INGENIERÍA ESPECIALIZADA S.A.

GESTIÓN DE CONOCIMIENTO

PRUEBA TÉCNICA

DESARROLLO SOFTWARE



Medellín, diciembre de 2023

1. INFORMACIÓN PERSONAL.

Nombre Completo: Andrés Juan Gutiérrez Castro

1. ASPECTOS GENERALES

Antes de presentar las reglas de la prueba, agradecemos tu tiempo en el desarrollo de esta prueba técnica, así como cualquier observación para mejorar el proceso de selección. Esta prueba se calificará de la siguiente manera:

Valoración de 1 a 10 puntos donde:

* **Aspectos teóricos acumulan 3 puntos**
* **Aspectos prácticos acumulan 7 puntos.**

Se debe enviar este documento con tus respuestas al correo Jhefferson.rios@ieb.co en versión pdf.

Para la sección práctica, el código debe compartirse en un sistema de control de versiones (SCV) que prefiera, compartiendo al final de la prueba el acceso al mismo (si es github: Jhefferson.rios@gmail.com, azure repos: Jhefferson.rios@ieb.co, u otro).

**Valoraremos en particular tu capacidad argumentativa y recursiva,** por tanto, puedes explicar tu respuesta directamente, apoyados en ejemplos o gráficas según consideres necesario.

Puedes presentar tus respuestas y soluciones en el lenguaje y herramientas que prefieras.

En caso que encuentres dificultad en algún paso necesario para la solución, te pedimos redactes tu análisis y continúes con los pasos restantes.

L**a prueba fue diseñada para un tiempo de 2 días, en caso que desee refactorizar el código puede emplear más tiempo (1 semana).** El tiempo de entrega no será un factor considerado en la calificación, únicamente se valorará su código y argumentación de pasos.

Puede consultar dudas en internet, emplear plantillas que ya posea, y herramientas que prefiera para su desarrollo (indicando cuales y donde las usó), así mismo, puede solicitar aclaraciones de redacción de este documento al supervisor de la prueba (correo supervisor: Jhefferson.rios@ieb.co).

**Explique las respuestas o procedimientos en sus propias palabras, la copia textual de otros autores le restará puntos. Si copia código fuera de su autoría, favor indicar la fuente.**

1. CONTROL DE VERSIONES

Nombre de programa control de versiones empleado en la prueba: git/github

1. ASPECTOS TEÓRICOS.

De acuerdo con el paradigma de programación orientado a objetos, explique:

1. ¿Qué diferencia existe entre clase y objeto?

R/ En el paradigma de la programación orientada a objetos una clase es una abstracción del un objeto, una forma fácil de entenderla es como una plantilla en la cual se definen las propiedades (atributos del objeto) y la forma en la que se comporta el objeto (que en este caso serían las funciones o métodos de la clase). La diferencia entre una clase es y un objeto es que este último es la instancia de la clase que hereda todos los atributos y métodos que esta posee.

1. ¿Qué modificadores existen y cuáles son sus criterios de uso?

R/ Según el concepto de encapsulamiento del lenguaje c# estos son los modificadores de acceso:

* **Internal:** Es comúnmente utilizado dentro de un mismo archivo de solución cuando de crea un proyecto en c# también este se usa cuando se crean las bibliotecas de clases, ya que este modificador se acceso permite acceder a las propiedades y métodos a las clases que pertenezcan al mismo proyecto.
* **Private:** Este modificador de acceso solo se usa cuando se trabaja dentro de la misma clase, su uso en los atributos es una buena practica porque restringe el acceso directo a las propiedades desde otras clases, por lo general para acceder a los valores de estos atributos se emplean los métodos: {get; set;}.
* **Public:** En el caso de este modificador de acceso no se restringe la interacción de loa tributos y métodos cuando se crean instancias de clase ya sean en clases que estén dentro del mismo proyecto o fuera de este.
* **Protected:** Por lo general este modificador de acceso se emplea para la implementación de interfaces o clases abstractas (que son todas aquellas que obligan a implementar todos los métodos y atributos dentro de las clases que hereden de esta), en otras palabras, solo se permite el acceso a clases específicas que heredan de otras.

1. ¿Qué diferencia existe entre polimorfismo y sobrecarga de clase?

R/ En el caso del polimorfismo se pueden tener varias instancias también herencias de una clase que pueden tener otras propiedades y responder de diferentes formas, lo que hace que la implementación del código sea más flexible ya que todas las instancias pueden responder de forma distinta al momento de ser ejecutadas, a diferencia de la sobre carga que se trabaja dentro de la misma clase en la cual se tienen métodos con igual nombre, los cuales pueden recibir parámetros distintos.

1. ASPECTOS PRÁCTICOS.

Diseñe una aplicación web con su respectiva interfaz de usuario y API’s para el desarrollo del siguiente proceso (corriendo en local, no es necesario que lo suba a un servidor):

Plantilla interfaz web (solo una sugerencia, es libre de plantear su propia interfaz).

|  |  |
| --- | --- |
| Titulo WebApp |  |
| Botón correr selección | Zona de ingreso y presentación de variables |
| Botón guardar selección |

**Proceso**

La aplicación facilita la selección de un cable, para lo cual disponen de variables y tablas para su selección, bajo el siguiente diseño de clases (ver Ilustración 1).

Las tablas y la información que se pide guardar, deben almacenarse en una base de datos, para lo cual se pide pegar en el anexo las tomas de pantalla durante la configuración de las mismas. **De preferencia emplear PostgreSQL o SQLite como gestor y respetar el contenido de las tablas que se presentan, no obstante, se destaca que es libre de emplear el gestor que usted prefiera.**

**Se deberá presentar en lenguaje de programación C# .Net utilizando algún patrón de arquitectura de aplicaciones web.**

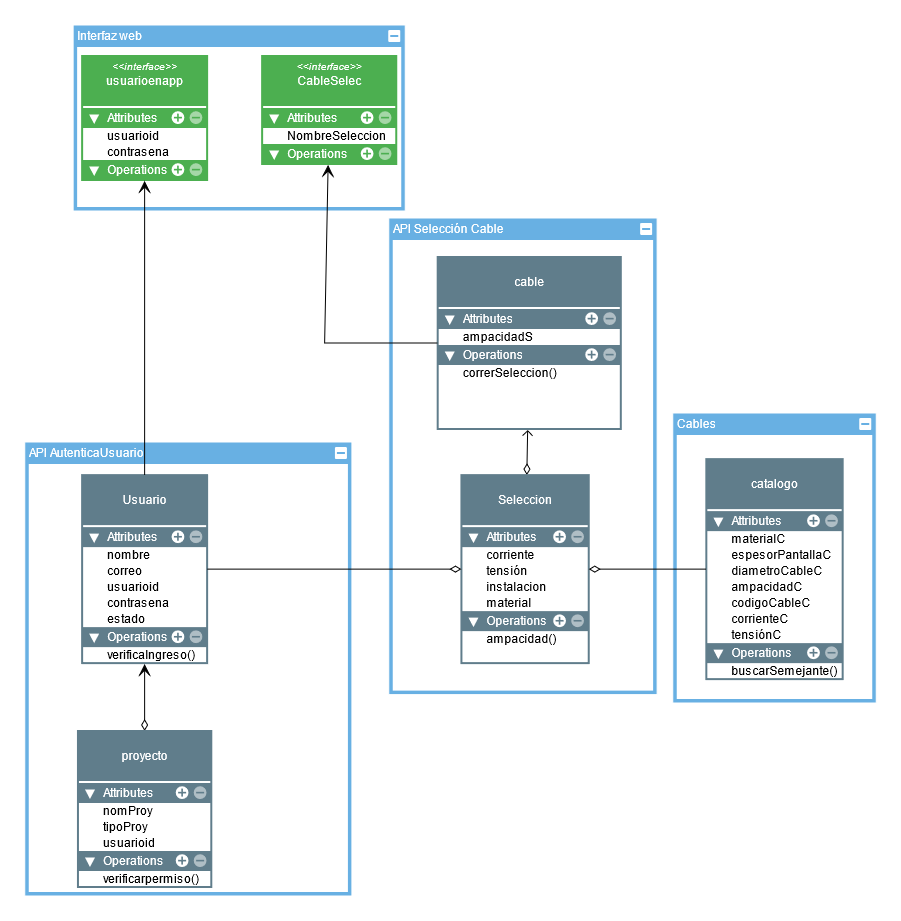


Ilustración 1. Diseño de aplicación.

El anterior diagrama UML, presenta explicito el alcance de los métodos que manipularan cada API y en general la estructura de las clases, favor respetar este diseño. En cuanto a las relaciones entre tablas es libre de plantear sus correspondencias.

El flujo de datos es el siguiente

1. Al ingresar a la página web se solicita usuario y contraseña, el cual será verificado por la API AutenticaUsuario, el cual verifica:

Método: verificaingreso()

Determina si el usuario ingresado, se encuentra entre los usuarios permitidos (ver tabla 1)

Tabla 1. Usuarios permitidos

|  |  |
| --- | --- |
| **Usuario** | **Contraseña** |
| juan.cobo | Jun@ |
| Esmeralda.gutierrez | Es\*45 |
| Jake.grajales | Jak180 |

En caso de encontrarse el usuario como permitido, se carga la siguiente tabla, tomando la fila del usuario ingresado, **cargando la información en algún lugar de la página web**.

Tabla 2. Información de usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Usuario** | **Nombre** | **Correo** | **Estado** |
| juan.cobo | Juan Cobos | juncobos@gmail.com | 1 |
| Esmeralda.gutierrez | Esmeralda Gutierrez | Esmeraldita45@gmail.com | 1 |
| Jake.grajales | Jake Grajales | jgrajales@gmail.com | 1 |

Si el usuario es válido, debe permitir ingresar la siguiente información

* Corriente (numérico)
* Tensión (numérico)
* Instalación (string)
* Material (string)
* Nombre Proyecto (string)

Una vez llene los valores y oprima el **Botón correr selección**, se ejecutan los siguientes métodos

Método: verificarpermisos()

Este método verifica si el usuario tiene permiso de correr cálculos, según la siguiente tabla:

Tabla 3. Usuarios autorizados por proyecto

|  |  |  |
| --- | --- | --- |
| **Usuario** | **nomProy** | **tipoProy** |
| juan.cobo | Cuestesitas1 | Conceptual |
| Jake.grajales | Cuestesitas2 | Ingeniería detalle |

En caso negativo, indicar que el usuario no ha sido ingresado al proyecto, por tanto, no tiene permitido continuar los cálculos.

En caso afirmativo, se corre el método CorrerSelección(), quien corre los siguientes métodos de manera organizada.

1. Método: ampacidad()
2. Método: buscarSemejante()

Ampacidad(), entrega:

1.26 si la corriente ingresada es menor a 40

2.5 si la corriente ingresada es mayor o igual a 40

buscarSemejante(), entrega información según la siguiente tabla:

Tabla 4. Selección de cable y parámetros corregidos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ampacidad** | **Material** | **Tensión Corregida** | **Espesor pantalla Corregido** | **Diámetro cable Corregido** | **CodCable** | **Corriente corregida** |
| 1.26 | Cobre | 13 | 1 | 3 | 1 | 25 |
| 2.5 | Cobre | 110 | 2 | 5 | 2 | 50 |
| 1.26 | Aluminio | 20 | 1 | 3 | 3 | 25 |
| 2.5 | Aluminio | 130 | 2 | 5 | 4 | 50 |

La selección de la fila se da por valor de ampacidad y material, reportando todos los valores de la fila como resultado, valores que serán presentados en la interfaz gráfica como cable seleccionado.

Para finalizar se permitirá ingresar el nombre de esta selección

NombreSeleccion (string)

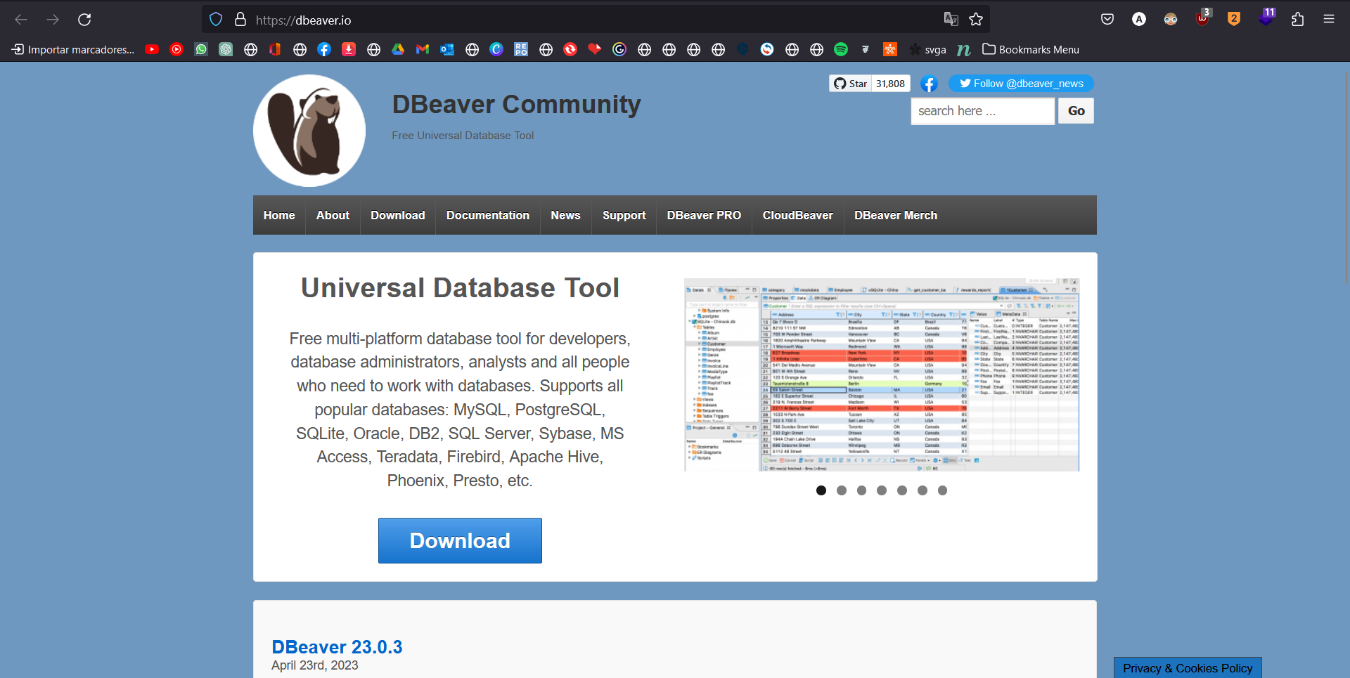
Al oprimir el **Botón guardar selección**, se ejecuta el guardado de la siguiente información en base de datos.

Tabla 5. Registro de selección de cable.

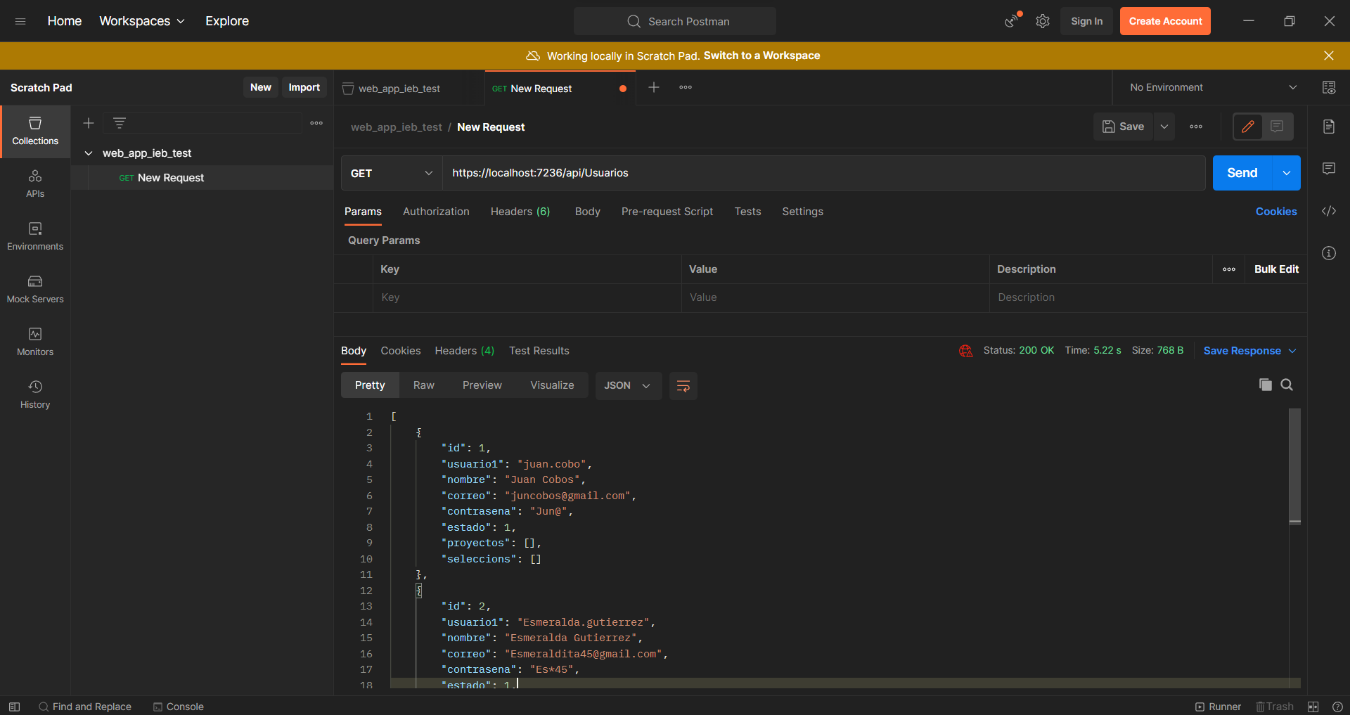
|  |
| --- |
| **NombreSelección**  **Corriente**  **Tensión**  **Ampacidad**  **Corriente corregida**  **Usuario**  **nomProy**  **Fecha inserción (formato dd/MM/aaaa hh:mm) (tomado del equipo donde corre la WebApp)** |

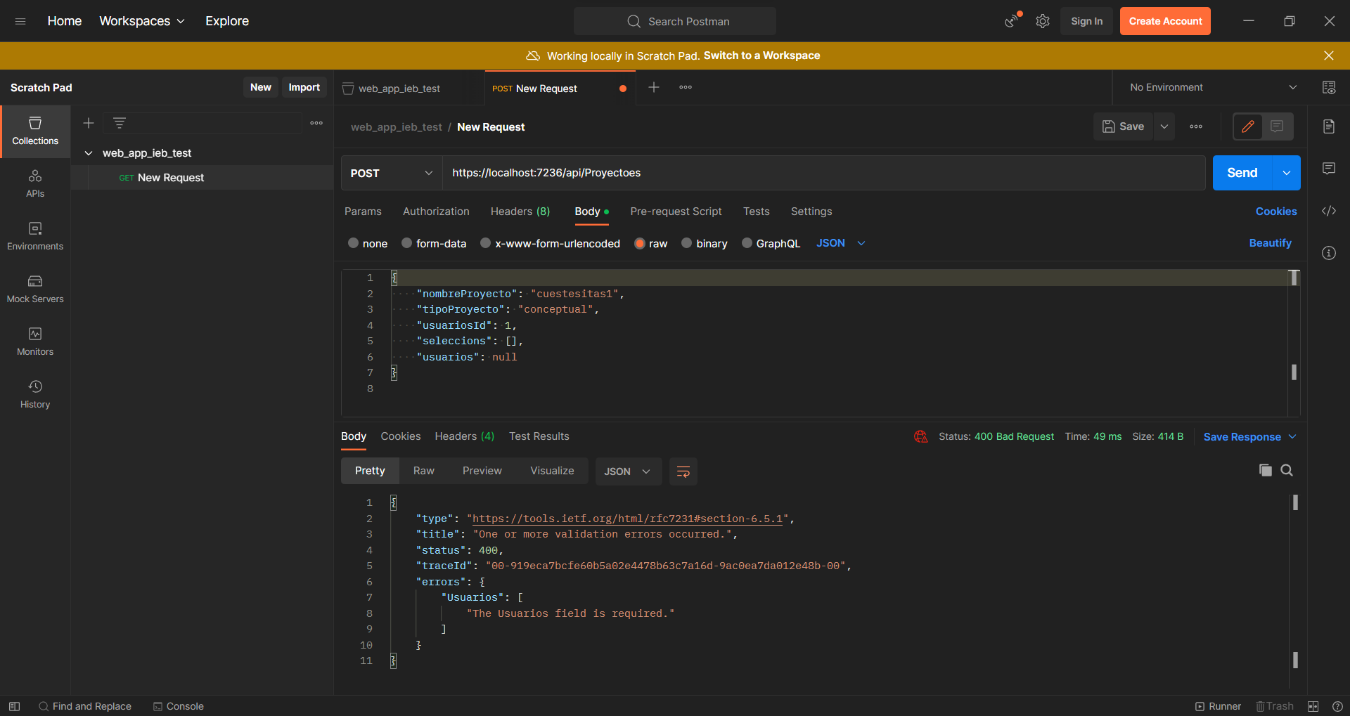
1. ANEXO, CAPTURAS DE PANTALLA CONFIGURACIÓN DE BASE DE DATOS.

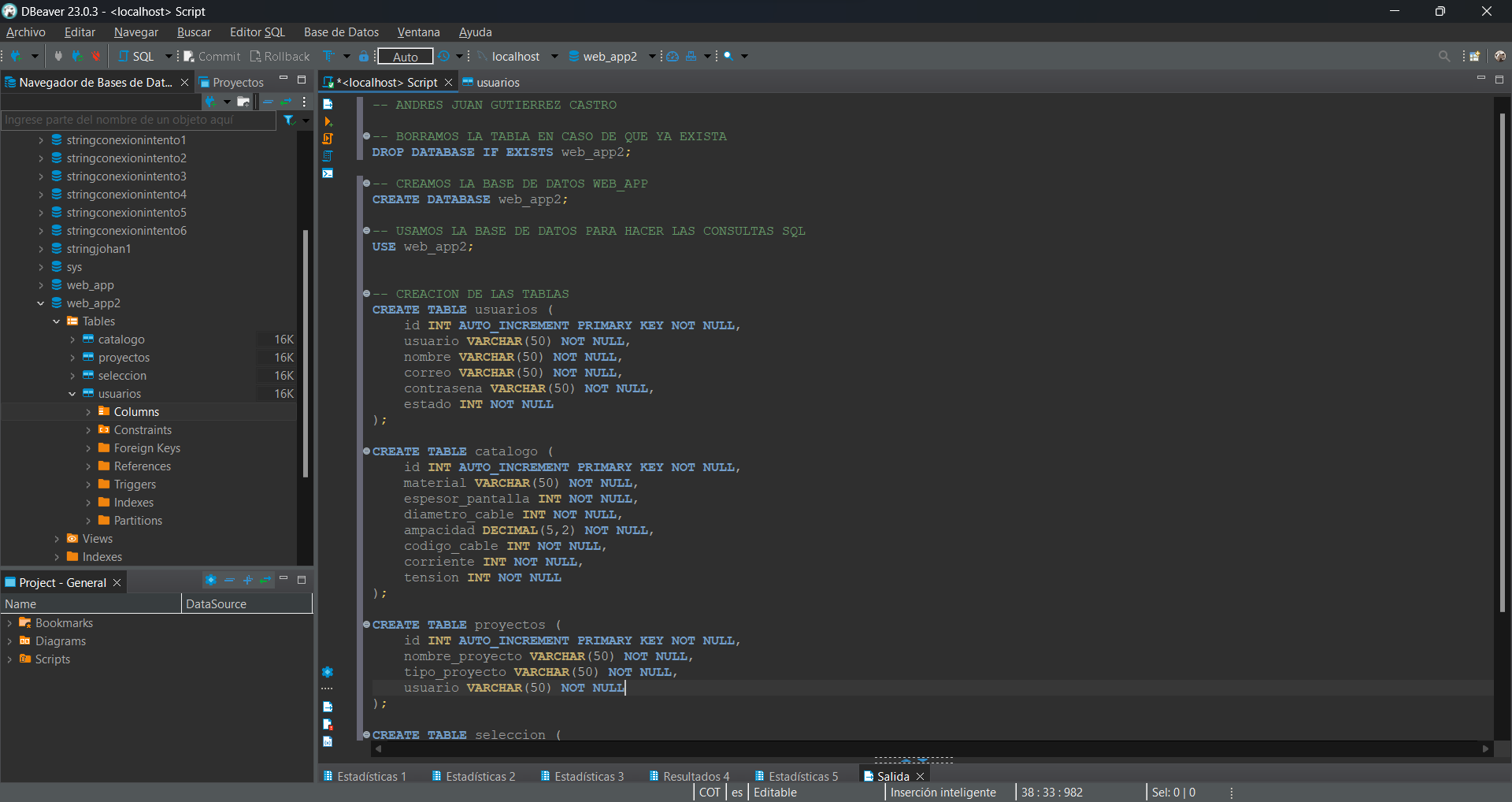
R/ Para la solución de esta parte practica se usó una base de datos MySQL:

Para la creación de la base de datos se pueden usar diferentes gestores de bases de datos, como lo pueden ser heidiSQL, workbench, etc. Para la solución de este problema se va a implementar el gestor de bases de datos DBeaver.

Lo siguiente que se debe hacer es la creación de las tablas, como ahora visualstudio 2022 no tiene la opción para usar el conector de MySQL, se procederá a usar las librerías de: pomelo y entity framework, por lo cual cuando se vayan a crear las tablas no van a tener relación entre sí, esto porque al momento de hacer el scaffold y generar lo modelos que representan las tablas y sus relaciones, hacen que la información sea redundante al momento de agregar registros a las tablas. En este ejemplo se está consumiendo la API del proyecto que tiene tablas relacionadas y se puede apreciar que a pesar de que la tabla usuarios definida en la base de datos no tiene claves foráneas, hace la representación de las tablas que si hacen relación con esta, lo cual es un problema ya que al agregar registros estos campos de las tablas deben llenarse una vez más.

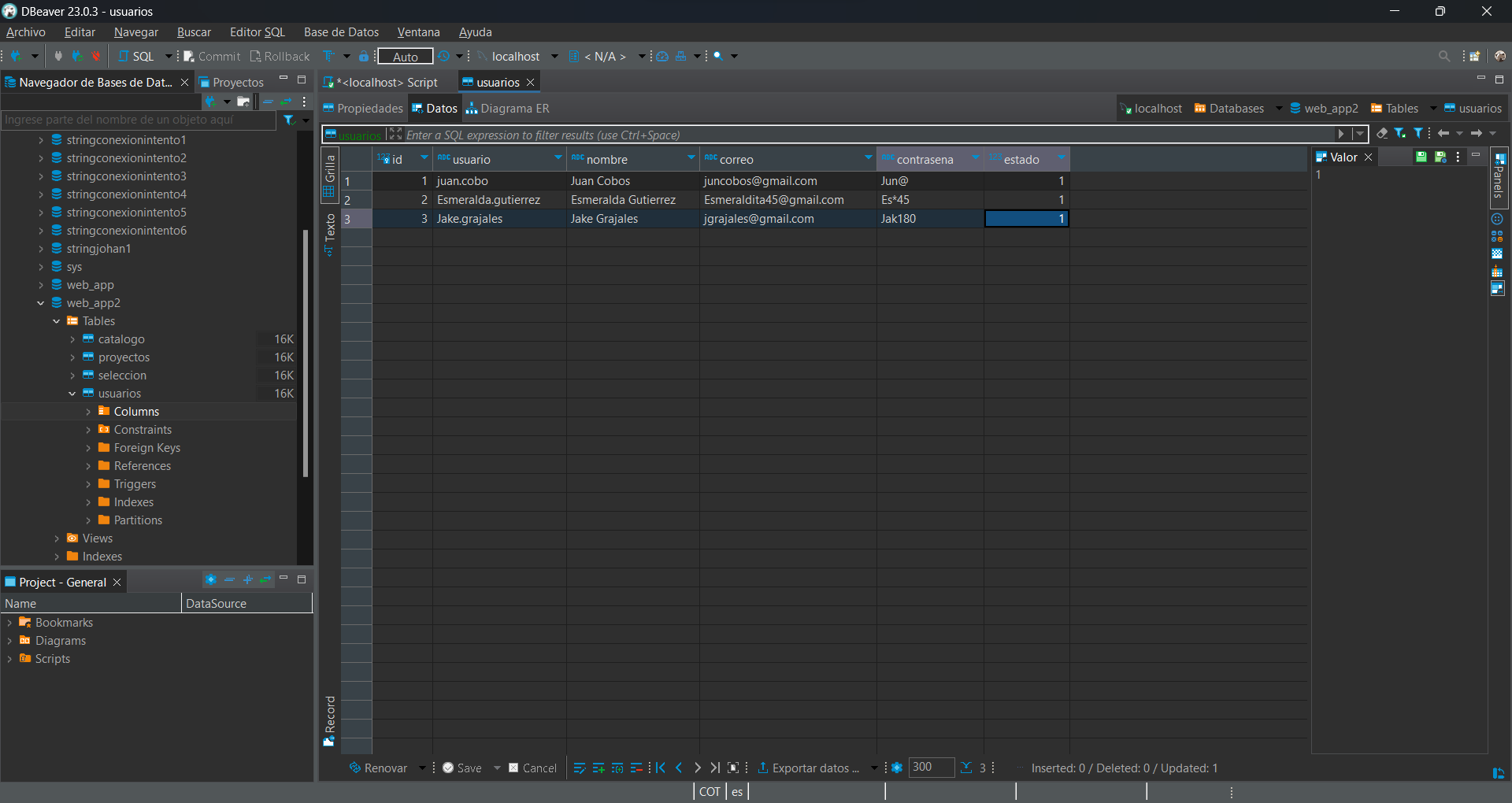


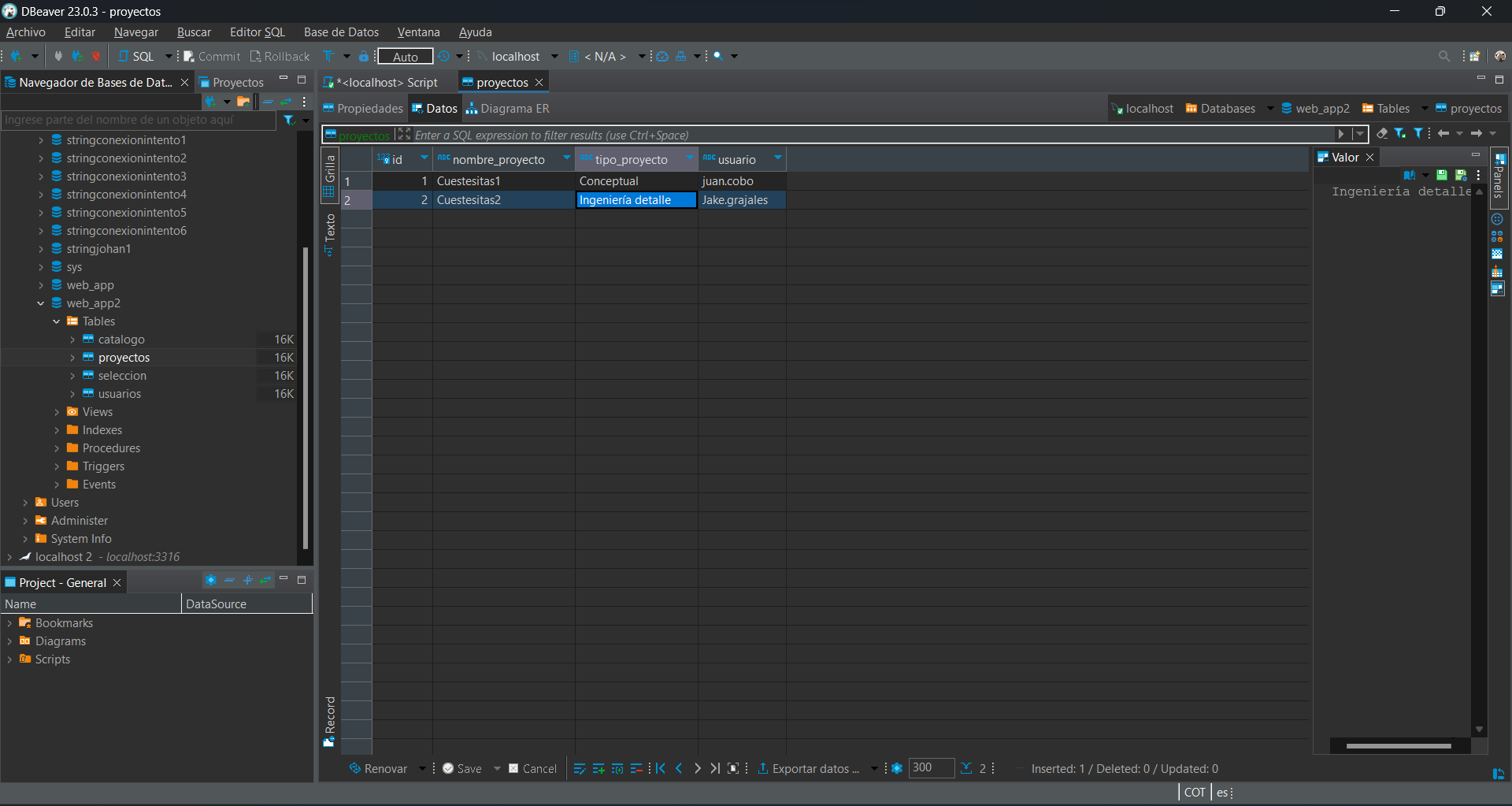


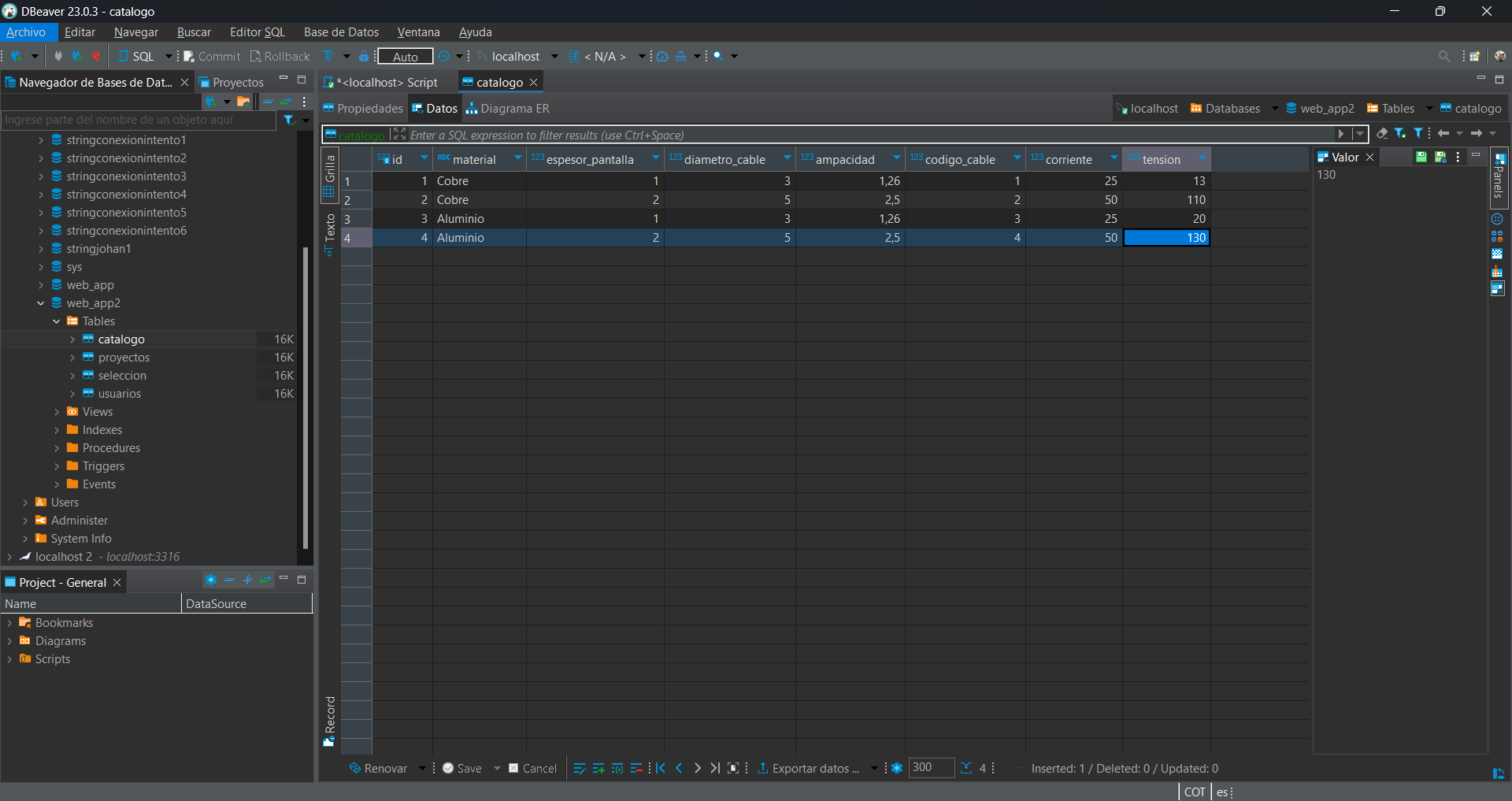
Se crea el script para crear la base de datos y las tablas.

Captura de pantalla de computadora

Descripción generada automáticamente

Lo siguiente de deba hacerse es agregar los registros estáticos que el problema proporciona, en este caso son los de la tabla de usuarios, proyectos y catálogo.

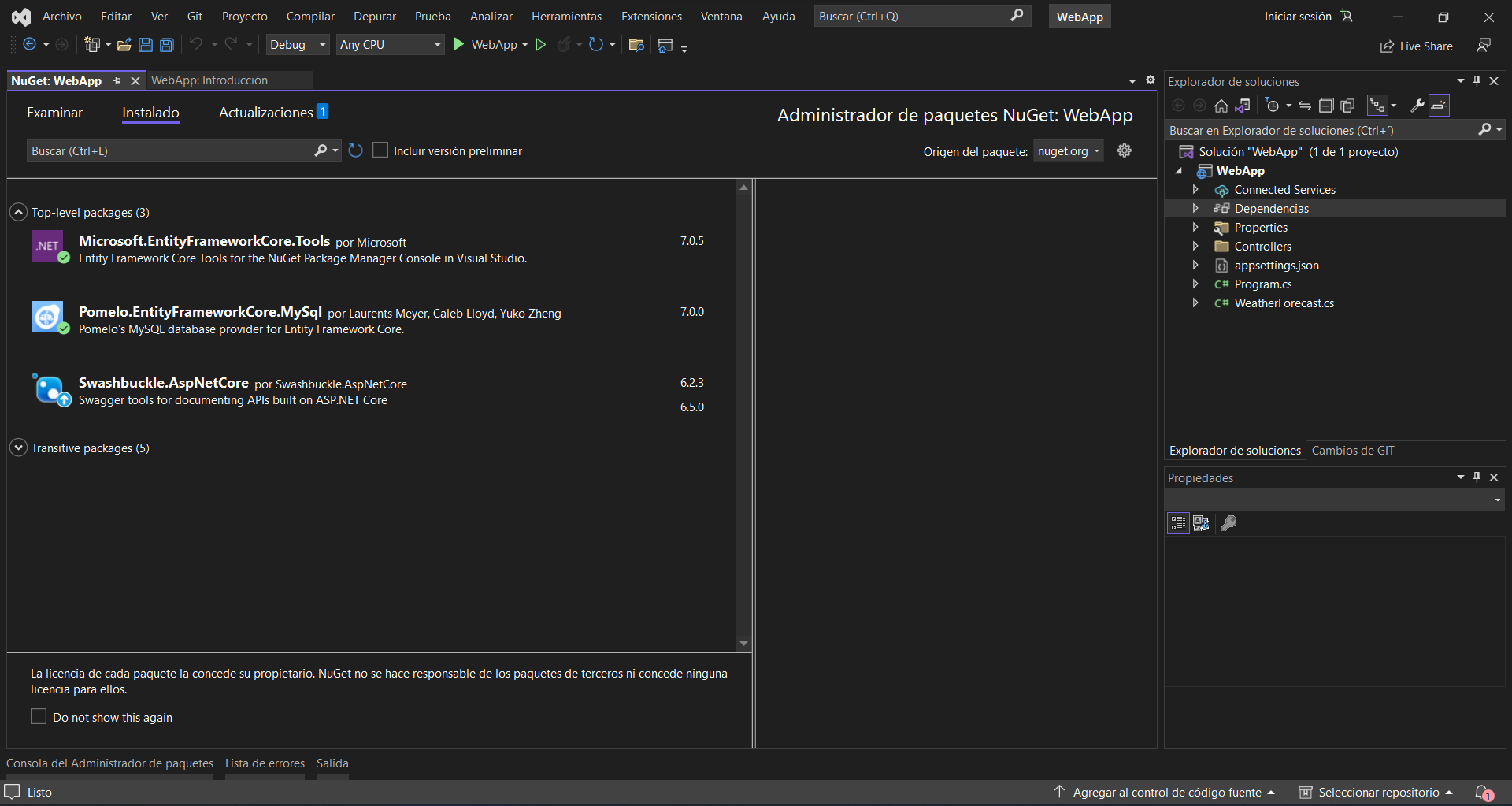


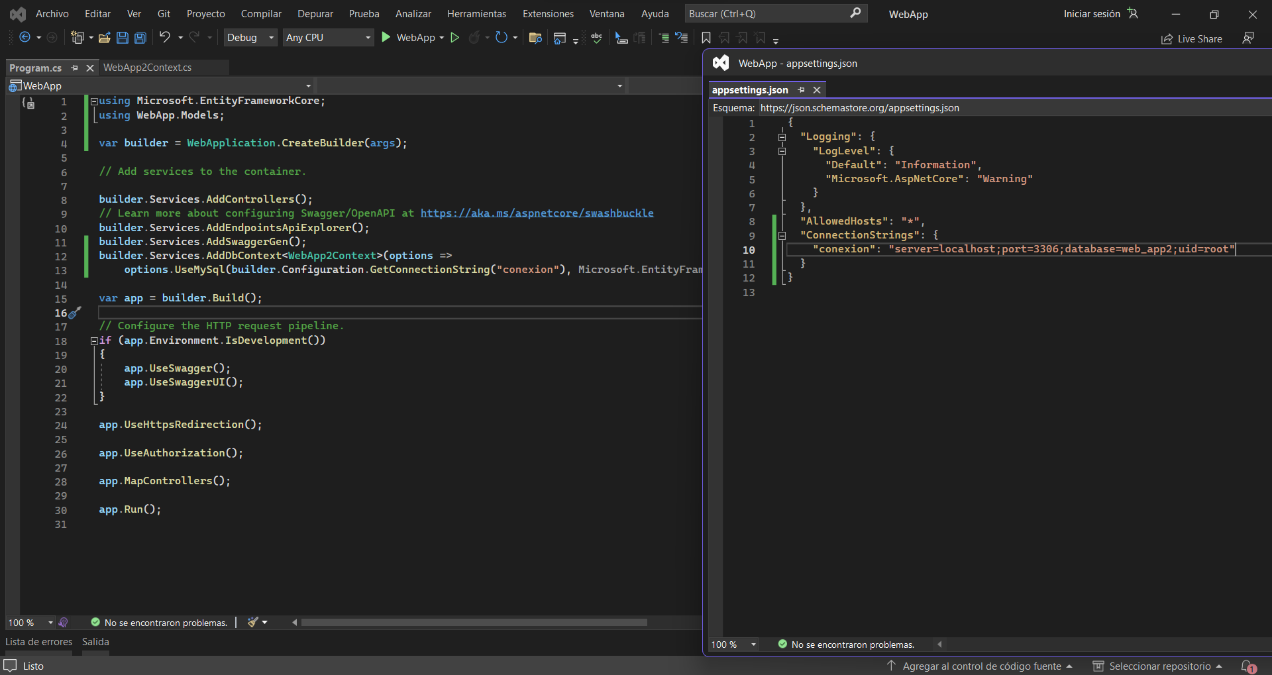


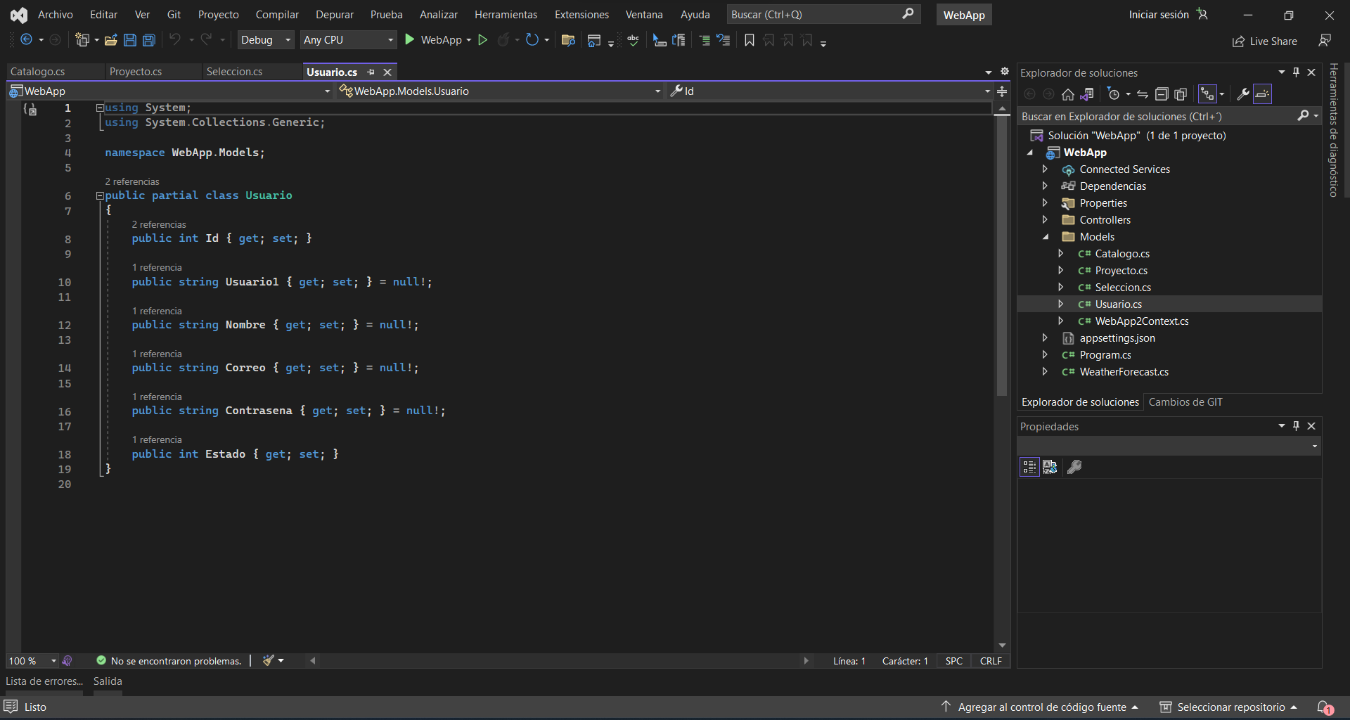
1. OTROS ANEXOS

Espacio adicional para anexar fotos de tus diseños u otro material complementario en la documentación o soporte de esta prueba.

Para la solución de este problema se optó por una arquitectura basada en microservicios, para la cual se creó un API para hacer las peticiones https para interactuar con la información de la base de datos.

Una vez el proyecto se haya creado, el siguiente paso a realizar es la instalación de las dependencias, para este proyecto se va a usar pomelo que establece la conexión con la base de datos y también nos permite hacer el scaffold de las tablas y finalmente vamos a usar entity framework para un mejor manejo de las consultas SQL.

El archivo de conexión debe configurarse en el program y en el appsettings.

Al realizar el scaffold de la base de datos, vamos a obtener la representación de las tablas de la base de datos dentro del código que estamos escribiendo, este código se le denomina modelos y también son usados en el patrón de diseño modelo vista controlador.

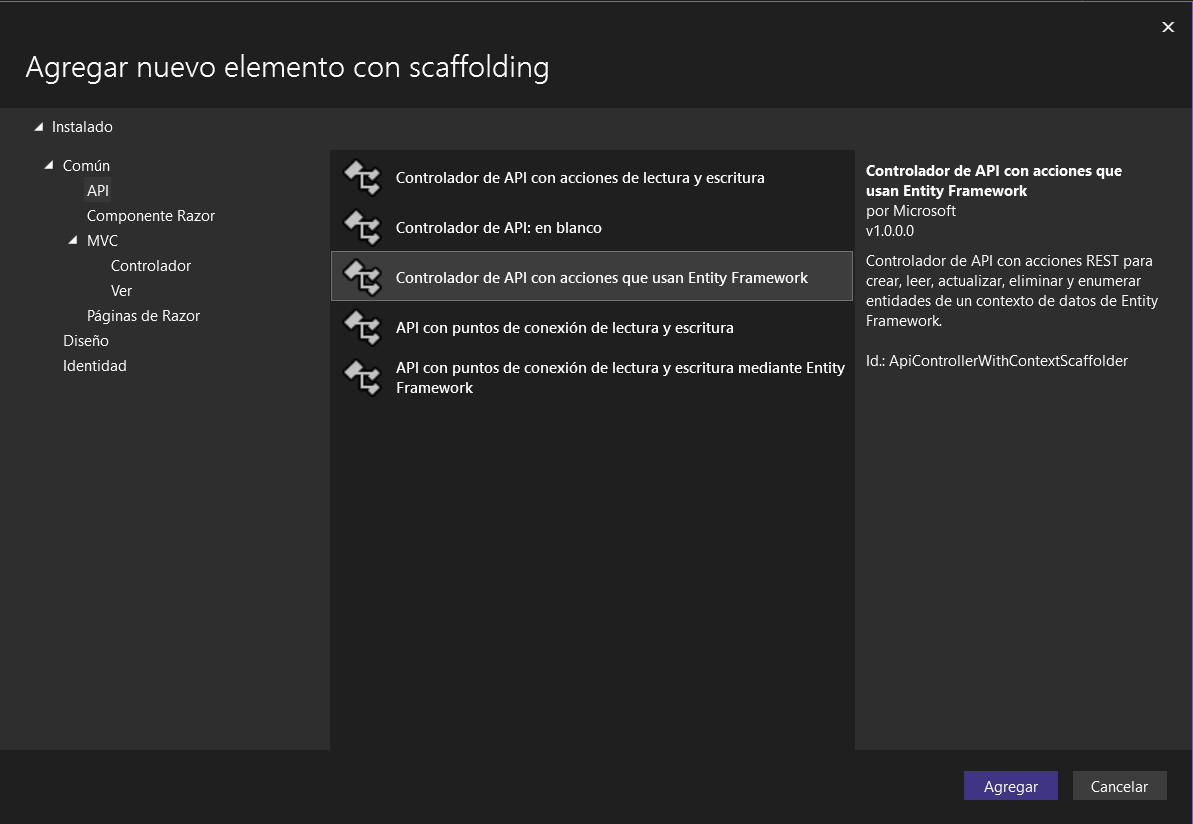
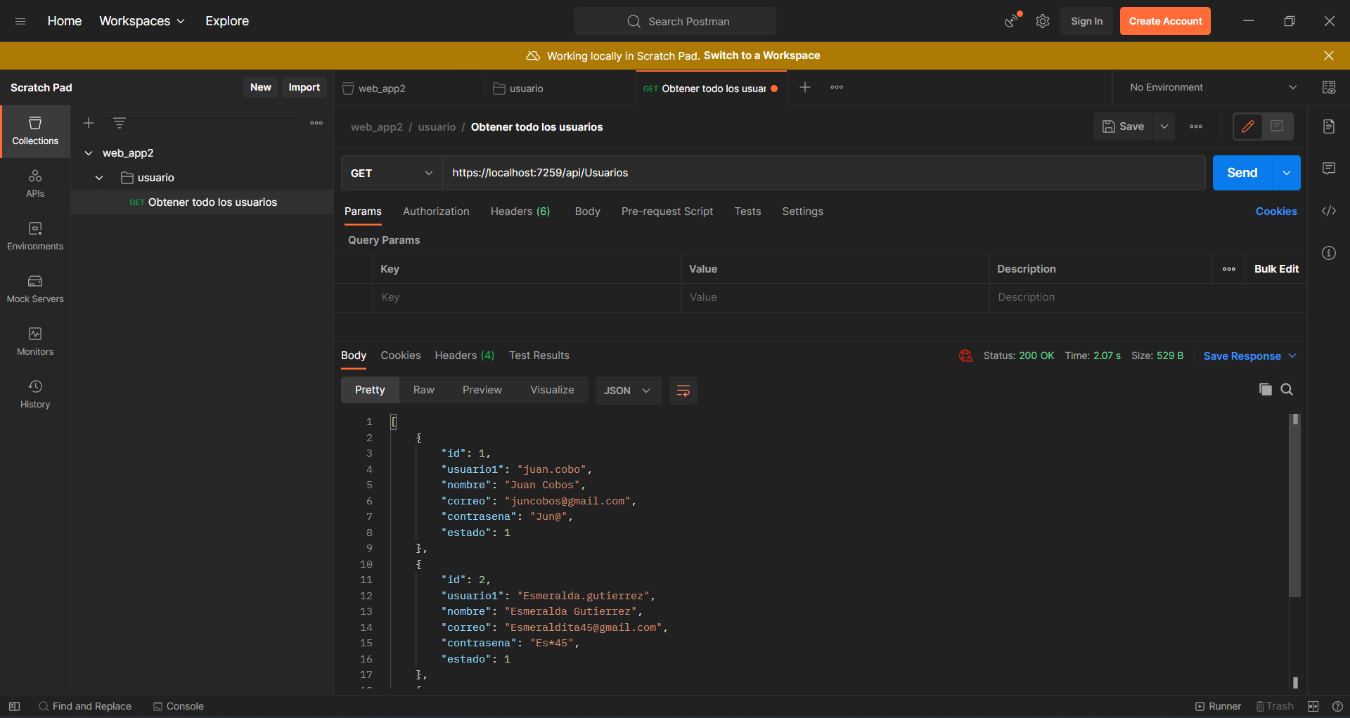
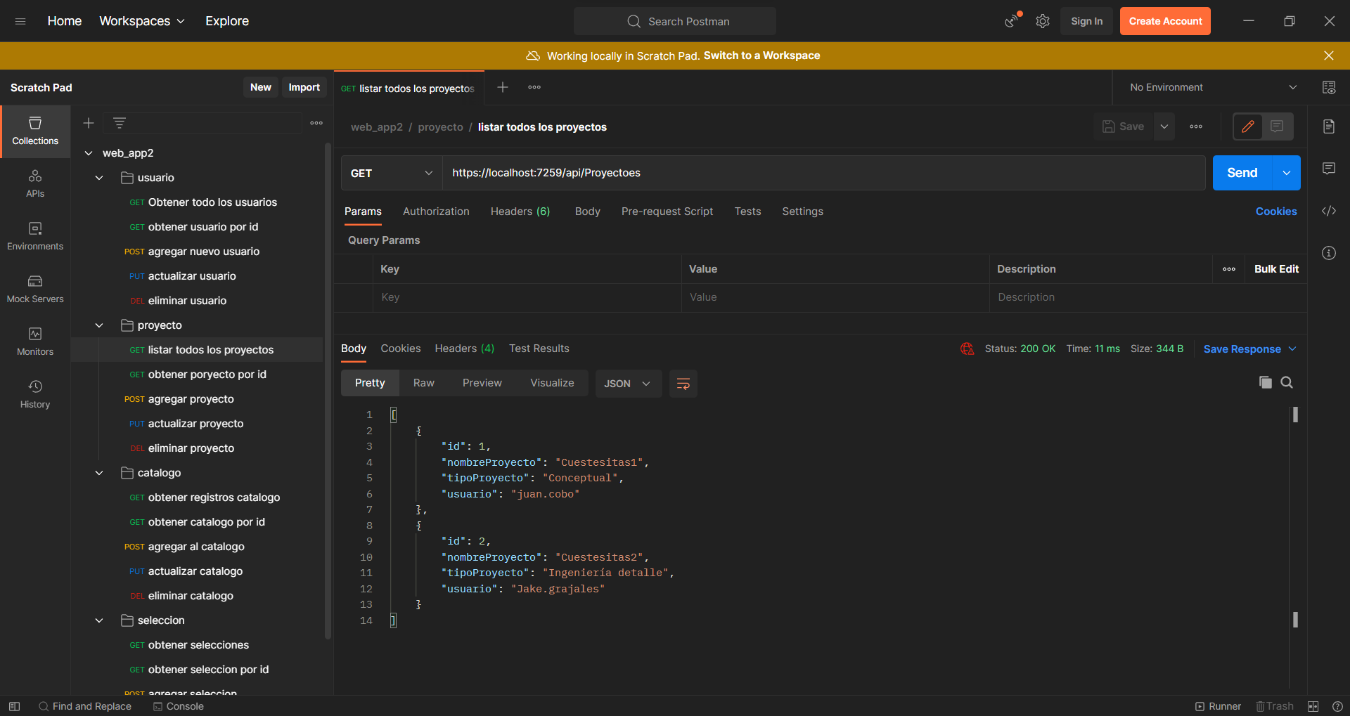
Como ya tenemos los modelos para interactuar con la base de datos, lo siguiente que debe hacerse es establecer la lógica usando los protocolos https de las APIs. Para ello por cada modelo que tengamos debemos crear su respectivo controlador con la lógica.

Imagen que contiene Interfaz de usuario gráfica

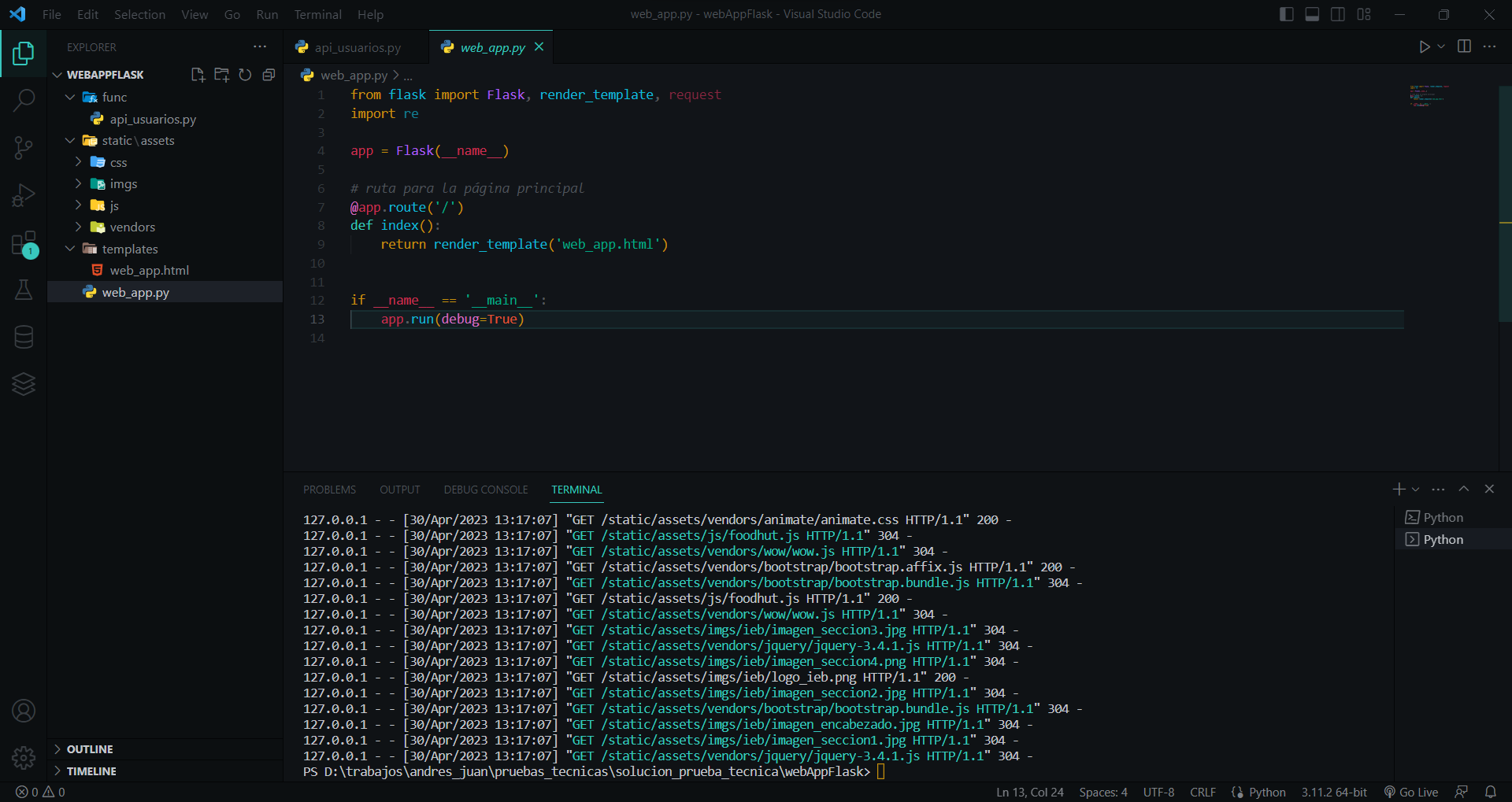
Descripción generada automáticamenteUna vez la lógica de los controladores este escrita, se procede a ejecutar el proyecto para manipular el API, para ello visual studio provee de un gestor web de API, también se pueden usar programas como postman.

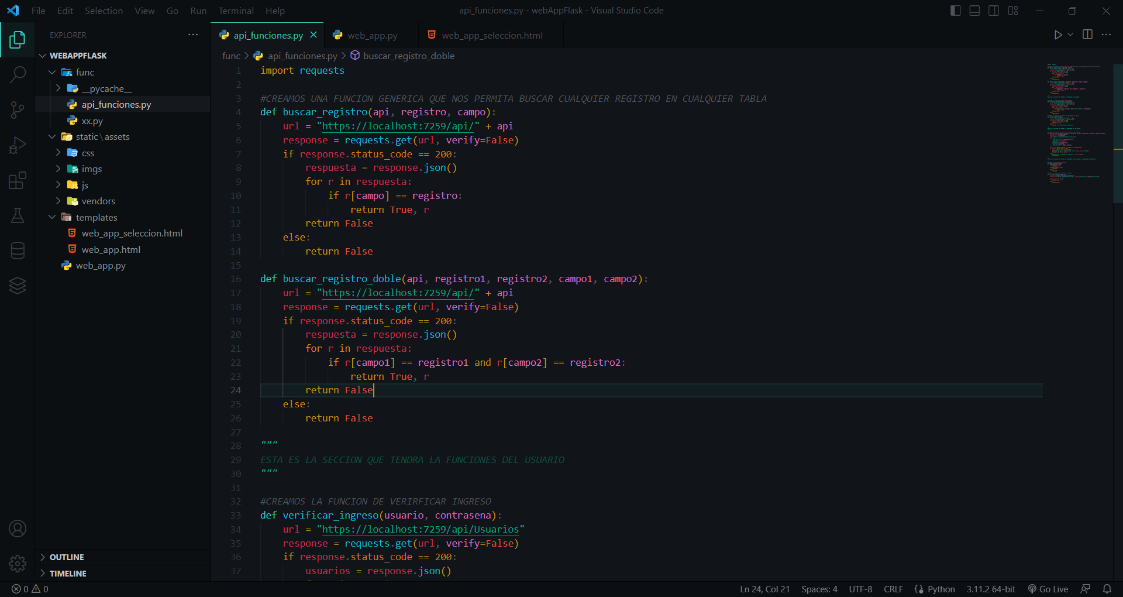


Antes de comenzar con el desarrollo frontend para la interfaz de usuario se prueban que todos los métodos definidos en la API funcionen correctamente.



Para le diseño de la pagina web que va a consumir la API, se implementó un diseño de Bootstrap y css, tomando como inspiración el diseño original de la empresa.

La ventaja de trabajar con APIs es que agiliza y hace flexibe el desarrollo de proyectos, estas pueden ser consumidas por diferentes lenguajes de programación. Como ya la mayoría de la parte lógica está escrita en c#, se va a implementar en la parte del frontend por medio del framework de flask.

Para manipular las funciones que se piden en el problema se va a usar un script, el cual será importado desde el archivo de flask.

Captura de pantalla de computadora

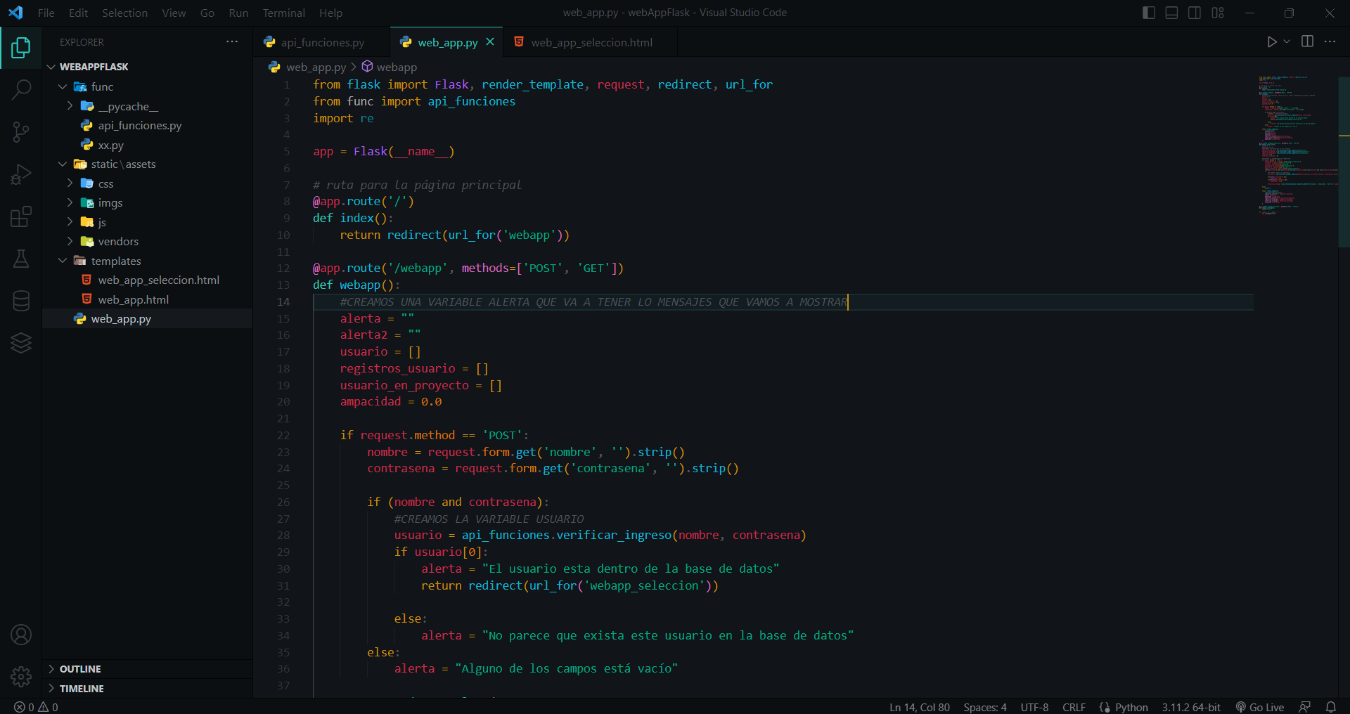
Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Por medio del archivo de flask se van a establecer las cláusulas de validación, se llamarán las funciones y se administrarán las rutas de la aplicación.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamenteEsta seria la vista general del resultado final.



Captura de pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente