1. Preparativos Iniciales\*\*

\*\*1.1. Instalación de Herramientas:\*\*

- \*\*Visual Studio 2022:\*\* Descárgalo desde [Visual Studio](https://visualstudio.microsoft.com/). Instala las herramientas necesarias para desarrollo en VB.NET.

- \*\*SQL Server 2022:\*\* Instala el [SQL Server 2022](https://www.microsoft.com/en-us/sql-server/sql-server-downloads) y SQL Server Management Studio (SSMS) para la gestión de bases de datos.

\*\*1.2. Configuración del Entorno:\*\*

- Asegúrate de que Visual Studio esté configurado para trabajar con VB.NET y SQL Server.

- Verifica la conexión entre Visual Studio y SQL Server usando el servidor y la base de datos de prueba.

#### \*\*2. Creación del Proyecto en Visual Studio\*\*

\*\*2.1. Crear un Nuevo Proyecto:\*\*

1. Abre Visual Studio 2022.

2. Selecciona \*\*"Create a new project"\*\*.

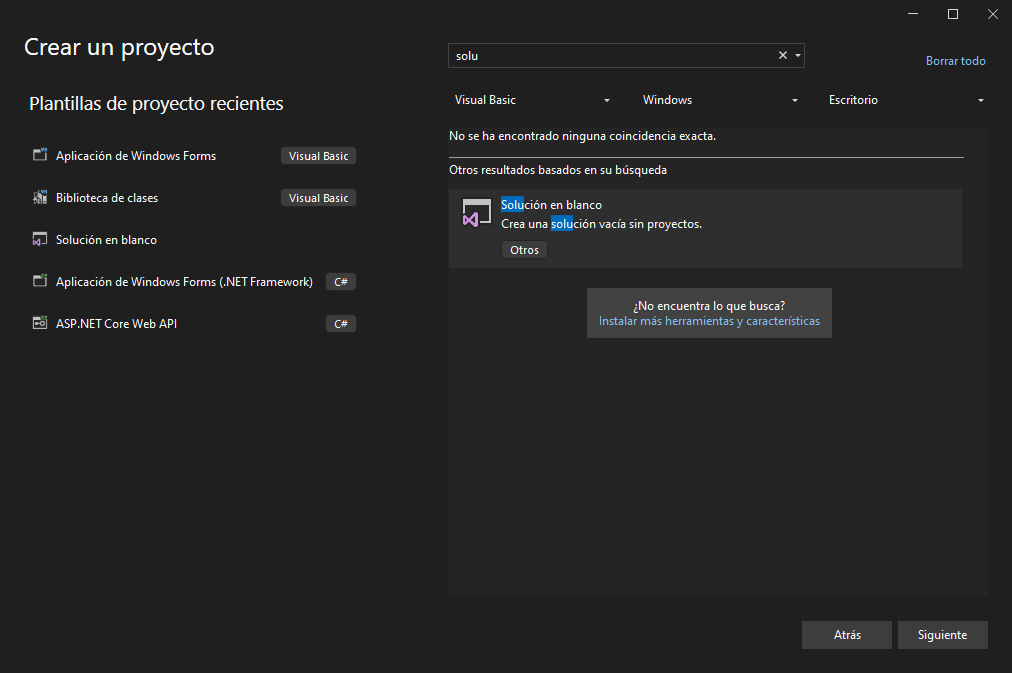
**2. Crear un nuevo proyecto**

* Haz clic en **"Crear un nuevo proyecto"** en la ventana de inicio de Visual Studio.
* En el cuadro de búsqueda, escribe **"Biblioteca de clases"**.
* Asegúrate de seleccionar **"Biblioteca de clases (.NET)"** para Visual Basic.
* Haz clic en **"Siguiente"**.

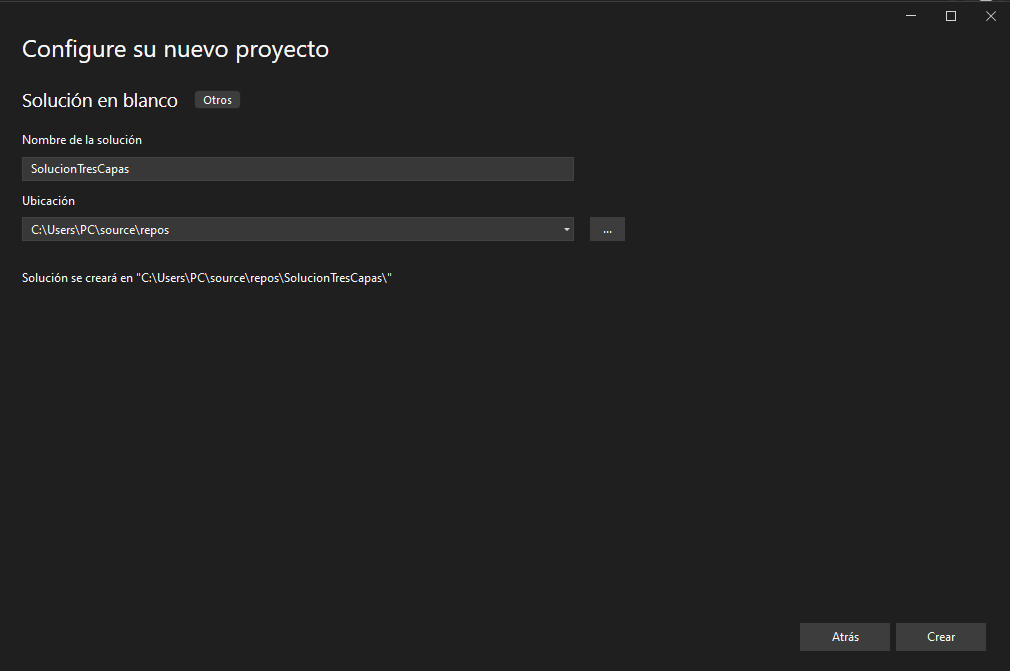
**3. Configurar el proyecto**

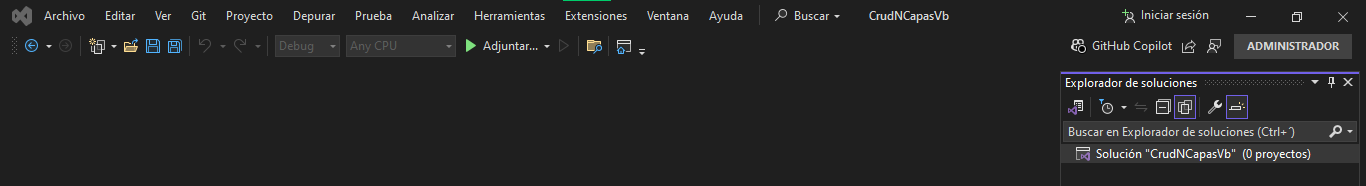
* **Nombre del proyecto**: Escribe un nombre para tu biblioteca de clases, por ejemplo, MiBiblioteca.
* **Ubicación**: Elige la ubicación donde quieres guardar el proyecto o deja la ubicación predeterminada.
* **Solución**: Puedes elegir crear una nueva solución o agregar el proyecto a una solución existente. Para empezar, generalmente es más sencillo crear una nueva solución.
* **Marco de trabajo**: Asegúrate de que el marco de trabajo (framework) seleccionado sea adecuado para tus necesidades. Por defecto, elige el .NET 6.0 o .NET Framework más reciente compatible con tu proyecto.
* Haz clic en **"Crear"**.

En este caso como es un proyecto con la arquitectura de tres capas comenzamos creando una solución en blanco.



Una vez seleccionado el proyecto debe de seleccionar siguiente, aparecerá una nueva ventana de configuración de nuevo proyecto en la cual se debe ingresarse el nombre de la solución en blanco y seleccionar la ruta donde se va almacenar el proyecto.

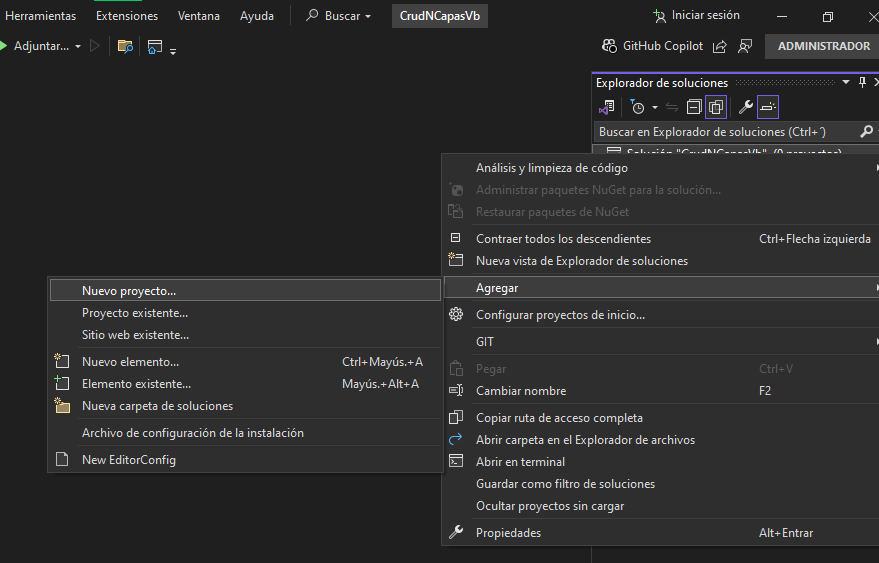




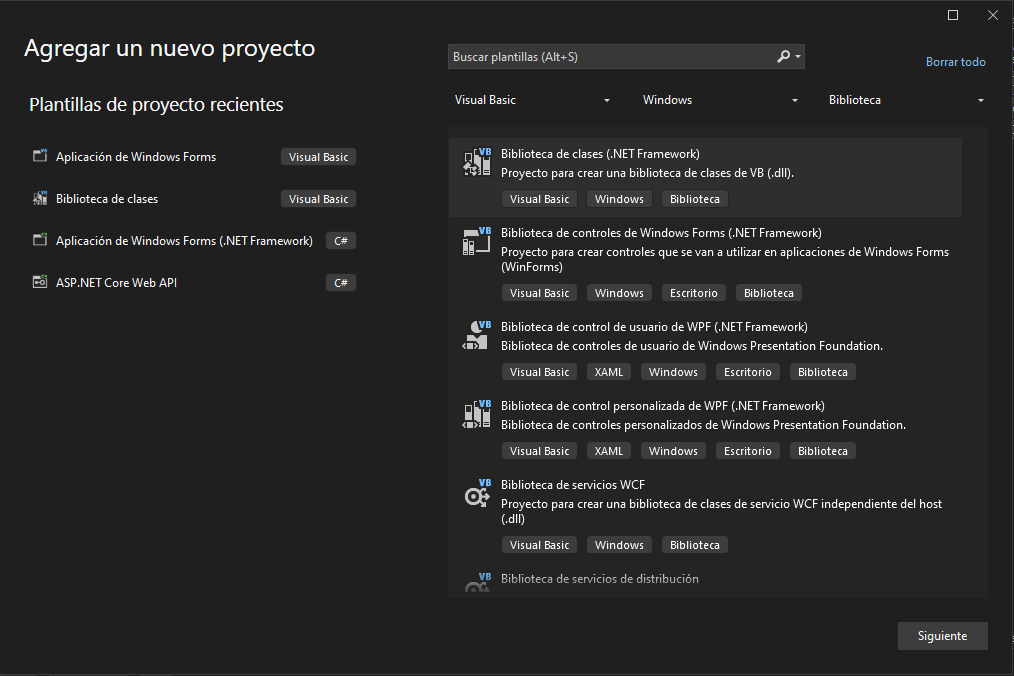
**3. Agregar proyectos a la solución**

**Ahora necesitas agregar tres proyectos a la solución, uno para cada capa (entidades, datos, negocio).**

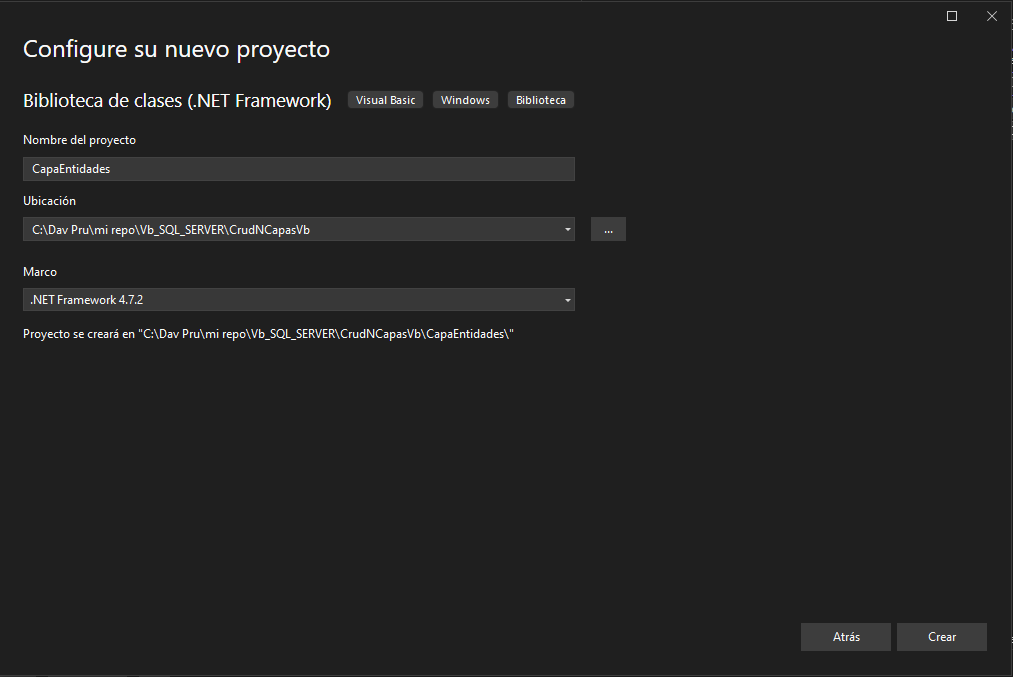
* **Agregar el proyecto de Entidades:**
  1. **Haz clic derecho en la solución en el Explorador de soluciones y selecciona "Agregar" > "Nuevo proyecto".**



* 1. **En el cuadro de búsqueda, escribe "Biblioteca de clases" y selecciona "Biblioteca de clases (.NET)" para Visual Basic.**

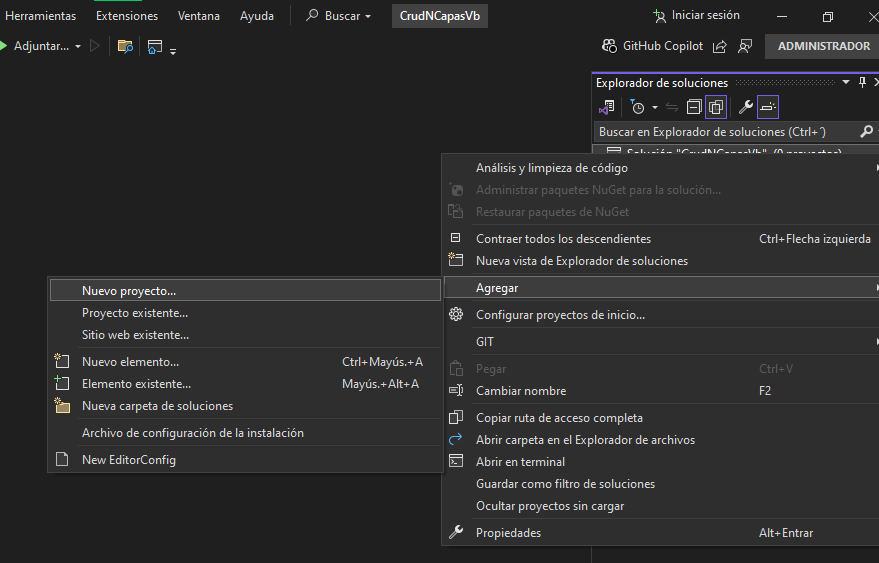
****

* 1. **Asigna un nombre al proyecto, por ejemplo, MiAplicacion.Entidades.**
  2. **Haz clic en "Crear".**

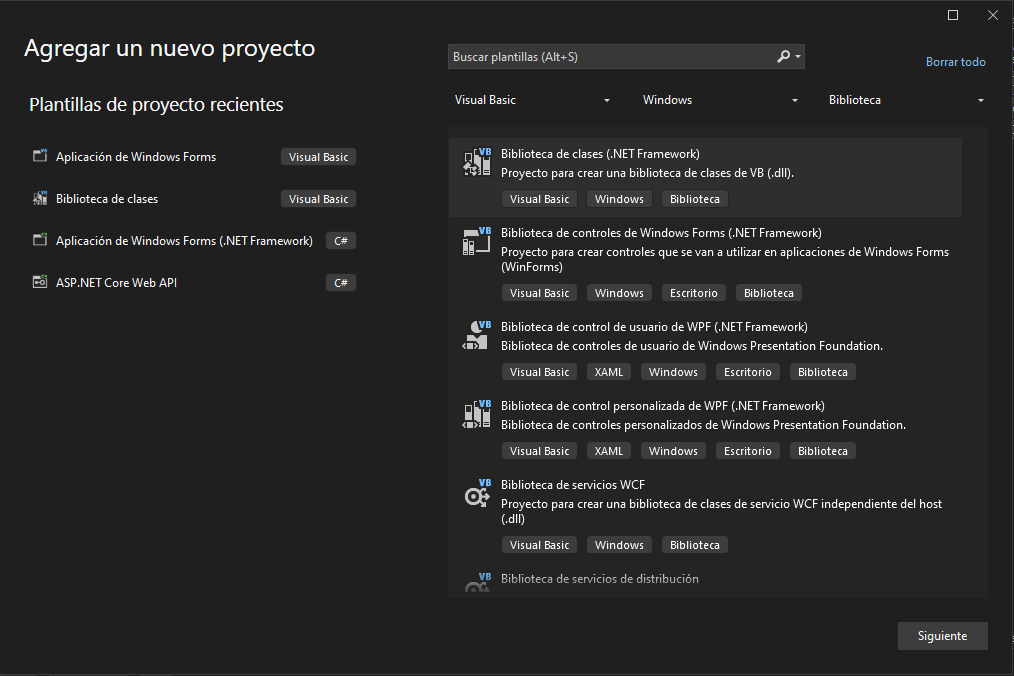
****

**Agregar el proyecto de Datos:**

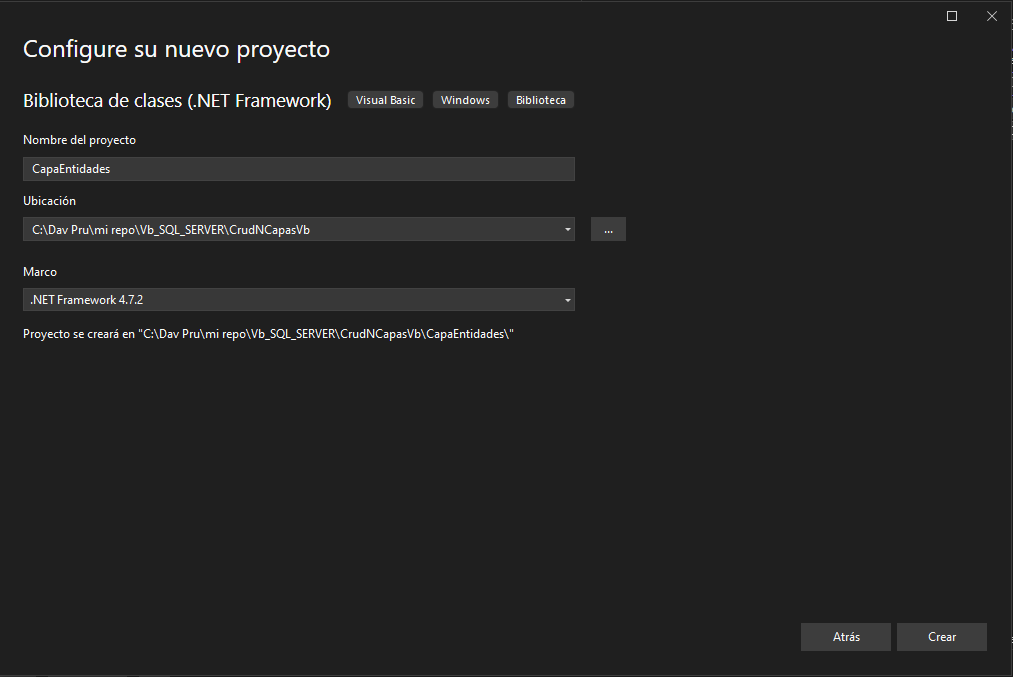
1. **Haz clic derecho en la solución en el Explorador de soluciones y selecciona "Agregar" > "Nuevo proyecto".**



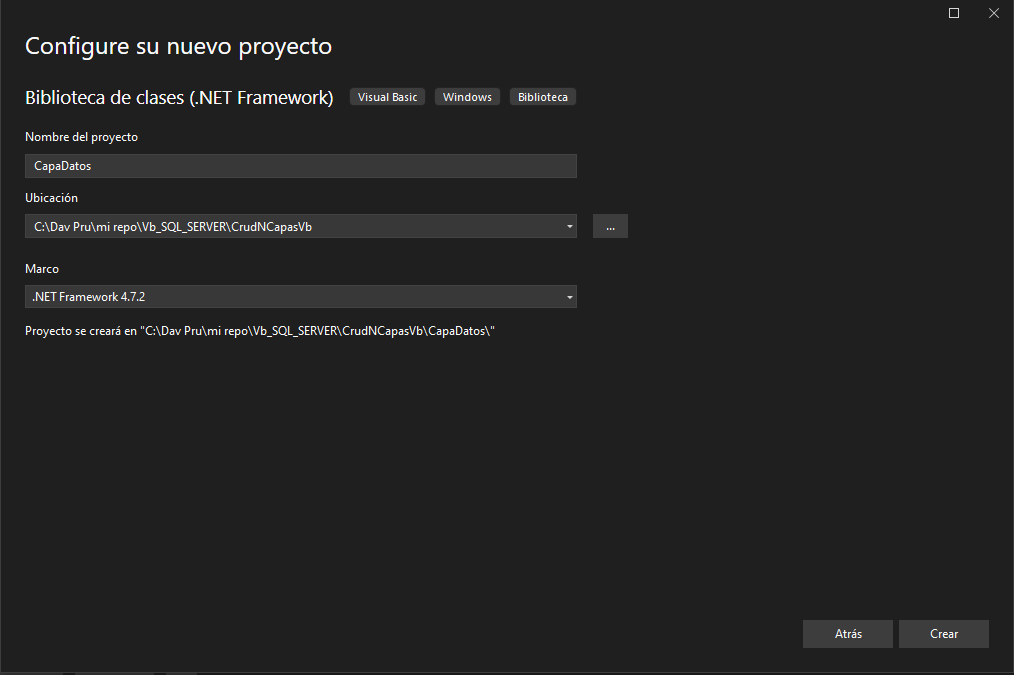
1. **Busca y selecciona "Biblioteca de clases (.NET)" para Visual Basic.**

****

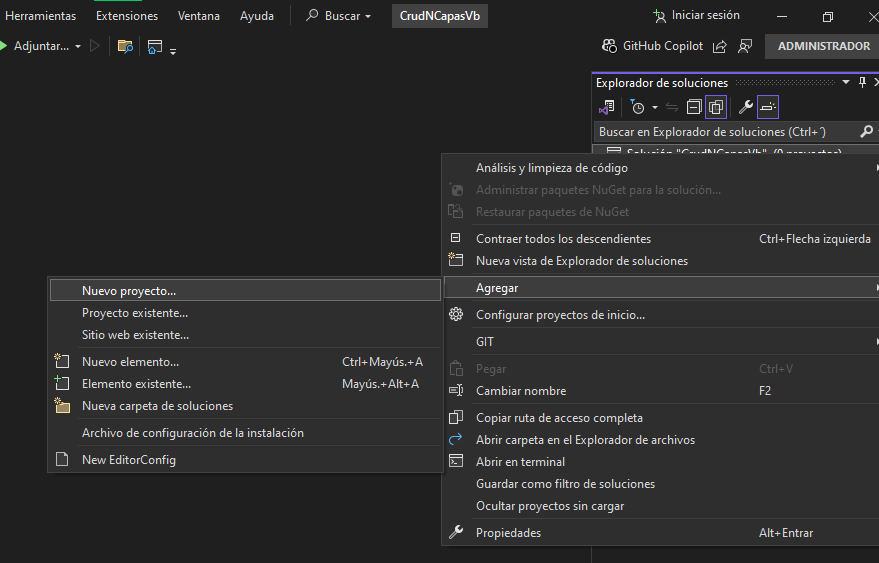
1. **Asigna un nombre al proyecto, por ejemplo, MiAplicacion.Datos.**

****

1. **Haz clic en "Crear".**

****

* **Agregar el proyecto de Negocio:**
  1. **Haz clic derecho en la solución en el Explorador de soluciones y selecciona "Agregar" > "Nuevo proyecto".**



* 1. **Busca y selecciona "Biblioteca de clases (.NET)" para Visual Basic.**
  2. **Asigna un nombre al proyecto, por ejemplo, MiAplicacion.Negocio.**
  3. **Haz clic en "Crear".**
* **Agregar el proyecto de Negocio:**
  1. Haz clic derecho en la solución en el **Explorador de soluciones** y selecciona **"Agregar"** > **"Nuevo proyecto"**.
  2. Busca y selecciona **"Biblioteca de clases (.NET)"** para Visual Basic.
  3. Asigna un nombre al proyecto, por ejemplo, MiAplicacion.Negocio.
  4. Haz clic en **"Crear"**.

**4. Configurar las referencias entre proyectos**

* **Referencias del proyecto de Negocio:**
  1. **Haz clic derecho en el proyecto MiAplicacion.Negocio en el Explorador de soluciones y selecciona "Agregar" > "Referencia".**
  2. **En la ventana de referencia, selecciona "Proyectos" en el panel izquierdo.**
  3. **Marca los proyectos MiAplicacion.Entidades y MiAplicacion.Datos.**
  4. **Haz clic en "Aceptar".**
* **Referencias del proyecto de Datos:**
  1. **Haz clic derecho en el proyecto MiAplicacion.Datos y selecciona "Agregar" > "Referencia".**
  2. **En la ventana de referencia, selecciona "Proyectos" en el panel izquierdo.**
  3. **Marca el proyecto MiAplicacion.Entidades.**
  4. **Haz clic en "Aceptar".**

**5. Implementar las capas**

* **Capa de Entidades (MiAplicacion.Entidades):**
  1. **En el proyecto MiAplicacion.Entidades, agrega una nueva clase.**
  2. **Asigna un nombre a la clase, por ejemplo, Empleado.vb.**
  3. **Define la clase con propiedades que representen los atributos del empleado.**

**Capa de Datos (MiAplicacion.Datos):**

1. **En el proyecto MiAplicacion.Datos, agrega una nueva clase.**
2. **Asigna un nombre a la clase, por ejemplo, EmpleadoRepository.vb.**
3. **Define la clase con métodos para acceder a la base de datos.**

**Capa de Negocio (MiAplicacion.Negocio):**

1. **En el proyecto MiAplicacion.Negocio, agrega una nueva clase.**
2. **Asigna un nombre a la clase, por ejemplo, EmpleadoService.vb.**
3. **Define la clase con métodos que usan el repositorio de datos.**

**6. Probar la implementación**

* **Agrega un proyecto de aplicación (como una aplicación de consola) a la solución para probar la capa de negocio.**
* **Haz clic derecho en la solución y selecciona "Agregar" > "Nuevo proyecto".**
* **Selecciona "Aplicación de consola (.NET)" y asígnale un nombre, por ejemplo, MiAplicacion.Prueba.**
* **En el proyecto MiAplicacion.Prueba, agrega código para probar la capa de negocio:**

3. Elige \*\*"Windows Forms App (.NET Framework)"\*\* para VB.NET y haz clic en \*\*"Next"\*\*.

4. Asigna un nombre al proyecto y elige la ubicación donde se guardará.

5. Haz clic en \*\*"Create"\*\*.

\*\*2.2. Estructura del Proyecto:\*\*

El proyecto se creará con una estructura básica. A continuación, debes crear las carpetas para organizar las diferentes capas de tu aplicación.

#### \*\*3. Creación de la Capa de Datos\*\*

\*\*3.1. Diseño de la Base de Datos:\*\*

1. Abre SQL Server Management Studio (SSMS).

2. Conéctate a tu servidor SQL Server.

3. Crea una nueva base de datos para tu proyecto.

```sql

CREATE DATABASE MiBaseDeDatos;

```

4. Diseña tus tablas, relaciones y otros objetos necesarios para la base de datos.

\*\*3.2. Creación de la Capa de Datos en Visual Studio:\*\*

1. En el Explorador de Soluciones en Visual Studio, agrega un nuevo proyecto de tipo \*\*"Class Library (.NET Framework)"\*\* y nómbralo `Datos`.

2. Dentro de este proyecto, agrega una nueva clase para cada tabla en tu base de datos. Utiliza el siguiente código para una clase básica:

```vb

Public Class Cliente

Public Property Id As Integer

Public Property Nombre As String

Public Property Direccion As String

End Class

```

3. Agrega una clase `DataAccess` para manejar la conexión a la base de datos y las operaciones CRUD (Crear, Leer, Actualizar, Eliminar):

```vb

Imports System.Data.SqlClient

Public Class DataAccess

Private connectionString As String = "Server=tuServidor;Database=MiBaseDeDatos;Trusted\_Connection=True;"

Public Function ObtenerClientes() As List(Of Cliente)

Dim clientes As New List(Of Cliente)()

Using connection As New SqlConnection(connectionString)

Dim command As New SqlCommand("SELECT \* FROM Clientes", connection)

connection.Open()

Dim reader As SqlDataReader = command.ExecuteReader()

While reader.Read()

clientes.Add(New Cliente() With {

.Id = Convert.ToInt32(reader("Id")),

.Nombre = reader("Nombre").ToString(),

.Direccion = reader("Direccion").ToString()

})

End While

End Using

Return clientes

End Function

End Class

```

#### \*\*4. Creación de la Capa de Negocio\*\*

\*\*4.1. Crear el Proyecto de la Capa de Negocio:\*\*

1. Agrega otro proyecto de tipo \*\*"Class Library (.NET Framework)"\*\* y nómbralo `Negocio`.

\*\*4.2. Implementar la Lógica de Negocio:\*\*

1. Dentro del proyecto `Negocio`, agrega una clase que maneje la lógica de negocio. Por ejemplo, para la gestión de clientes:

```vb

Public Class GestionClientes

Private \_dataAccess As New DataAccess()

Public Function ObtenerClientes() As List(Of Cliente)

Return \_dataAccess.ObtenerClientes()

End Function

End Class

```

#### \*\*5. Creación de la Capa de Presentación\*\*

\*\*5.1. Agregar una Referencia a los Proyectos:\*\*

1. Agrega referencias a los proyectos `Datos` y `Negocio` desde el proyecto principal de la aplicación de Windows Forms.

\*\*5.2. Diseñar la Interfaz de Usuario:\*\*

1. Abre el formulario principal (`Form1.vb`) en el proyecto de Windows Forms.

2. Utiliza el diseñador de Windows Forms para agregar controles como `DataGridView`, `TextBox`, `Button`, etc.

\*\*5.3. Implementar la Lógica del Formulario:\*\*

1. En el código del formulario (`Form1.vb`), instancia las clases de la capa de negocio y maneja los eventos de los controles para interactuar con el usuario:

```vb

Public Class Form1

Private \_gestionClientes As New GestionClientes()

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CargarClientes()

End Sub

Private Sub CargarClientes()

Dim clientes As List(Of Cliente) = \_gestionClientes.ObtenerClientes()

DataGridView1.DataSource = clientes

End Sub

End Class

```

#### \*\*6. Pruebas y Validación\*\*

\*\*6.1. Ejecutar la Aplicación:\*\*

- Inicia la aplicación desde Visual Studio y verifica que la interfaz de usuario funcione como se espera y que los datos se muestren correctamente.

\*\*6.2. Validar Funcionalidad:\*\*

- Asegúrate de que todas las funciones CRUD estén implementadas y funcionando adecuadamente.

\*\*6.3. Manejo de Errores:\*\*

- Implementa manejo de errores tanto en la capa de datos como en la de negocio para manejar cualquier excepción que pueda ocurrir.

#### \*\*7. Documentación del Proyecto\*\*

\*\*7.1. Documentar el Código:\*\*

- Utiliza comentarios XML en tu código para documentar las clases, métodos y propiedades.

```vb

''' <summary>

''' Obtiene la lista de clientes desde la base de datos.

''' </summary>

''' <returns>Lista de clientes.</returns>

Public Function ObtenerClientes() As List(Of Cliente)

' Código...

End Function

```

\*\*7.2. Crear Documentación Adicional:\*\*

- Elabora un documento que explique la estructura de la arquitectura de tres capas, las clases principales, y cómo se interrelacionan.

\*\*7.3. Incluir Diagramas:\*\*

- Utiliza diagramas UML para representar la arquitectura del proyecto y las relaciones entre las clases.

¡Con esta guía, deberías poder crear un proyecto de aplicación de escritorio siguiendo la arquitectura de tres capas usando Visual Studio 2022 y SQL Server 2022 con VB.NET! Si tienes alguna pregunta o necesitas más detalles, no dudes en preguntar.