**FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION**

**DESARROLLO DE APLICACIONES WEB**

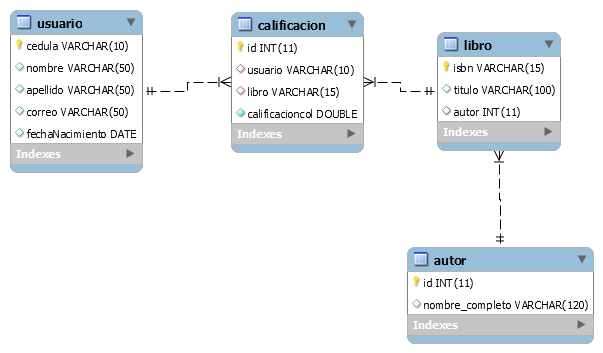
**I TÉRMINO 2019**

**TALLER 5**

|  |  |
| --- | --- |
| **Grupo:** | Master Developers |
| **Integrantes:** | Karla Burgos, Carlos Sesme, Tony Veas, Jonathan Quintana |
| **Paralelo:** | 2 |

**Reporte**

1. Modelo lógico

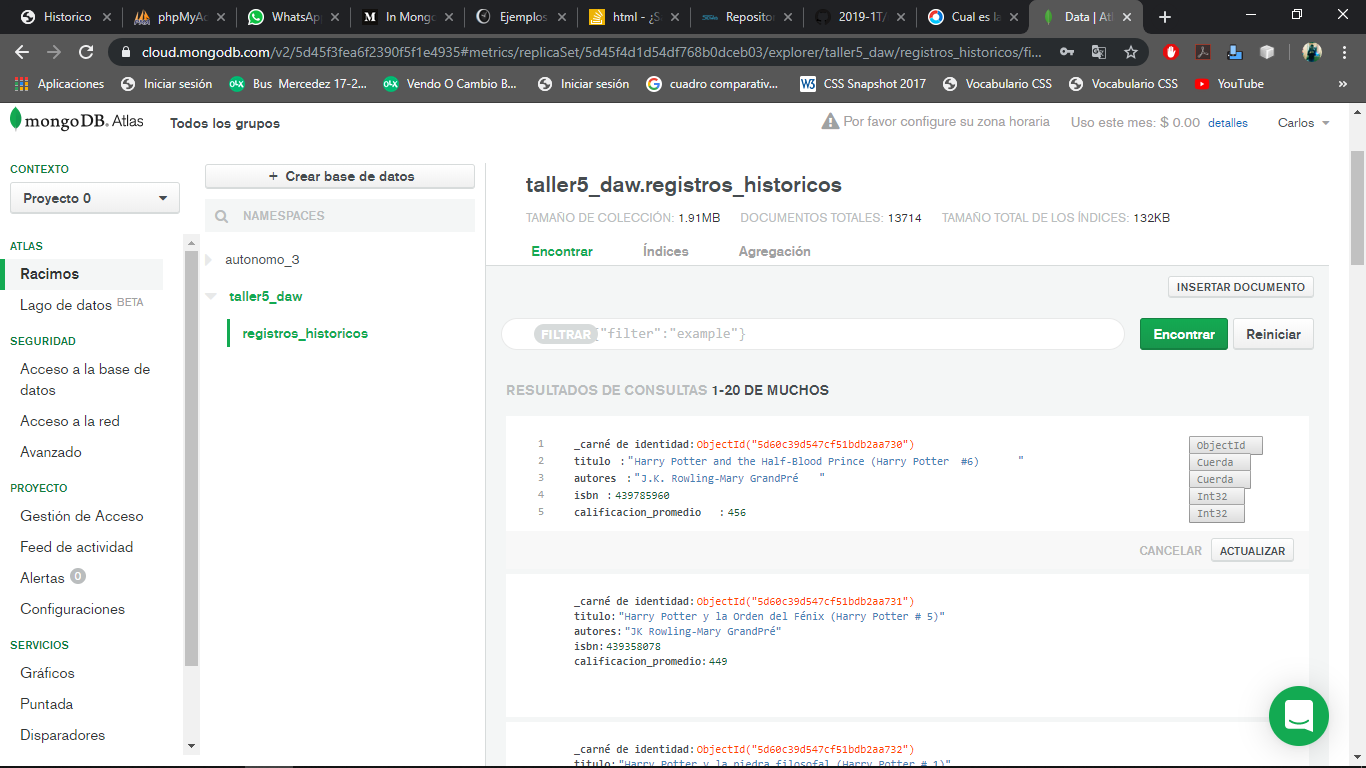


1. Descripción del API REST

Se creó una api básica para poder obtener la data almacenada en la base de datos, las rutas de las cuales consta son:

1. “/”, ruta principal en la cual se muestra formulario de login en esta se usa el método get.
2. “/login”, utilizando el método post es la encargada de realizar la autenticación en el sistema y verificar que el usuario sea válido.
3. “/logout”, utilizando el método post, esta ruta se encarga de cerrar la sesión iniciada previamente.
4. “/panel”, método get, muestra la página principal en la cual se enlaza ambas bases y renderiza el template de panel de administración de la página.
5. "/loadautores", método get, se utiliza para pasar la información de la base de datos no relacional a la relacional.
6. "/autores", método get, devuelve todos los autores que están almacenados en la base para poder llenar el combo de la página principal.
7. "/libautor/:autor", método get, devuelve los registros de libros del autor que recibe por parámetro en url.
8. "/notas/:cedula", método get, obtiene todos los libros calificados por el autor y renderiza el template asignado.
9. "/calificar", método post, recibe la nota del usuario y la guarda en la base de datos.
10. **Descripción de la Base de Datos no relacional: estructura, motor de la noRBDMS.**

Para la base de datos no relacional se usó el motor MongoDB Atlas, la estructura de la colección utilizada está relacionada al archivo histórico.txt provisto para esto se pasó la data de este archivo en un csv y se realizó la importación a MongoDB:



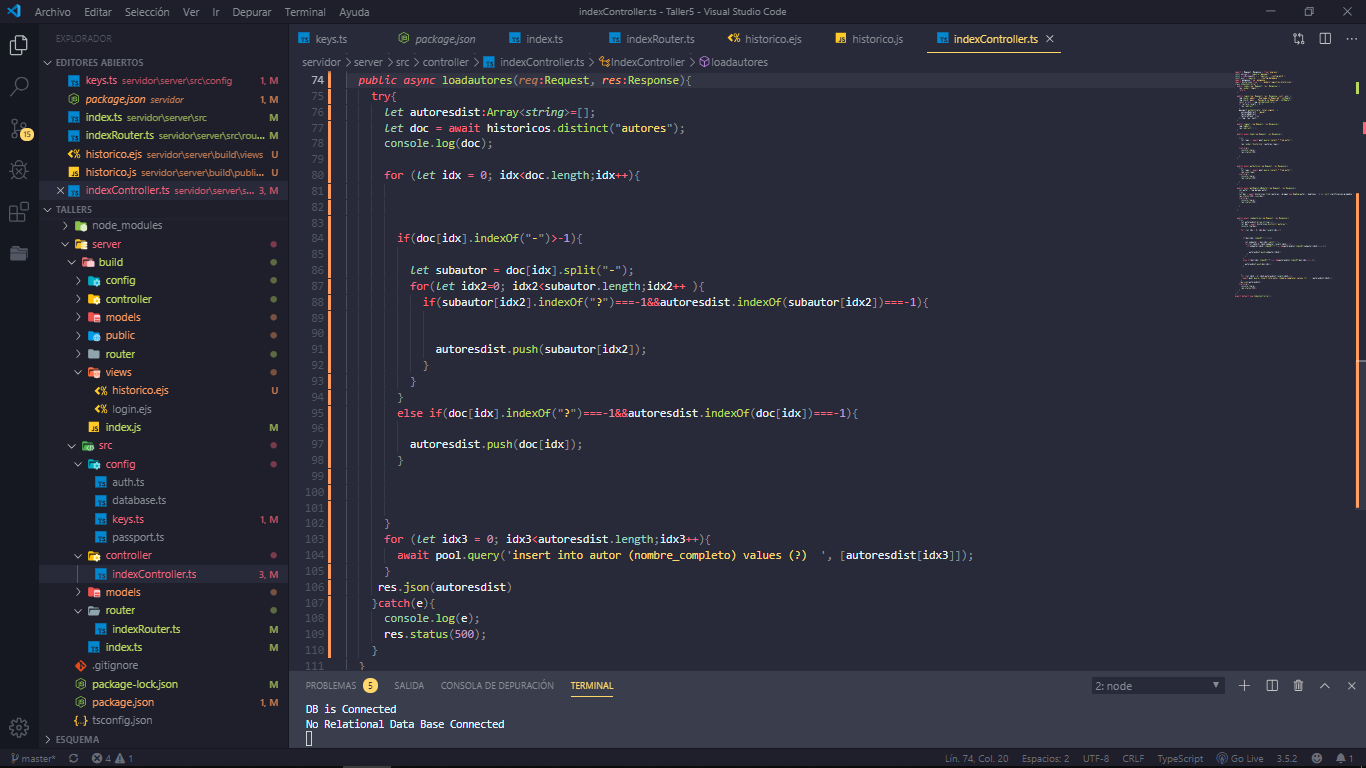
1. **Evidencia de la implementación de los requerimientos Req1, Req2 y Req3.**

Evidencia del requerimiento 1:

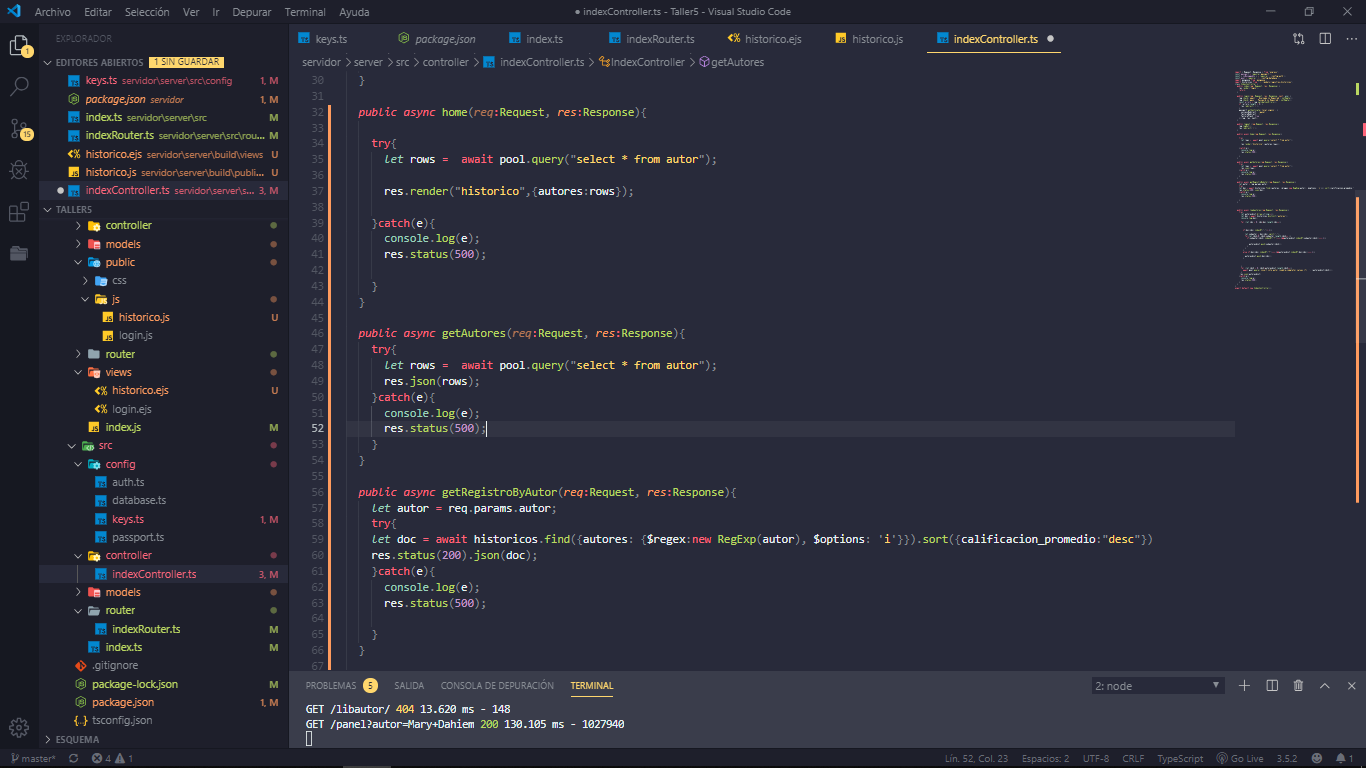
Para cumplir con el primer requerimiento lo primero que se realizo fue la importación de los registros históricos a la base de datos no relacional mostrados anteriormente.

Después se realizó la extracción de los autores de los libros desde la base de datos no relacional y se los registro en la base de datos relacional en su respectiva tabla.

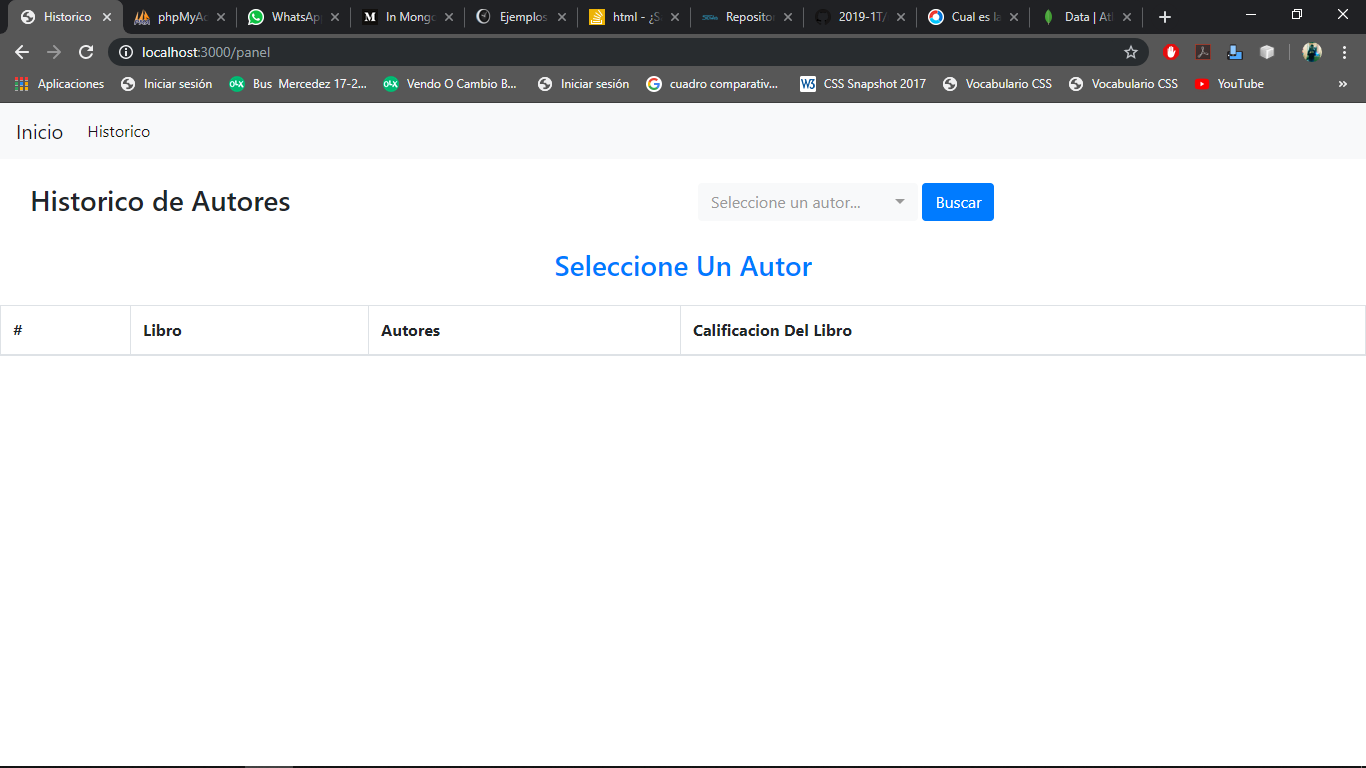
Para esto se creó un servicio dentro del api con el cual se podría realizar este proceso teniendo en cuenta que cada libro podría llegar a tener más de 1 autor por lo cual se debía tener cuidado al realizar los registros en la base de datos relacional para evitar nombres repetidos.



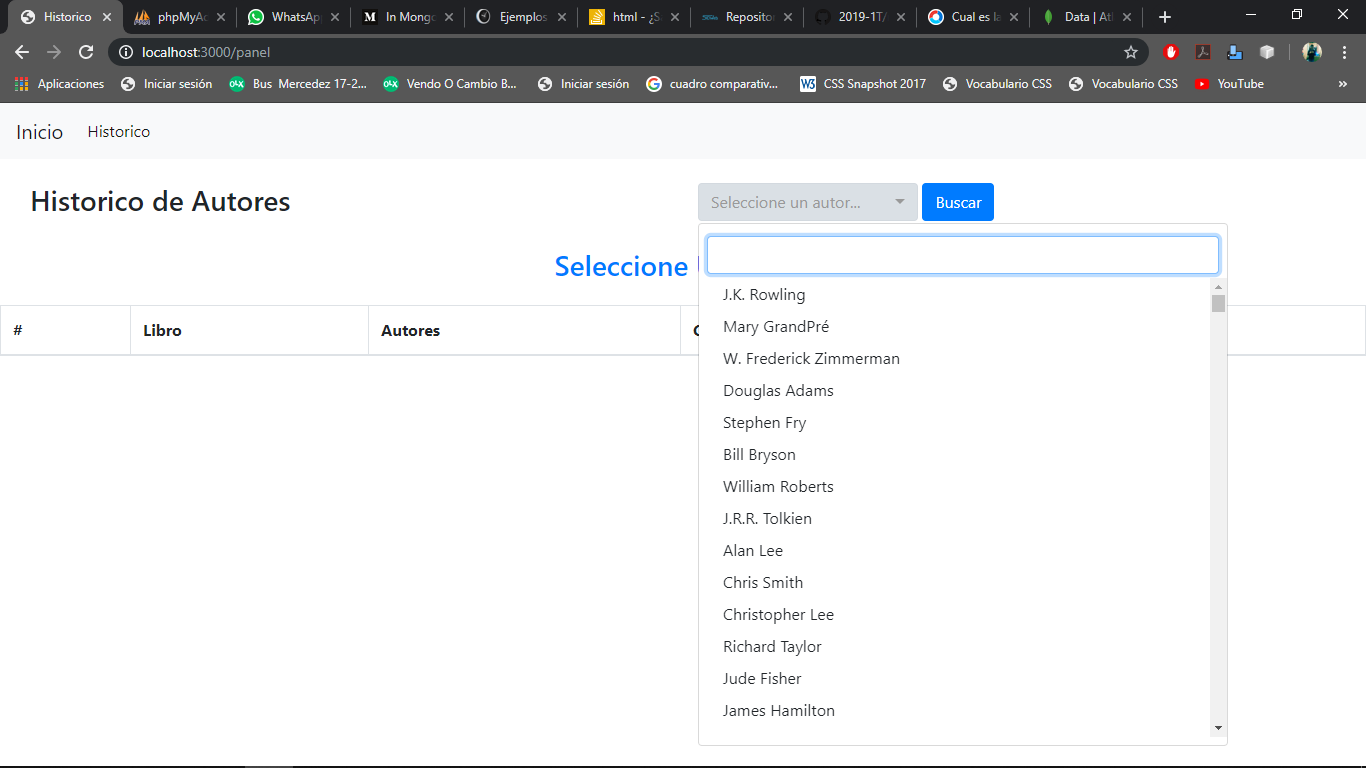
Una vez cargados los nombres de los autores a la base de datos relacional se planteó los servicios con los cuales el usuario podría seleccionar el autor y obtener todos los libros relacionales a el ordenados de acuerdo a su calificación:

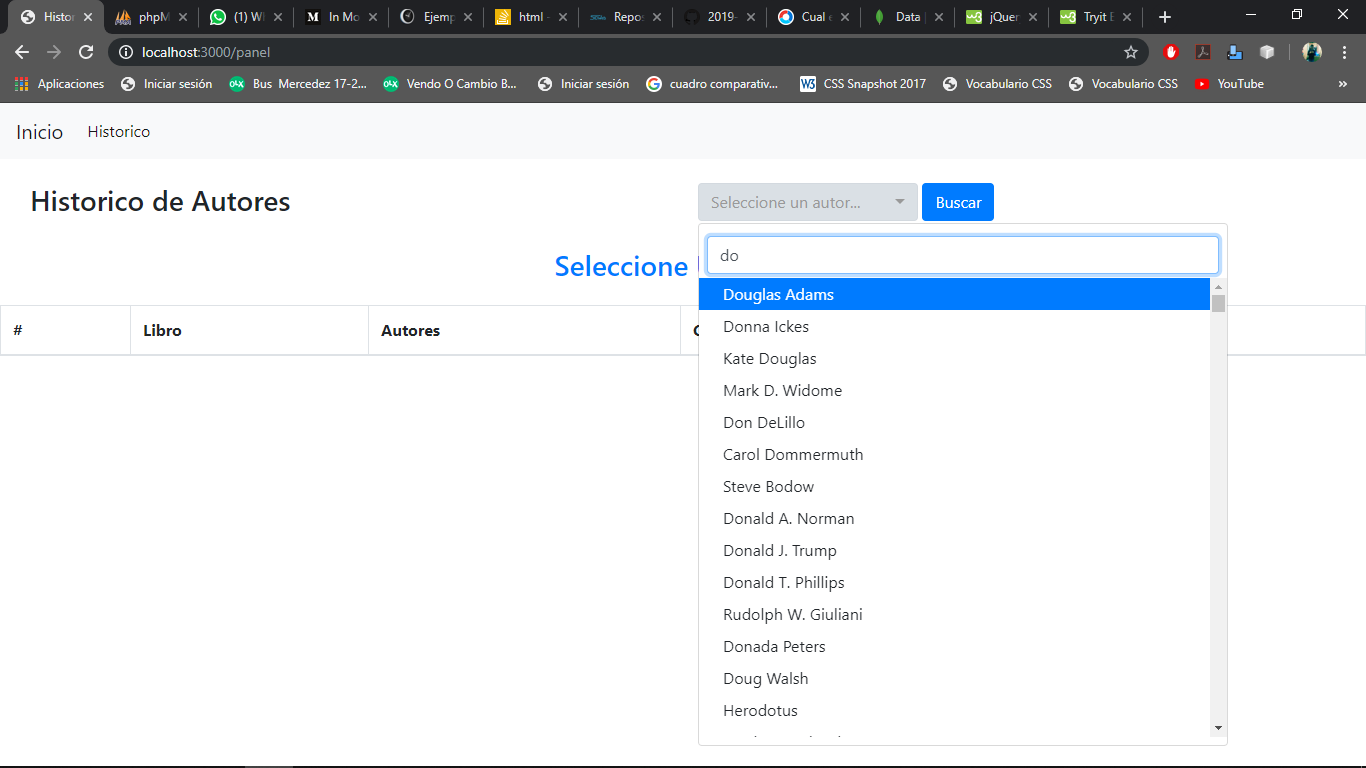


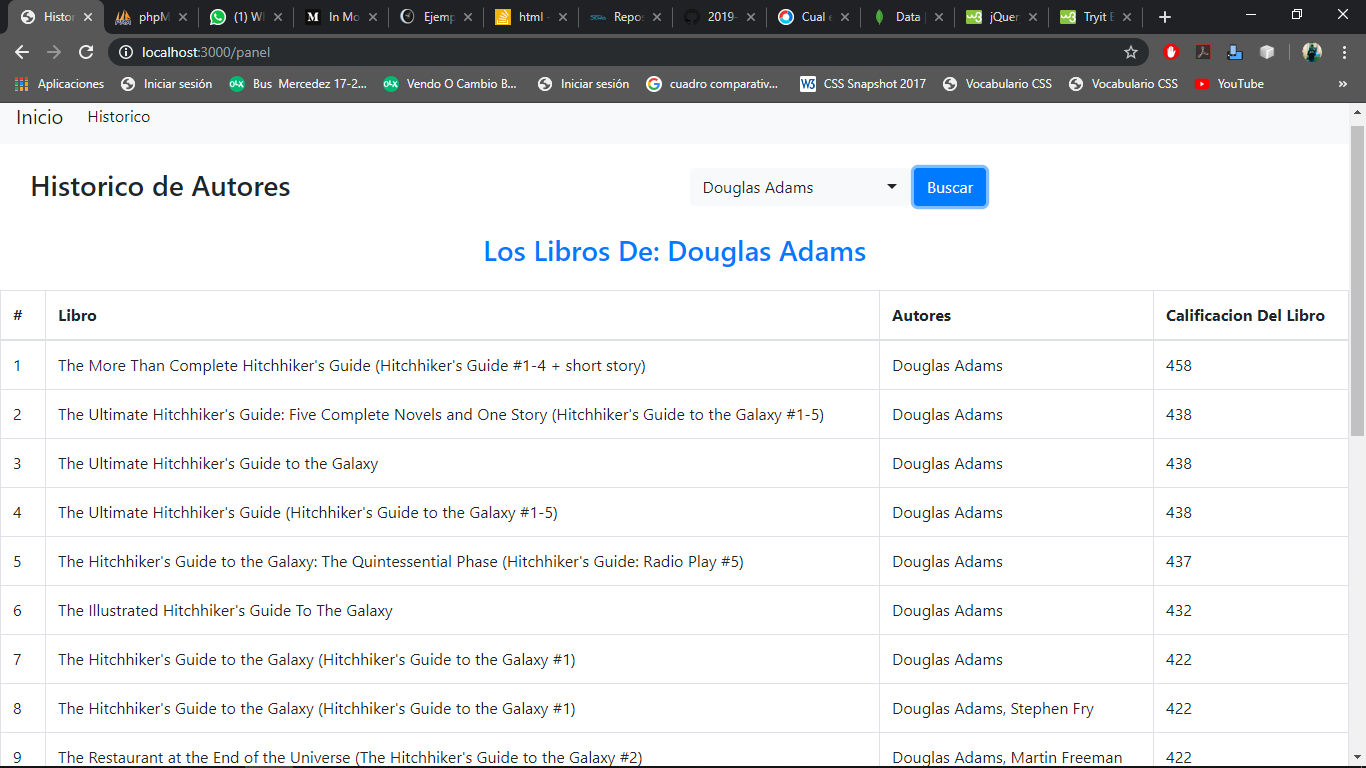
Como resultado visual se mostrar de la siguiente manera:

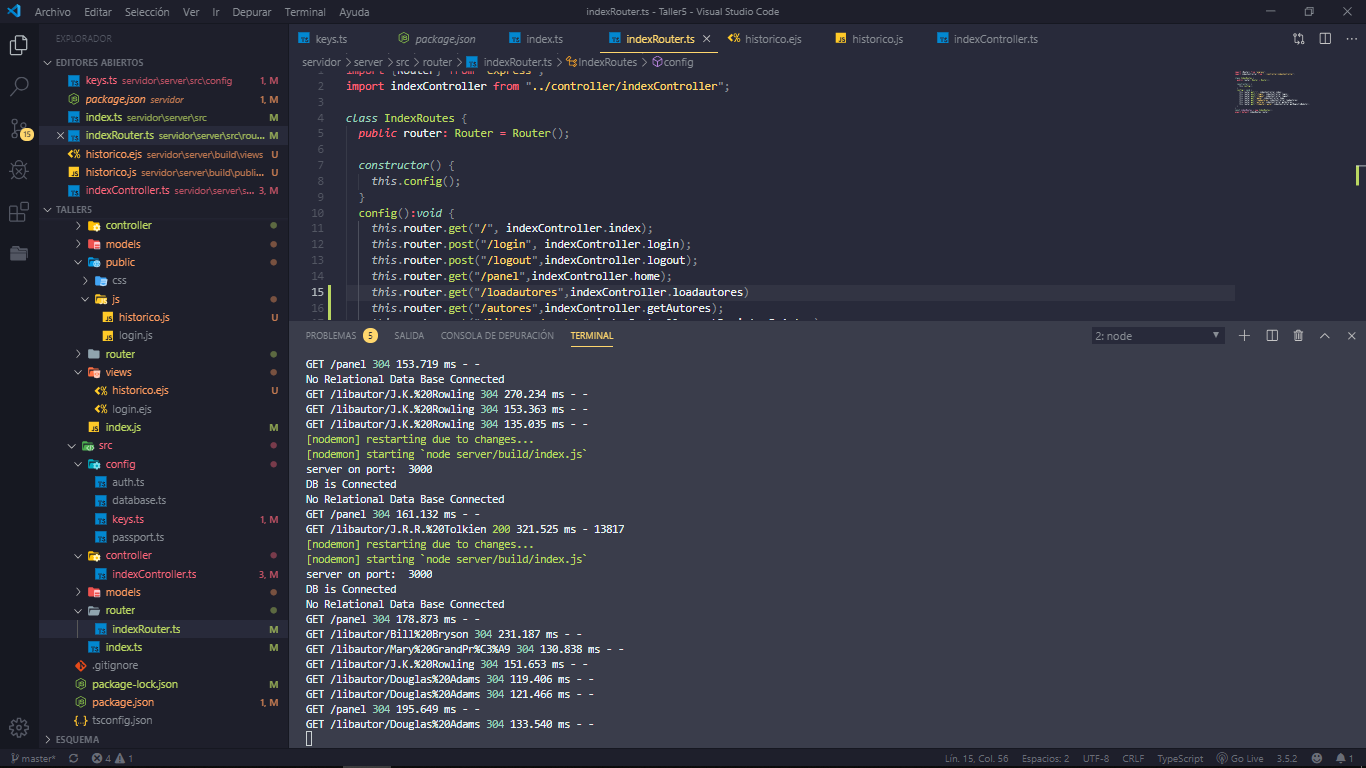


Los usuarios podrán seleccionar o buscar desde un search select box el nombre del autor y al realizar la búsqueda se mostrarán todos los libros de dicho autor contenidos en el historial (base no relacional) ordenados de acuerdo a su calificación en dicho historial:



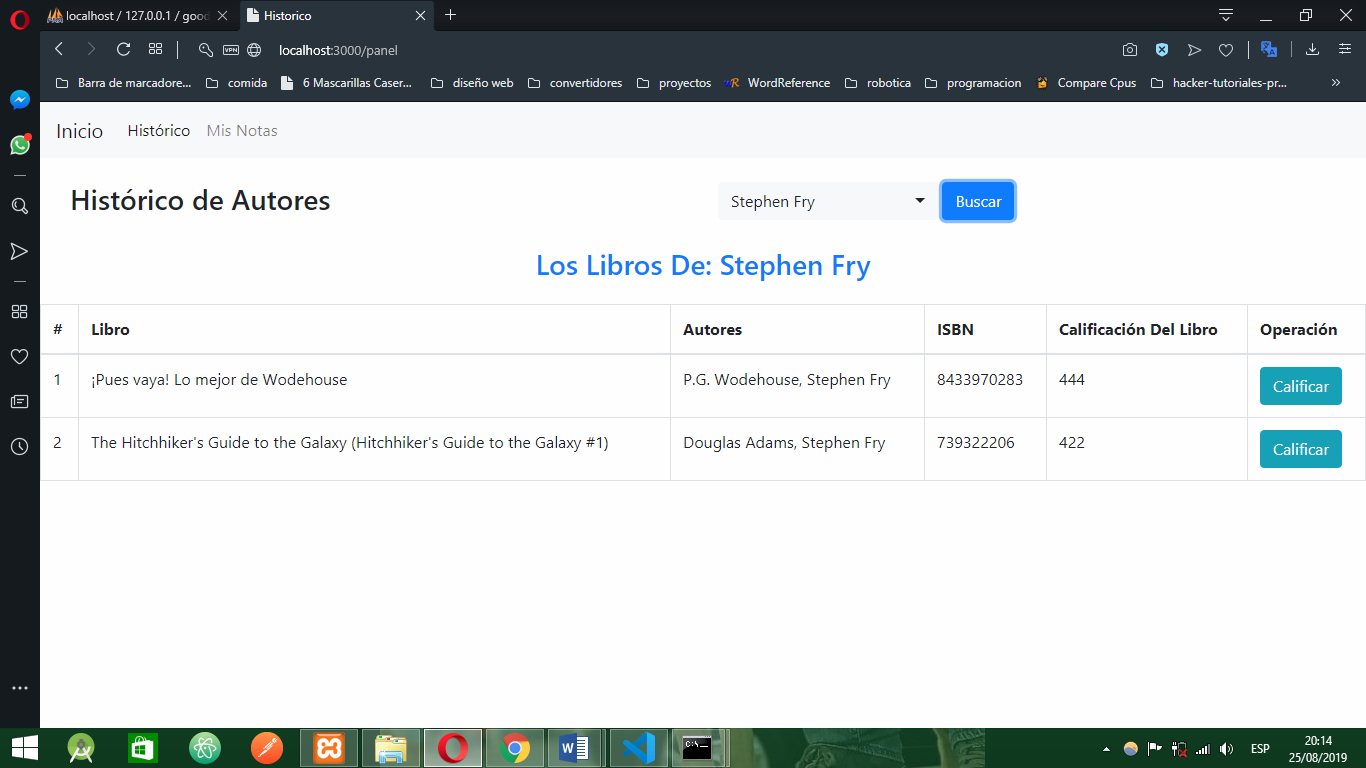




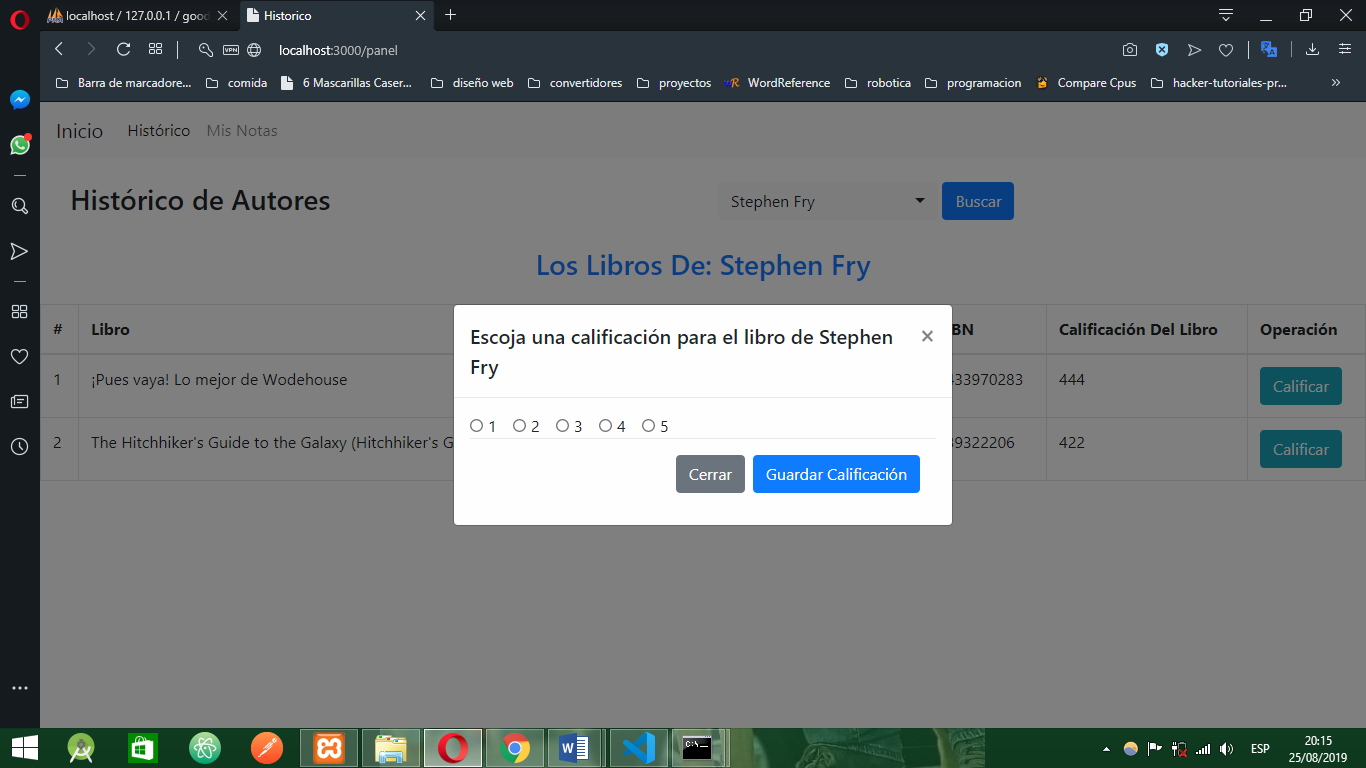


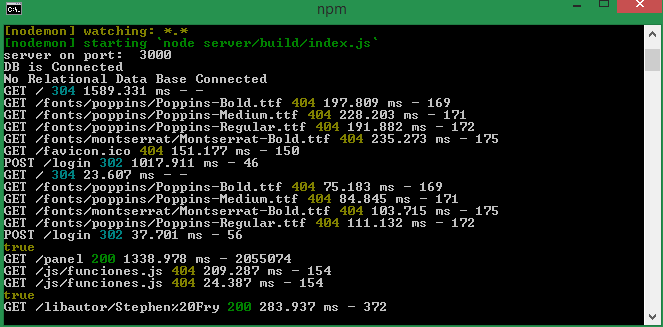
Requerimiento 2

Para proceder con este requerimiento se implementó una opción más en la pantalla principal, en este caso un botón que enlace hacia la sección de calificaciones.

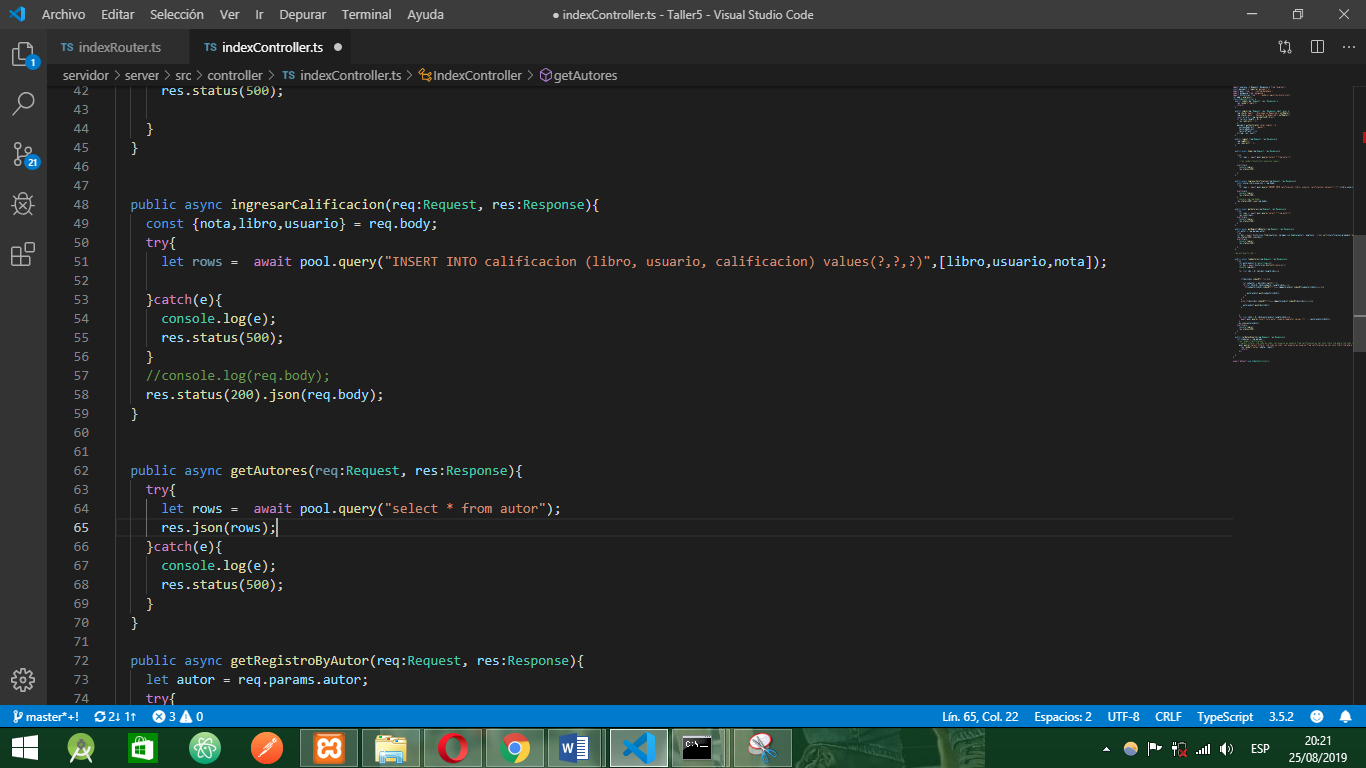


Y al presionar el botón de calificar se presenta un modal en el cual se procederá a calificar el libro en un rango el 1 al 5



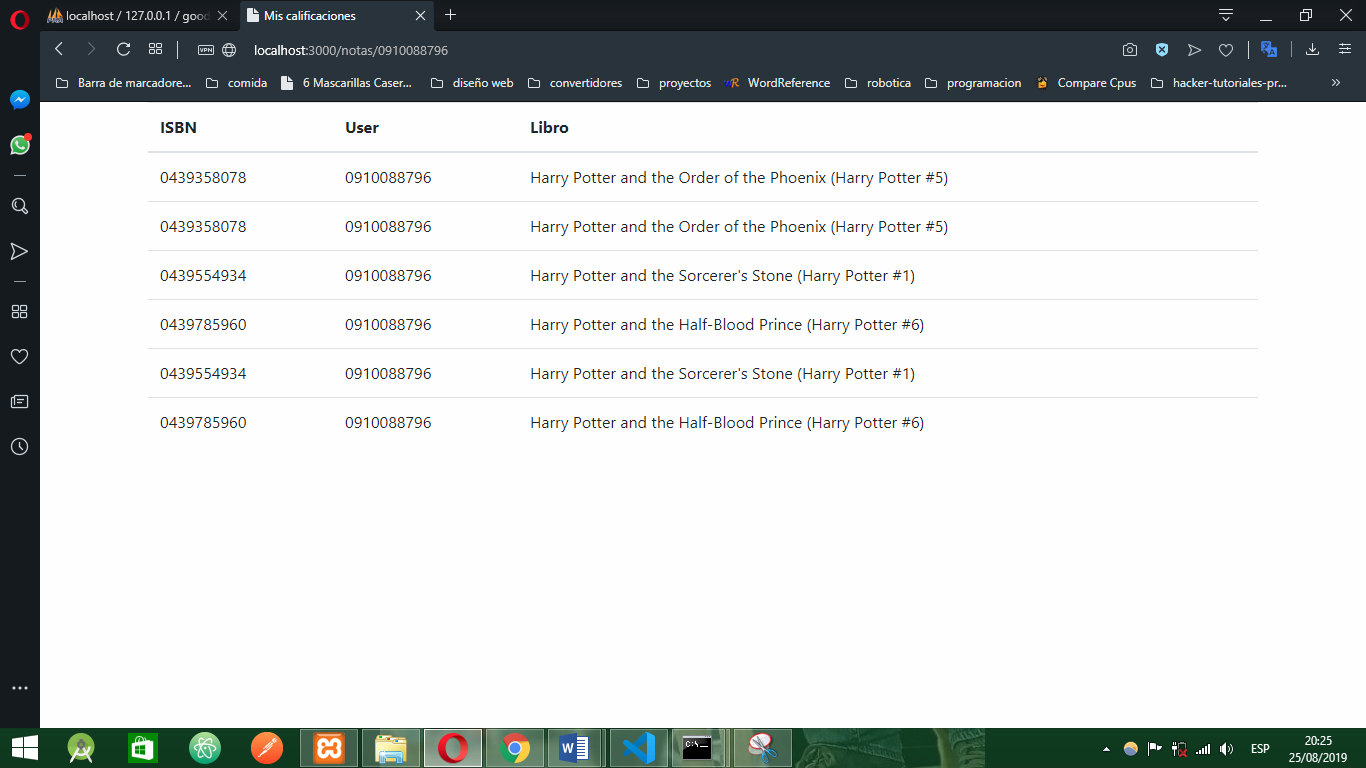


Código de la calificación



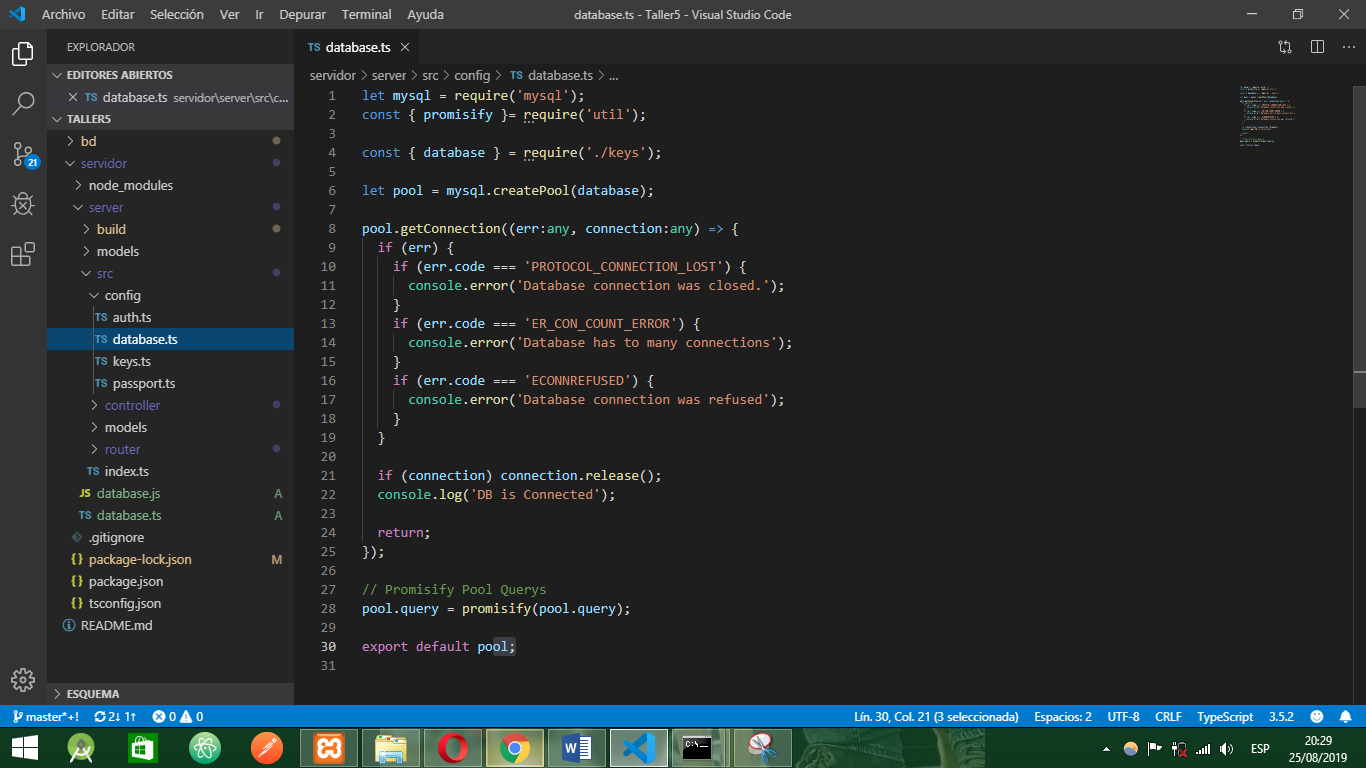
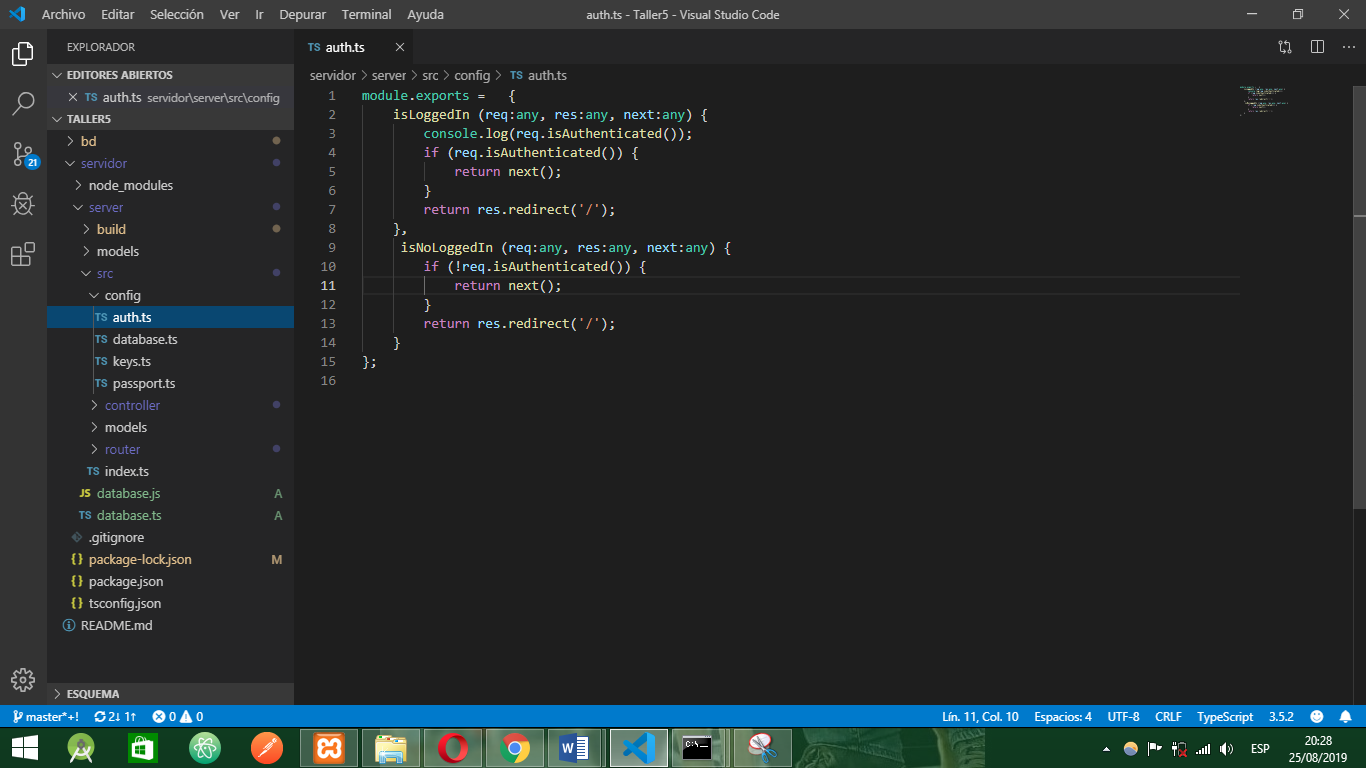
Requerimiento 3

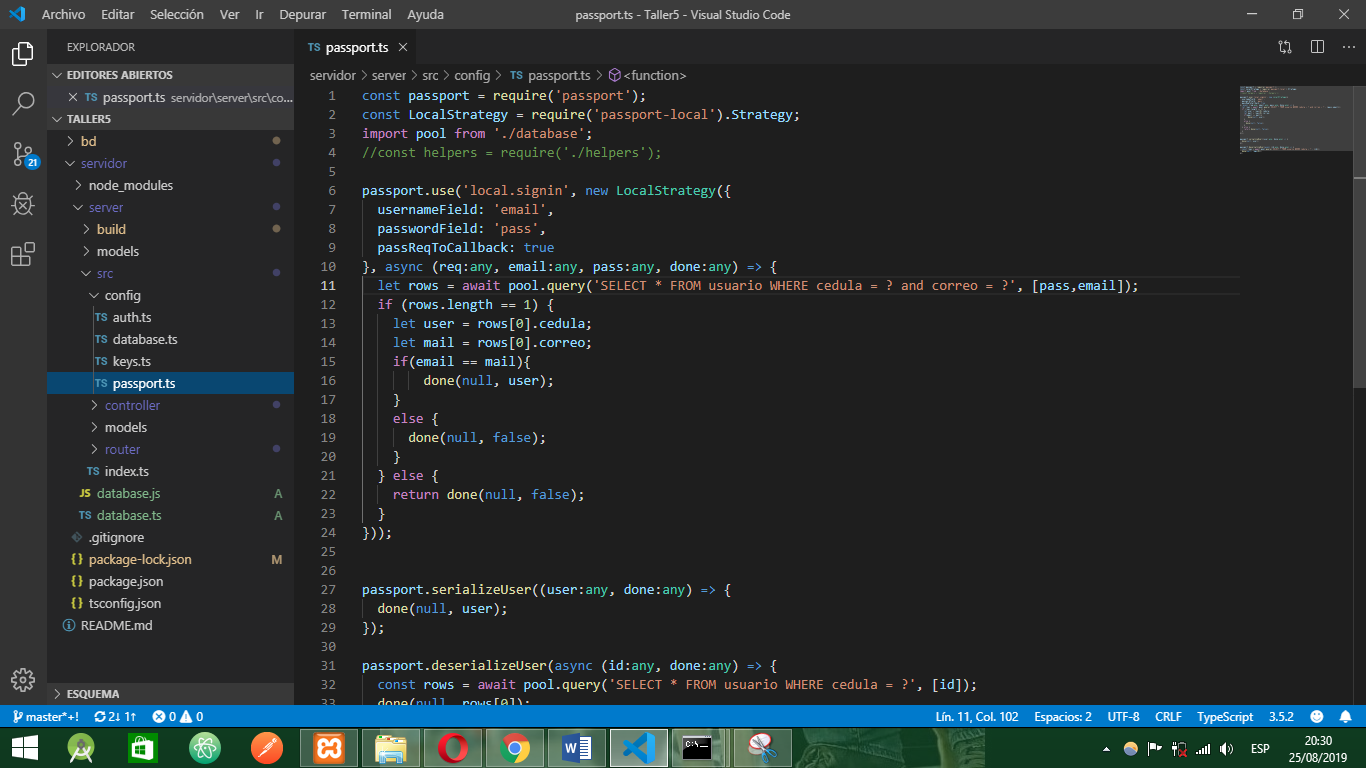
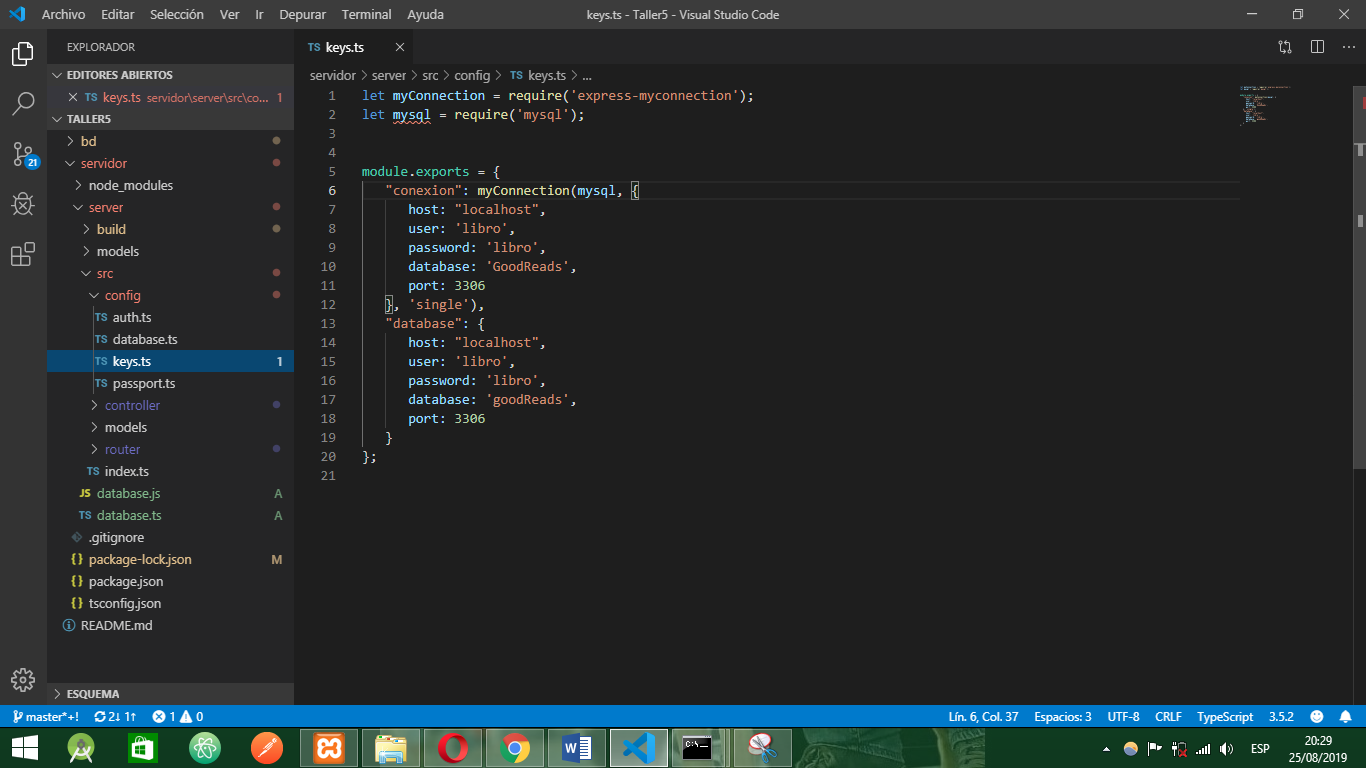
Para este se implementó una interfaz minimalista en la cual solo se mostrará los datos en una tabla con los detalles de todos los libros que el usuario haya calificado.



Esta sección necesita recibir por la url la cedula del usuario que se desea consultar, es algo poco seguro, pero se lo implemento debido a que son fines académicos y para tratar de hacerlo de una manera más sencilla esto se puede solucionar mediante una cookie de sesión extra ya que nuestro sistema de login se basa en un middleware de express el cual almacena las sesiones en la base de datos relacional.

Funcionamiento del login.





url del repositorio: <https://github.com/kbburgos/Taller5>