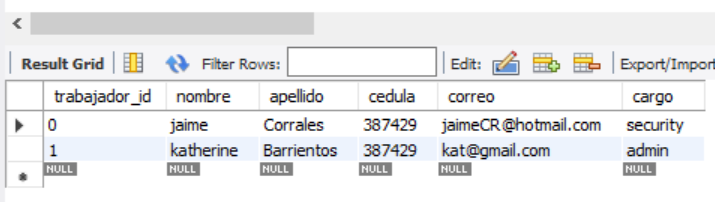
1. Actualizar Datos

Se actualizan los objetos de un registro ya existente en la base de datos, necesita de los setters de la clase, y el objeto solo se modifica con su instancia del objeto.



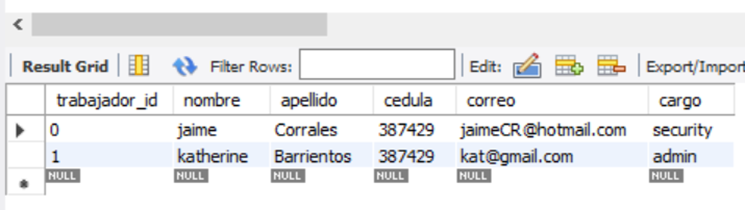


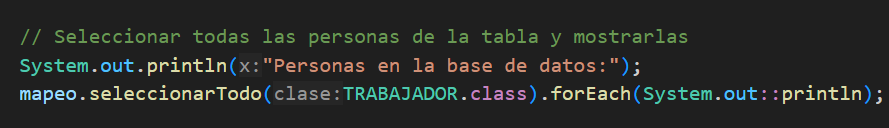
Resultado de la de la prueba My SQL, con los objetos Trabajador modificados.

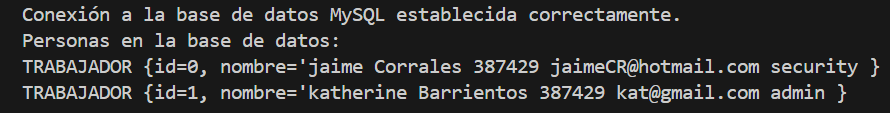


1. Leer Todos los datos

Recupera los registros de la tabla, si existe, y muestra por consola todos los objetos encontrados. Esta es la Tabla que se usará para las pruebas:



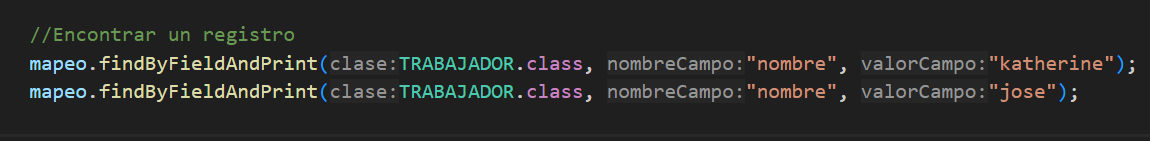




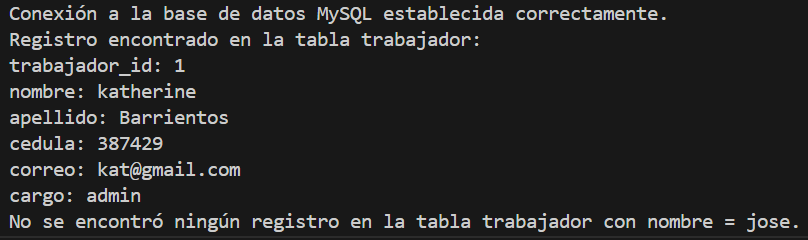
1. Buscar un objeto por campo (leer)

Recupera un registro específico de la tabla, si existe; y esté identificado por cualquier campo del nombre de la Tabla creada. La Tabla usada para la prueba está en el pto 3.

Ejemplo de búsqueda por el campo de tipo “nombre”.

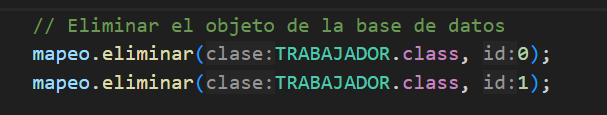


En la primera línea se colocó un nombre que sí existe en la tabla con el campo “nombre”, mientras que en la segunda línea de código se puso a propósito un nombre que no existe en dicha tabla. Y en la consola se muestran los 2 mensajes.



1. Eliminar objetos

Se elimina un registro específico de la tabla, con su ID, el identificador único del objeto.



Consola.

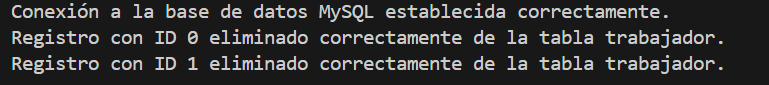
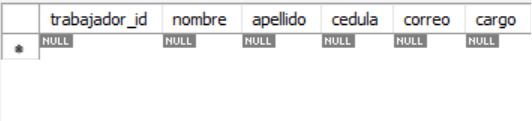


Tabla Trabajador al eliminar sus objetos.



Conclusión

Este proyecto ha demostrado la viabilidad de integrar diferentes sistemas de bases de datos en una única aplicación, aprovechando las ventajas de cada uno. Las pruebas unitarias han sido fundamentales para garantizar la calidad del modelo y detectando así algunos errores, asegurando que el modelo se integra bien con varios entornos de desarrollo y pueda funcionar simultáneamente desde VS code.