



Arquitectura De Software

Realizado por:

Esteban A Villa Berrio

Stefania García

Laboratorio 1 – Arquitectura

Presentado a:

Andrés Armando Sanchez Martin

Octubre del 2022

Marco conceptual

El objetivo de este laboratorio es aprender sobre distintas herramientas y realizar una práctica para su uso. A lo largo de este documento se mostrará el procedimiento que se tuvo que realizar para este laboratorio y también las lecciones aprendidas.

A continuación, se explicarán las herramientas que se utilizaron con sus características para el desarrollo.

1. GitHub: Es un sitio “social coding” que permite subir repositorios de código para almacenarlo en el sistema de versión de controles de Git, se puede colaborar en proyectos de código, y el sistema es código abierto por defecto, lo que significa que cualquier persona en el mundo puede encontrar tu código, usarlo, aprender de él, y mejorarlo.
 - ¿Para que se utilizó? Esta herramienta fue de gran utilidad para crear un nuevo repositorio que permitiera subir el código realizado para la solución de este laboratorio
 - Procedimiento: Primero se ingresó al github, se creó un nuevo repositorio, se invitó a los integrantes del grupo para la colaboración, se puso el nombre personapi-dotnet dado por el profesor, se seleccionó como público y se agregó un archivo readme como base en el repositorio.
2. SQL Server 2019 Express: Es una edición gratis y llena de características de SQL Server, ideal para aprender, desarrollar y fortalecer las aplicaciones de escritorio, de servidores web y pequeños y para la redistribución por parte de fabricantes de software independiente.
 - ¿Para que se utilizó? Esta herramienta fue de gran utilidad para administrar las tablas pertenecientes al modelo dado en el enunciado del laboratorio
 - Procedimiento: Primero se realizó la descarga en la página de Microsoft en la edición gratuita express para el desarrollo, al ejecutar la descarga se selecciona el tipo básico de instalación y ya al terminar de instalar se dio aceptar, se muestra al usuario las especificaciones y el nombre del servidor para la conexión.
3. SQL Server Management Studio 18: Es un entorno de desarrollo integrado para administrar cualquier infraestructura SQL. Se utiliza para acceder, administrar, configurar y desarrollar todos los componentes de SQL Server.
 - ¿Para qué se utilizó? Crear, modificar, agregar objetos a una base de datos, implementar bases de datos, ejecutar una consulta en una base de datos, gestionar y optimizar bases de datos

- Procedimiento: Se realizó la descarga de la página de Microsoft específica para SQL Management, se termina la descarga y se abre la aplicación que muestra un cuadro donde se observa la conexión disponible que existe por defecto
4. Visual Studio Community 2022: Un IDE completo que es extensible y gratuito, sirve para crear aplicaciones modernas además de aplicaciones y servicios en la nube. Contiene un instalador basado en la carga de trabajo que optimiza y garantiza todo lo que se necesita
- ¿Para que se utilizó? Esta herramienta fue la base de todo el laboratorio ya que permitió crear el código de solución y la conexión a la base de datos
 - Procedimiento: Se realizó la descarga de la página de Microsoft, luego se muestra un recuadro donde se especifican los complementos que se van a utilizar, en este caso se seleccionó el desarrollo ASP.NET y web con las características avanzadas, el almacenamiento y procesamiento de datos, los elementos del .Net Framework con sus respectivas plantillas
5. Github Desktop: GitHub Desktop es una aplicación que te habilita para interactuar con GitHub utilizando una GUI en vez de la línea de comandos o de un buscador web. GitHub Desktop fomenta que tú y tu equipo colaboren utilizando las mejoras prácticas con Git y GitHub. Puedes utilizar GitHub Desktop para completar la mayoría de los comandos de Git desde tu computadora de escritorio con confirmaciones visuales para los cambios.
- ¿Para que se utilizó? Esta herramienta permitió subir los cambios que se realizaban en el repositorio generado desde la web por medio de una carpeta con la clonación de este
 - Procedimiento: Se seleccionó el repositorio creado, se clonó en una carpeta en el computador de manera local, y al momento de guardar un archivo en dicha carpeta se utilizaba el desktop para dar commit y push

Procedimiento

Para el desarrollo de este laboratorio se realizaron los siguientes pasos:

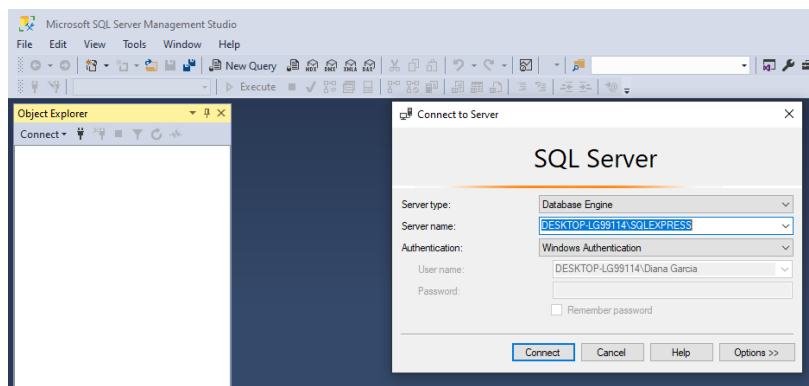
1. Se creó un repositorio en github llamado personapi-dotnet de manera pública que se puede encontrar en la url <https://github.com/DianaStefaniaGarcia/personapi-dotnet>

personapi-dotnet

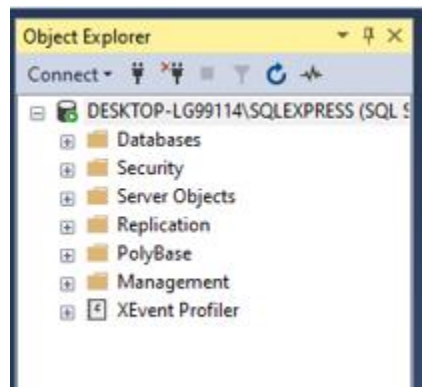
Public

Laboratorio 1

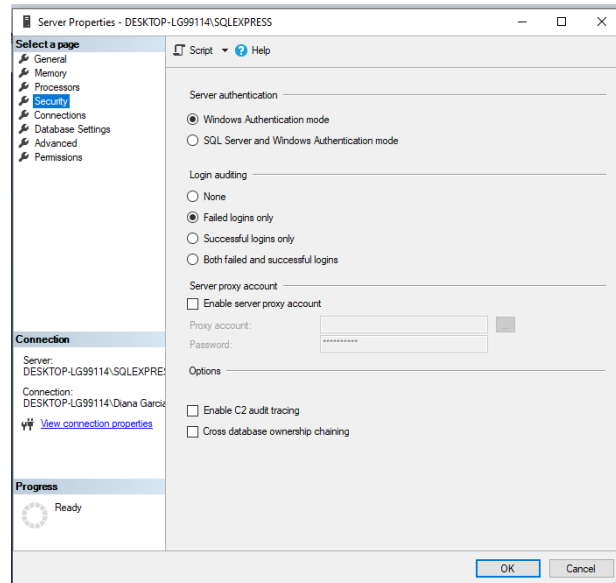
2. Se instaló SQL Server 2019 Express en la versión básica
3. Se instaló SQL Server Management Studio 18
4. Se abrió SQL Server Management Studio 18 mostrando el tipo de autenticación, el nombre del servidor que se generó al realizar la instalación del SQL SERVER Express y que se pone por defecto en el Management Studio



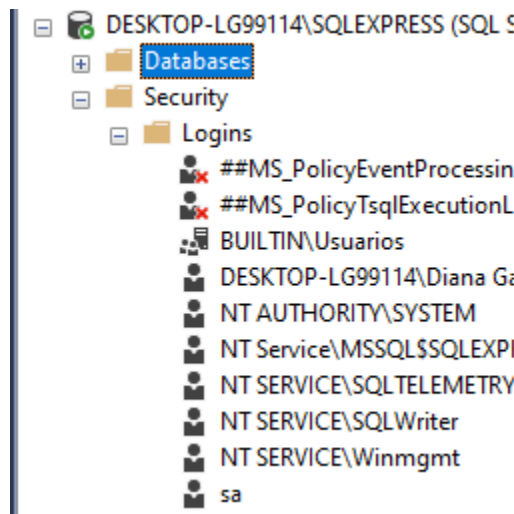
5. Luego, al darle en conectar se debe habilitar la autenticación para un usuario sa dándole clic derecho a la pestaña de la conexión generada por defecto en este caso sería DESKTOP-LG99114



6. Al darle clic aparecerá un recuadro donde se seleccionará la opción security y se habilitará la opción que dice SQL Server and Windows Authentication mode y se le da en ok



7. Nos vamos a la pestaña inicial nuevamente, se despliega la pestaña security y la opción logins buscando el usuario sa



8. En el usuario sa se da clic derecho seleccionando properties para establecer una contraseña a ese usuario, se confirma la contraseña y se da clic en ok

Login Properties - sa

Select a page

- General
- Server Roles
- User Mapping
- Status

Connection

Server: DESKTOP-LG99114\SQLEXPRES

Connection: sa

[View connection properties](#)

Progress

Ready

Script **Help**

Login name:

☐ Windows authentication

☒ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☐ Enforce password expiration

☐ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Mapped Credentials

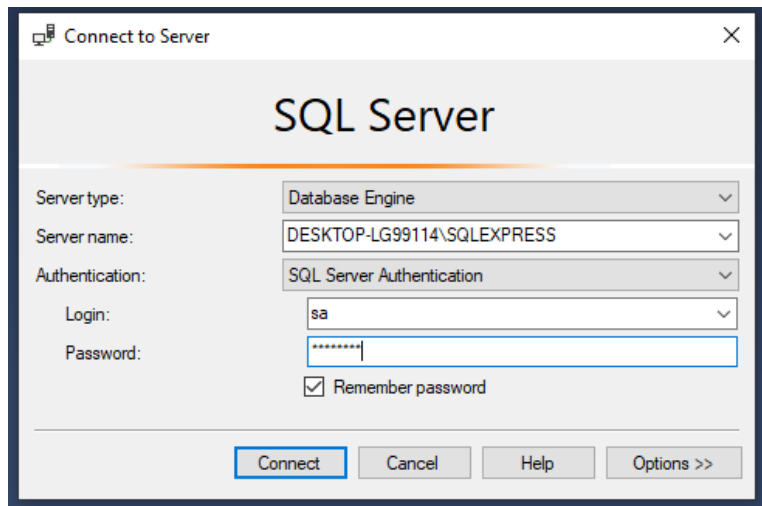
Credential	Provider
------------	----------

Default database:

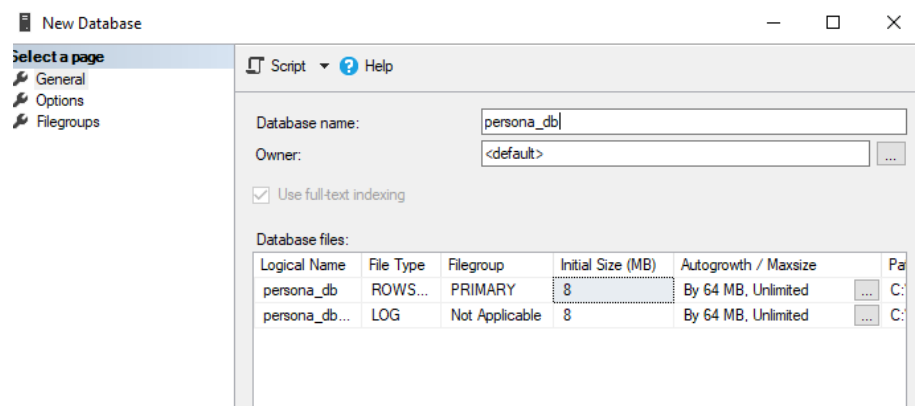
Default language:

9. Luego, se reinicia el computador para que guarde los cambios de configuración

10. Se vuelve a abrir el Management Studio donde volverá a aparecer la pestaña de conexión, pero en este caso al tener ya la autenticación habilitada se escribirá el usuario sa y se escribirá la contraseña que se estableció



11. Se conecta exitosamente y está listo para la creación de la base de datos
12. Se vuelve al menú principal y se da clic derecho en la pestaña Databases, new database y aparece un recuadro donde se pone el nombre de la base de datos, que en este caso será persona_db, y se da en ok



13. Luego se creará la base de datos y procederemos a crear las tablas. Para esto se despliega la pestaña de la base de datos persona_db, clic derecho a la pestaña tables y new table

14. Al dar clic en new table aparecerán los campos disponibles que se deben llenar para crear la tabla, el nombre de la columna, el tipo de dato y si habilita o no la opción para aceptar campos nulos

	Column Name	Data Type	Allow Nulls
▶			<input type="checkbox"/>

15. En este caso se crearon cuatro tablas dadas en el modelo llamadas: Persona, estudio, teléfono y profesión cada una con los campos necesarios y la selección de las llaves primarias respectivas

DESKTOP-LG99114\S..._db - dbo.estudio			
	Column Name	Data Type	Allow Nulls
	id_prof	int	<input type="checkbox"/>
	cc_per	int	<input type="checkbox"/>
	fecha	date	<input type="checkbox"/>
	univer	varchar(50)	<input type="checkbox"/>

DESKTOP-LG99114\S..._db - dbo.persona			
	Column Name	Data Type	Allow Nulls
🔑	cc	int	<input type="checkbox"/>
	nombre	varchar(45)	<input type="checkbox"/>
	apellido	varchar(45)	<input type="checkbox"/>
	genero	varchar(50)	<input type="checkbox"/>
	edad	int	<input type="checkbox"/>

DESKTOP-LG99114\S...b - dbo.profesion			
	Column Name	Data Type	Allow Nulls
🔑	id	int	<input type="checkbox"/>
	nom	varchar(90)	<input type="checkbox"/>
	des	text	<input type="checkbox"/>

DESKTOP-LG99114\S..._db - dbo.telefono			
	Column Name	Data Type	Allow Nulls
🔑	num	varchar(15)	<input type="checkbox"/>
	oper	varchar(45)	<input type="checkbox"/>
	duenio	int	<input type="checkbox"/>

16. Cada una de estas tablas se guardo con su respectivo nombre y se agregaron las respectivas llaves primaria como se evidencia anteriormente
17. Luego, se creó un diagrama para realizar las respectivas conexiones en las tablas, con una llave foránea en la tabla de estudio por parte de la tabla persona y una en la tabla teléfono por parte de la tabla persona



18. Para realizar la llave foránea, sólo se debía tomar la llave primaria de la tabla y arrastrar la conexión a la tabla que se requería, por ejemplo, en este caso desde la tabla persona con la llave primaria en cc se arrastraba a la tabla estudio apareciendo un recuadro así

Tables and Columns

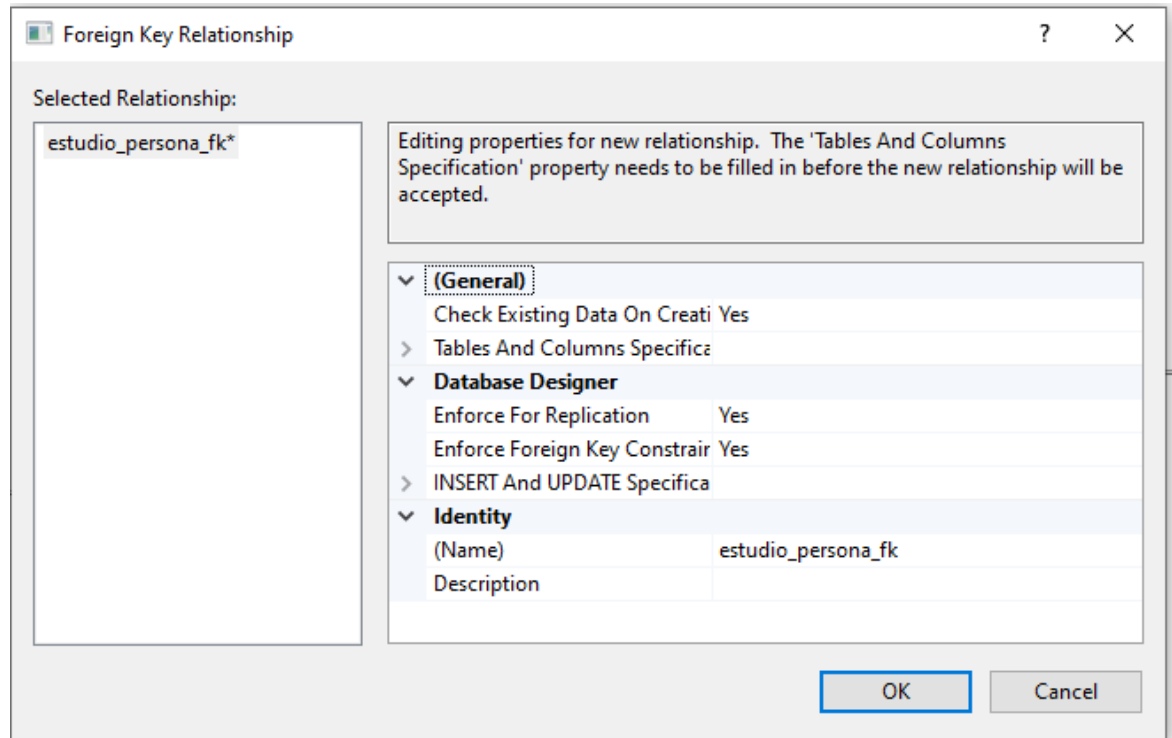
Relationship name:
estudio_persona_fk

Primary key table: persona Foreign key table: estudio

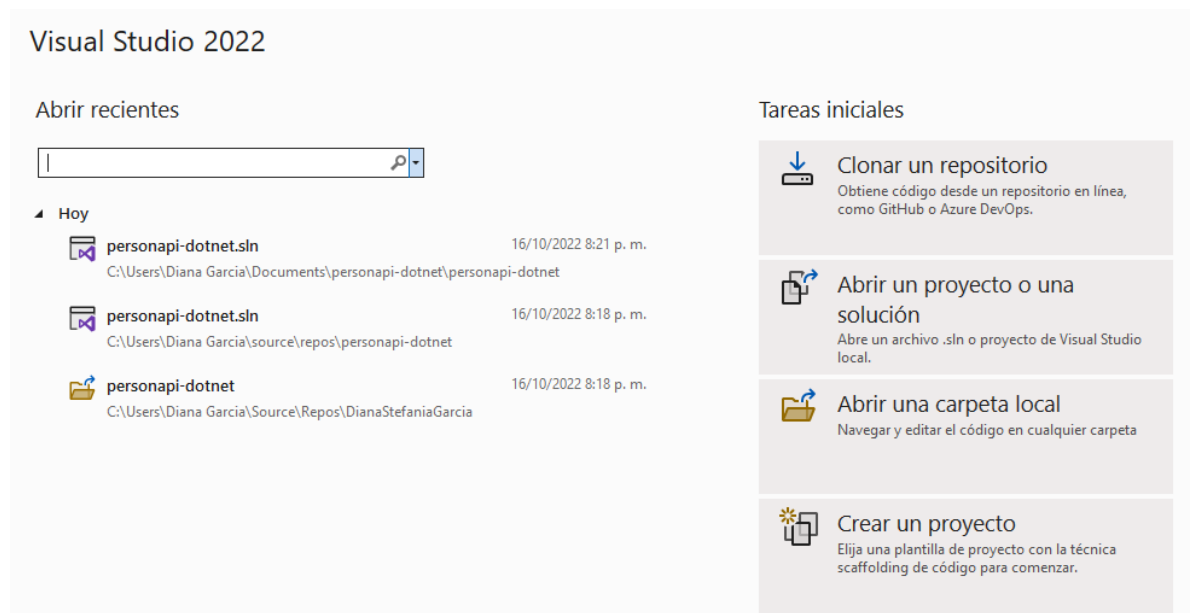
Primary key table	Foreign key table
cc	id_prof

OK Cancel

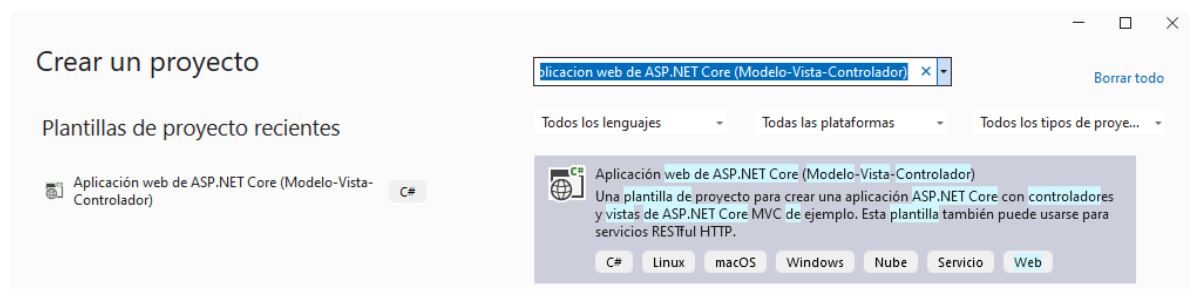
19. Se da clic en ok, luego aparece otro recuadro que especifica la llave foránea creada



20. Luego de tener todo listo en la base de datos con sus respectivas relaciones, se abre Visual Studio se da clic en crear un proyecto



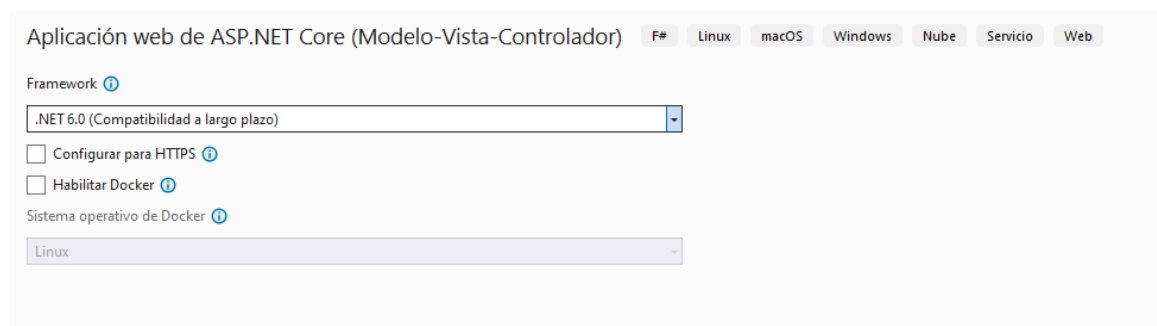
21. Aparece un cuadro con varias opciones disponibles donde en este caso escogeremos la opción de la plantilla Aplicación web de ASP.NET Core (Modelo-Vista-Controlador), se da clic en siguiente



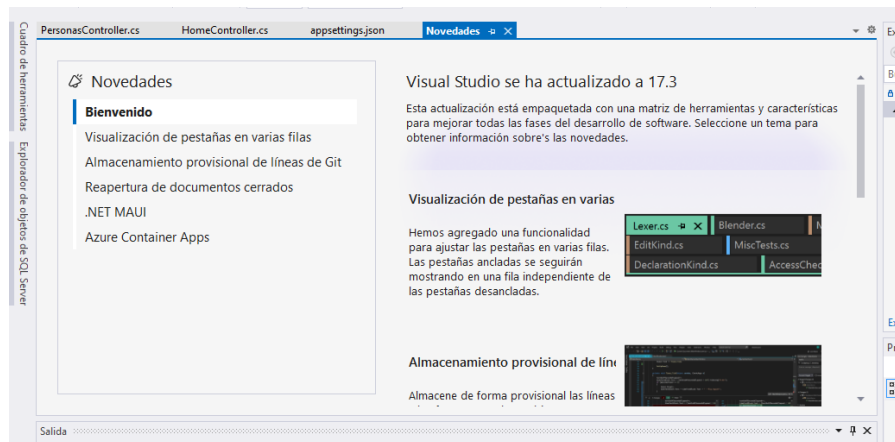
22. Se configura el proyecto con el mismo nombre del repositorio creado anteriormente personapi-dotnet, se da clic en siguiente



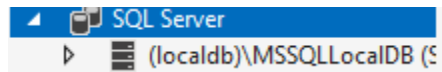
23. Se configura el framework que será .Net 6.0 y se deshabilita la opción de configuración de http, se da clic en siguiente



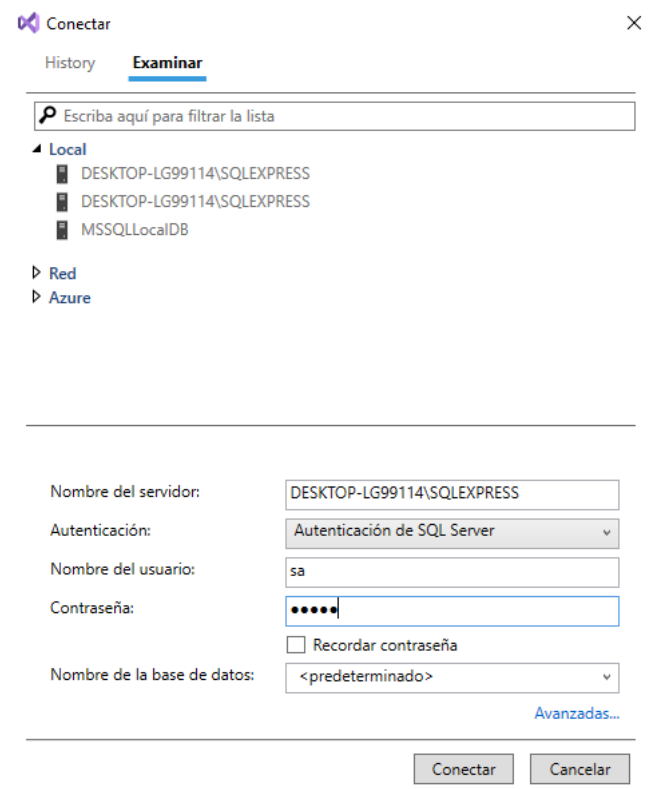
24. Luego se abrirá la pantalla principal del Visual Studio



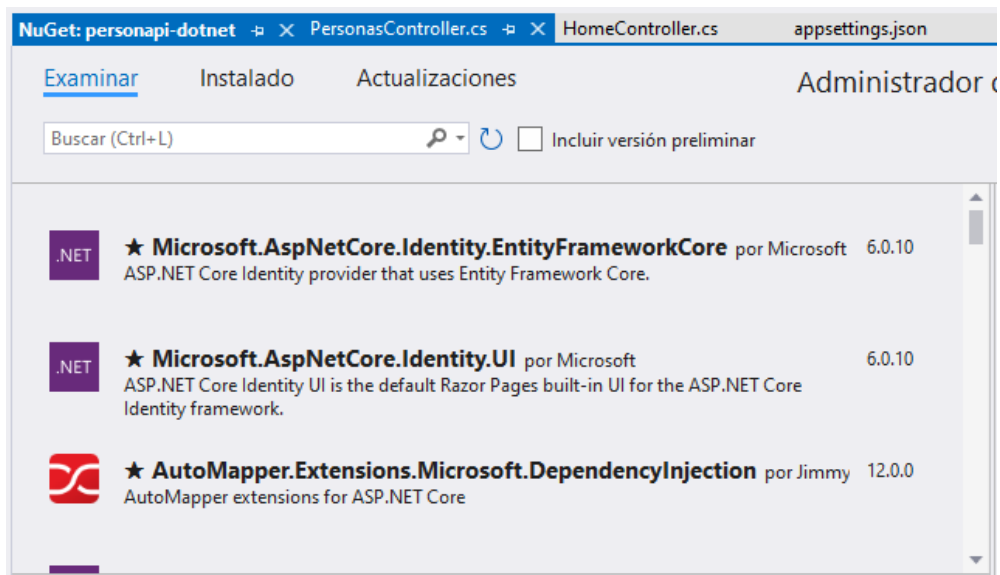
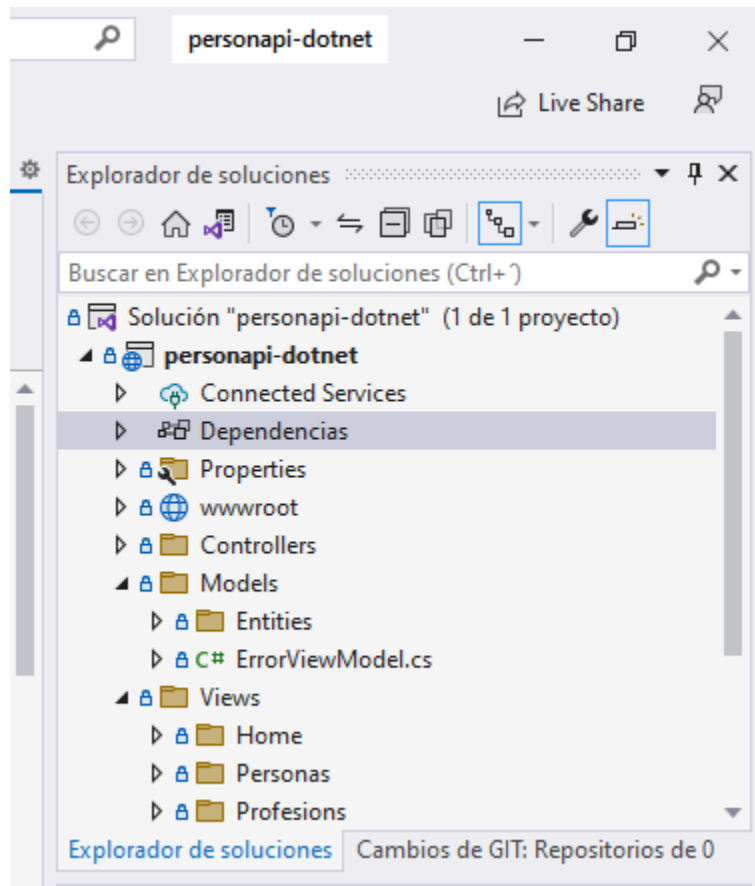
25. En la parte superior se encontrarán unas opciones, en este caso daremos clic a la pestaña Ver y se dará clic en el Explorador de objetos de SQL Server esto automáticamente hará la conexión con la base de datos por defecto que se creó anteriormente



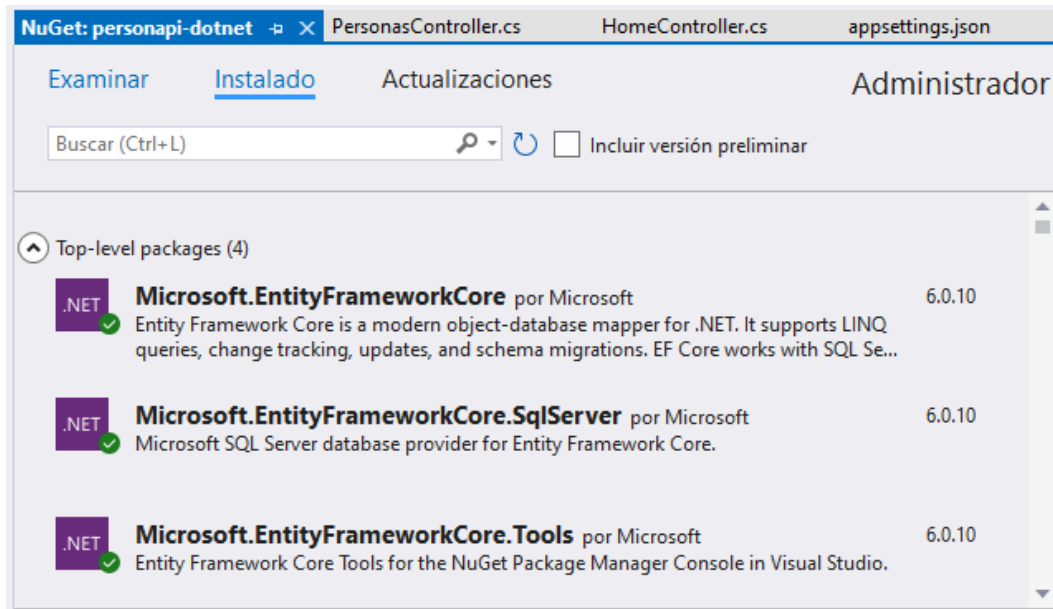
26. Se da clic derecho en SQL Server y se agrega un SQL Server, esto para conectarse a la base de datos persona_db seleccionando donde se encuentra localmente, la autenticación para acceder con el usuario sa y la respectiva contraseña



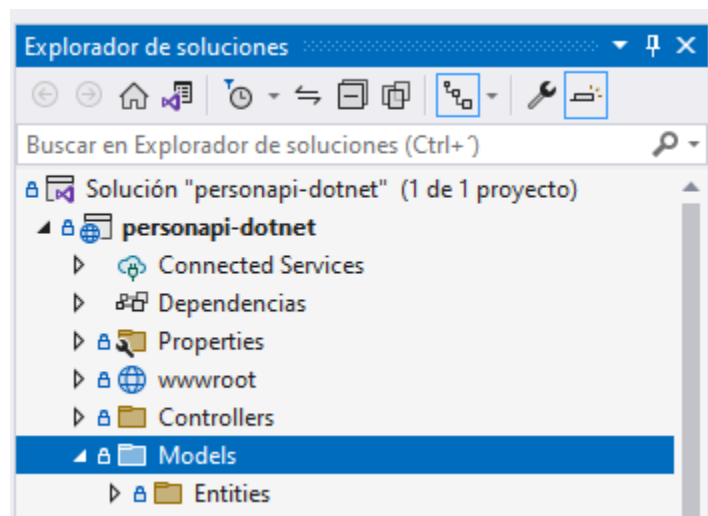
27. En el explorador de soluciones que se encuentra en la parte derecha del Visual Studio, se debe hacer clic derecho en dependencias, ir a Administrar paquetes NuGet, en la pestaña examinar e instalar:
- Microsoft.EntityFrameworkCore
 - Microsoft.EntityFrameworkCore.SqlServer
 - Microsoft.EntityFrameworkCore.Tools



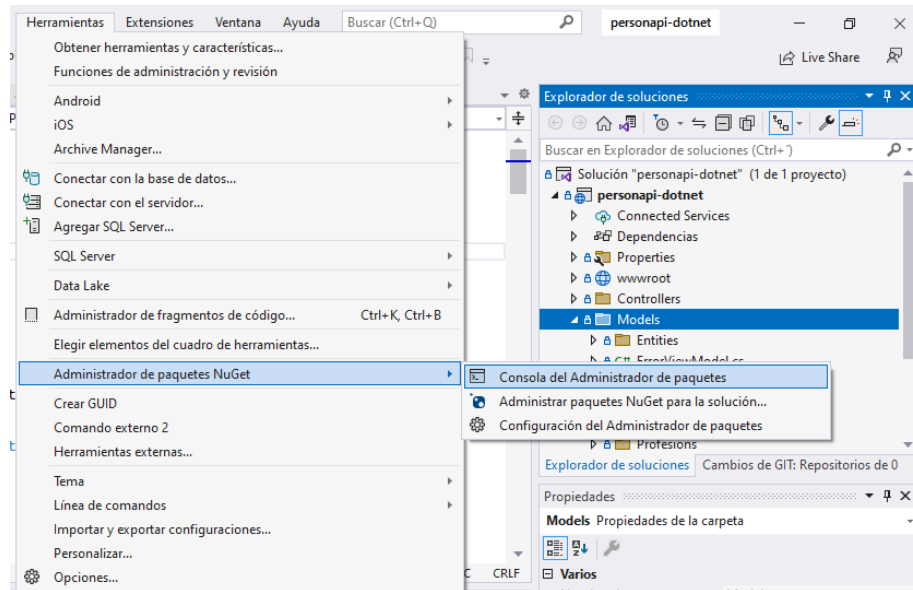
28. Ya al tenerlo instalado se da clic en instalado y aparecerán todas las dependencias



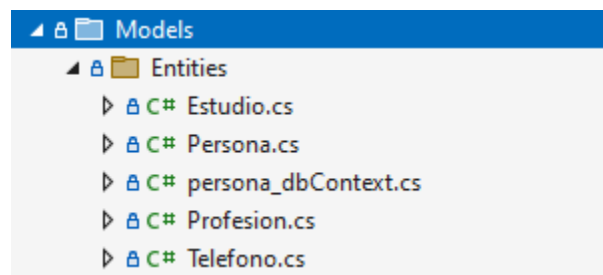
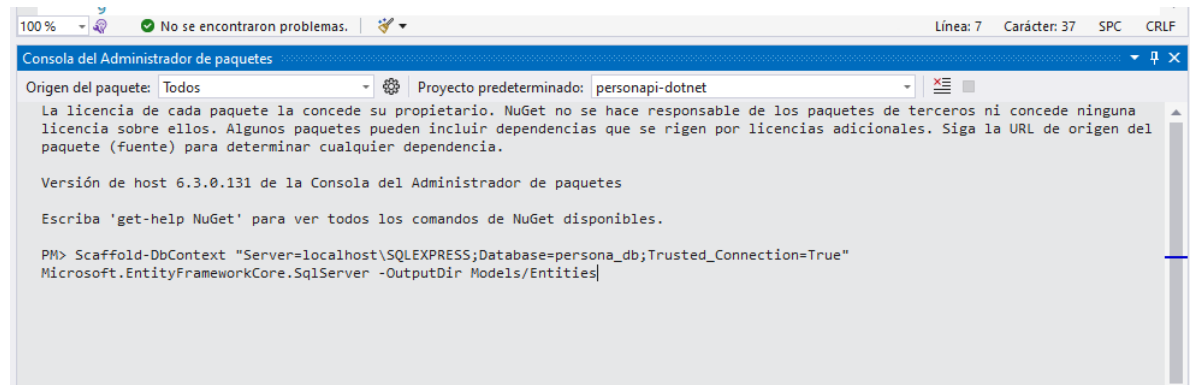
29. Luego en el explorador de soluciones se agregará en la pestaña de models una carpeta llamada Entities



30. Ir al menú herramientas, seleccionar administrador de paquetes NuGet, clic a la consola del administrador de paquetes



31. Luego se abrirá la consola en la parte inferior y se deberá escribir el comando Scaffold-DbContext "Server=localhost\SQLEXPRESS;Database=persona_db;Trusted_Connection=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models/Entities, esto tomará un momento mientras realiza un build y traerá todas las tablas de la base persona_db para la clases de la carpeta entities




```

1 using System;
2 using System.Collections.Generic;
3
4 namespace personapi_dotnet.Models.Entities
5 {
6     public partial class Estudio
7     {
8         public int IdProf { get; set; }
9         public int CcPer { get; set; }
10        public DateTime Fecha { get; set; }
11        public string Univer { get; set; } = null!;
12
13        public virtual Profesion IdProf1 { get; set; } = null!;
14        public virtual Persona IdProfNavigation { get; set; } = null!;
15    }
16 }
17

```

32. Se agrega la cadena de conexión en appsettings.json de esta manera

```

1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "ConnectionStrings": {
9     "DefaultConnection": "server = localhost; database = DemoApi; Trusted_Connection = true"
10  },
11   "AllowedHosts": "*"
12 }

```

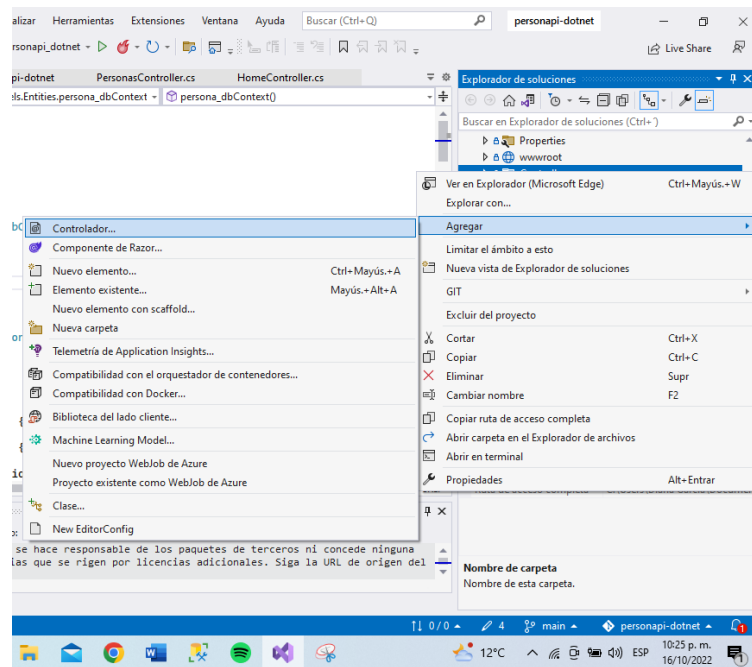
33. Se verifica en la entidad de persona_dbContext que no tenga ningún adicional en el constructor de la clase, se deberá tener así:

```

1 using System;
2 using System.Collections.Generic;
3 using Microsoft.EntityFrameworkCore;
4 using Microsoft.EntityFrameworkCore.Metadata;
5
6 namespace personapi_dotnet.Models.Entities
7 {
8     public partial class persona_dbContext : DbContext
9     {
10        public persona_dbContext()
11        {
12        }
13
14        public persona_dbContext(DbContextOptions<persona_dbContext> options)
15            : base(options)
16        {
17        }
18
19        public virtual DbSet<Estudio> Estudios { get; set; } = null!;
20        public virtual DbSet<Persona> Personas { get; set; } = null!;
21        public virtual DbSet<Profesion> Profesions { get; set; } = null!;
22    }
23 }

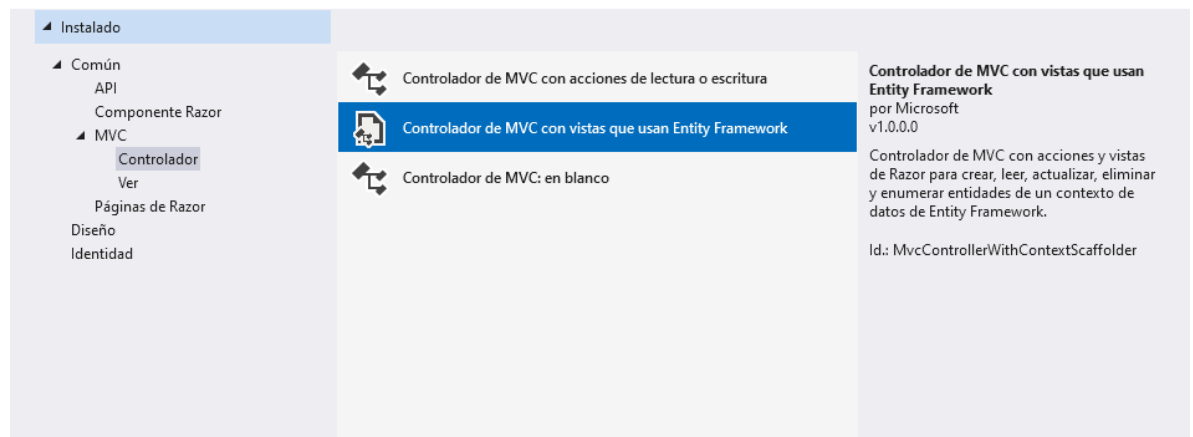
```

34. Se crean las interfaces, los controladores y los repositorios dándole clic derecho a la pestaña controllers



35. Se generará una pantalla con varias opciones, daremos clic a la opción Controlador de MVC

Agregar nuevo elemento con scaffolding



36. Se dará en aceptar y aparecerá otra pantalla que tendrá la información de la base de datos, este procedimiento se hizo por cada una de las tablas, lo que generó los controladores, las interfaces y los repositorios

Agregar Controlador de MVC con vistas que usan Entity Fram

Clase de modelo:

Clase de contexto de datos:

Vistas

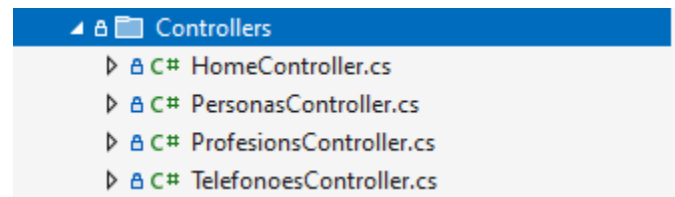
☒ Generar vistas

☒ Hacer referencia a bibliotecas de scripts

☒ Usar página de diseño

(Dejar en blanco si se define en un archivo _viewstart de Razor)

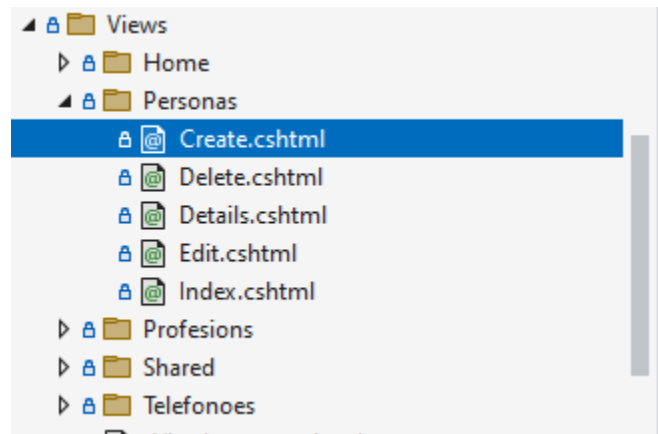
Nombre de controlador:



```

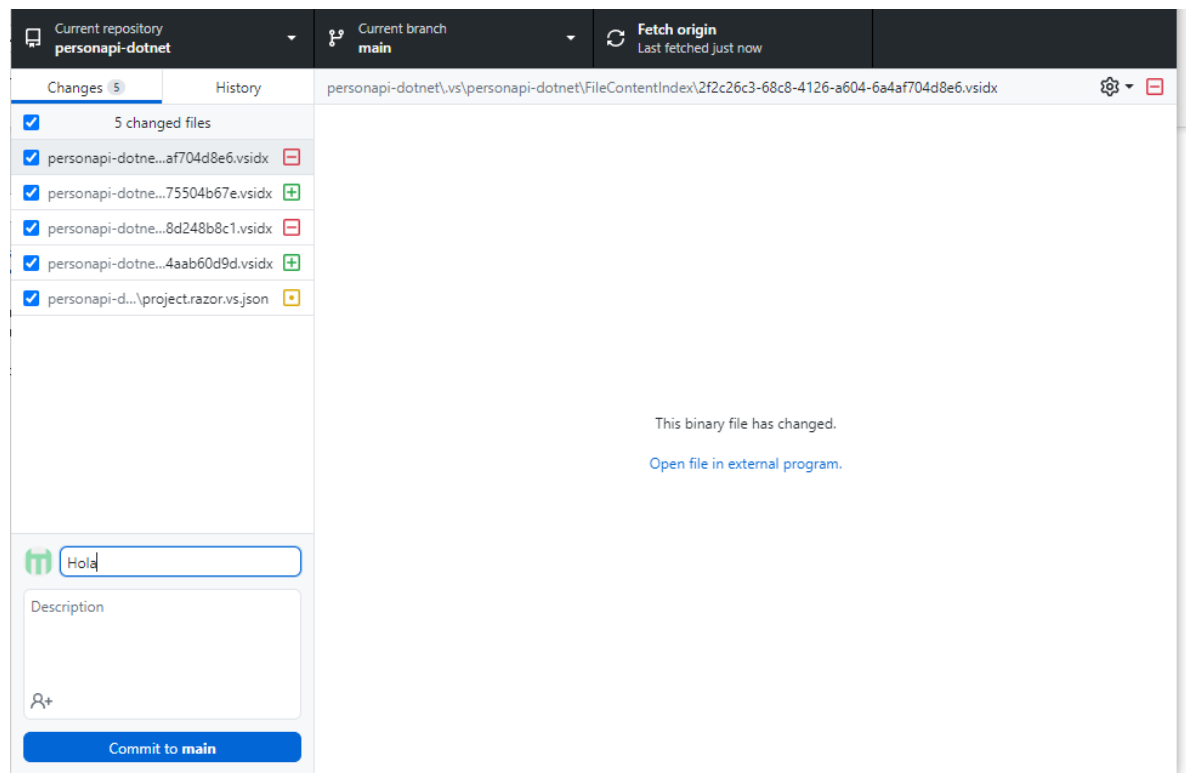
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using Microsoft.AspNetCore.Mvc;
6  using Microsoft.AspNetCore.Mvc.Rendering;
7  using Microsoft.EntityFrameworkCore;
8  using personapi_dotnet.Models.Entities;
9
10 namespace personapi_dotnet.Controllers
11 {
12     1 referencia
13     public class PersonasController : Controller
14     {
15         private readonly persona_dbContext _context;
16
17         0 referencias
18         public PersonasController(persona_dbContext context)
19         {
20             _context = context;
21         }
22
23         // GET: Personas
24         3 referencias
25         public async Task<IActionResult> Index()
26         {
27
28         }
29     }
30 }

```



37. Por último, se guardó todo el proyecto, se colocó en la carpeta donde se clonó el proyecto y utilizando el GitHub Desktop se realizó el commit poniendo la descripción y se realizó el push

Este equipo > Documentos > personapi-dotnet > personapi-dotnet					
	Nombre	Fecha de modificación	Tipo	Tamaño	
	.vs	16/10/2022 8:20 p. m.	Carpeta de archivos		
	personapi-dotnet	16/10/2022 8:20 p. m.	Carpeta de archivos		
	personapi-dotnet.sln	16/10/2022 6:30 p. m.	Visual Studio Solu...	2 KB	



Conclusiones y lecciones aprendidas

Se puede concluir que existen herramientas que contienen muchas características y eso permite que se puedan utilizar para diversas cosas al mismo tiempo, en este caso, el Visual Studio es una de las herramientas más completas que permiten un buen desarrollo, es una herramienta muy organizada que brinda al usuario un fácil uso y además, si no saben del tema permite aprender con facilidad. También, el Management Studio permite aparte de crear y manipular las tablas un entorno de diseño al crear los diagramas con acceso a todos los recursos disponibles lo que permite que se realice un uso de forma rápida y segura.

Lo que más nos gustó del laboratorio es la forma en que sólo al conectar la base de datos se integre totalmente con el modelo especificando cada entidad que se encuentra en cada tabla, inicializando sus características, el tipo de datos y todo de una forma automática, al igual que, al realizar los controladores e interfaces con la disponibilidad de la aplicación al hacerlo de forma genérica, pero con cada dato correctamente sin necesidad de que llene ningún campo por defecto.

Esto deja una reflexión de que hay que abarcar más estudios, hay que conocer nuevas herramientas que pueden ser de mejor utilidad a las que se está acostumbrado, esto permite un mayor conocimiento, aprender nuevas cosas cada día, salir de la zona de confort y de pronto al tener esa innovación se va facilitando un poco la vida en ámbitos de diseño y desarrollo.

Referencias

- [1]. *¿Cómo se utiliza Github pages? - Aprende sobre desarrollo web / MDN.* Developer.mozilla.org. (2022). Retrieved 15 October 2022, from https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages.
- [2]. (2022). Retrieved 15 October 2022, from <https://www.microsoft.com/es-cl/download/details.aspx?id=101064#:~:text=Microsoft%20SQL%20Server%202019%20Express,de%20fabricantes%20de%20software%20independiente>.
- [3]. Softtrader, M. (2022). *¿Qué es SQL Server Management Studio y qué puedo hacer con él?*. Softtrader. Retrieved 15 October 2022, from [https://softtrader.es/blog-microsoft/que-es-sql-server-management-studio/#:~:text=SQL%20Server%20Management%20Studio%20\(SSMS%20para%20abreviar\)%20es%20un%20entorno,SQL%20Server%20y%20SQL%20Database](https://softtrader.es/blog-microsoft/que-es-sql-server-management-studio/#:~:text=SQL%20Server%20Management%20Studio%20(SSMS%20para%20abreviar)%20es%20un%20entorno,SQL%20Server%20y%20SQL%20Database).
- [4]. *Visual Studio 2022 Community Edition: descargar la versión gratuita más reciente.* Visual Studio. (2022). Retrieved 15 October 2022, from <https://visualstudio.microsoft.com/es/vs/community/#:~:text=53%2D07%3A00-,Visual%20Studio%20Community,y%20servicios%20en%20la%20nube>.
- [5]. *Comenzar con GitHub Desktop.* GitHub Docs. (2022). Retrieved 15 October 2022, from <https://docs.github.com/es/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>.