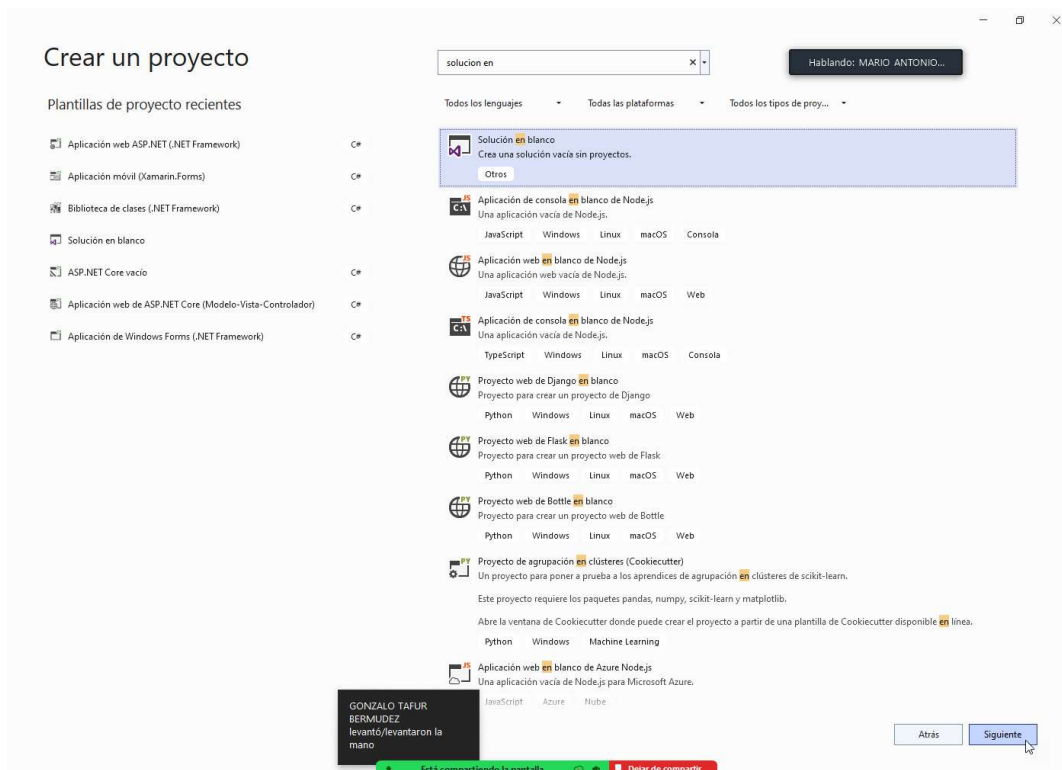
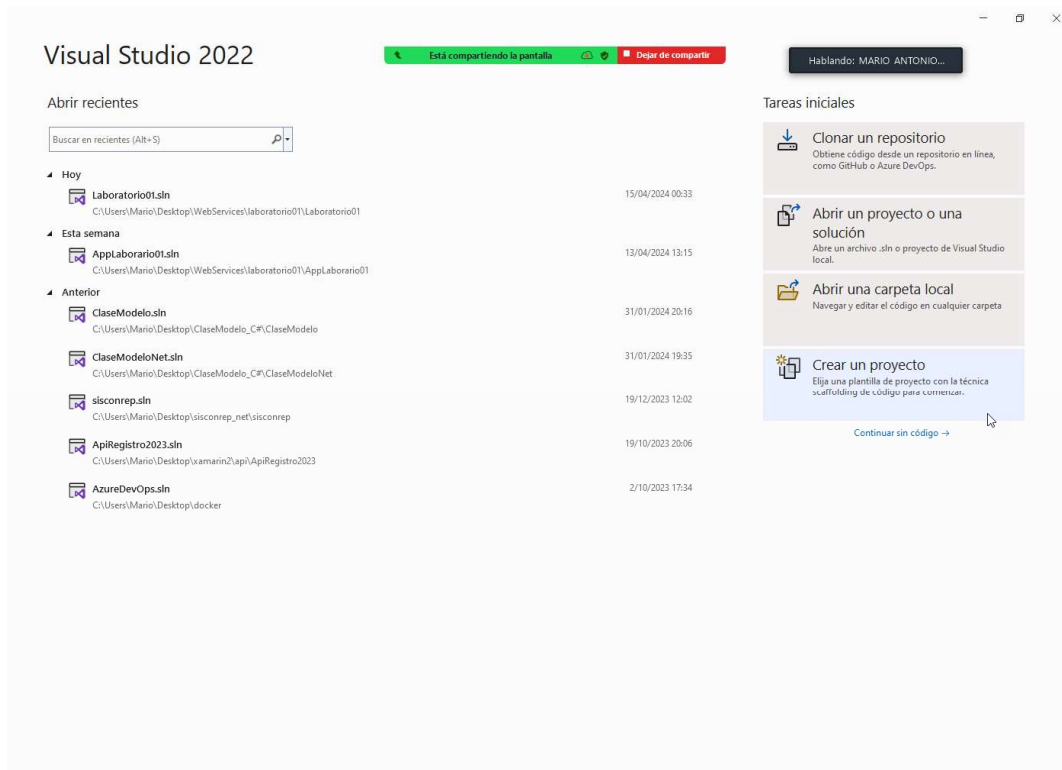


Creación de un Servicio Web con .Net



Configure su nuevo proyecto

Hablando: MARIO ANTONIO...

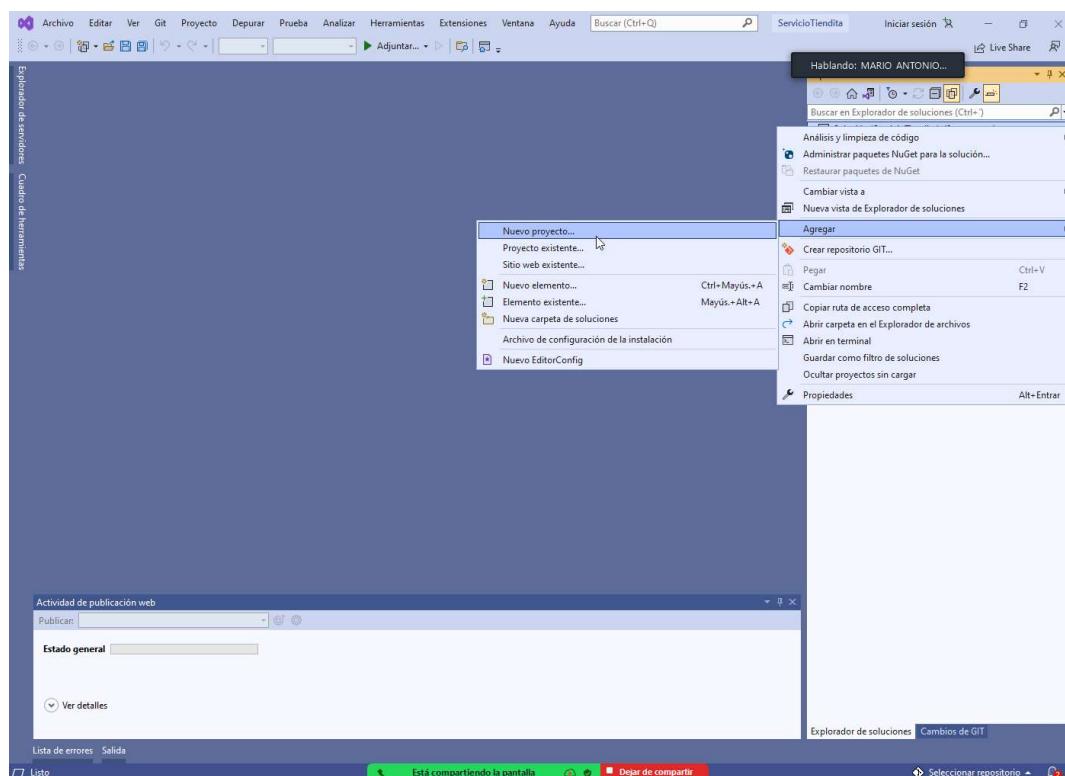
Solución en blanco Otros

Nombre de la solución

Ubicación

Está compartiendo la pantalla Dejar de compartir

Atrás Crear



Visual Studio IDE interface showing the 'File' menu and the 'New' submenu. The 'New' submenu is open, displaying options like 'Nuevo proyecto...', 'Proyecto existente...', 'Sitio web existente...', 'Nuevo elemento...', 'Elemento existente...', 'Nueva carpeta de soluciones', 'Archivo de configuración de la instalación', and 'Nuevo EditorConfig'. The 'File' menu is also open, showing options like 'Análisis y limpieza de código', 'Administrar paquetes NuGet para la solución...', 'Restaurar paquetes de NuGet', 'Cambiar vista a', 'Nueva vista de Explorador de soluciones', 'Agregar', 'Crear repositorio GIT...', 'Pegar', 'Cambiar nombre', 'Copiar ruta de acceso completa', 'Abrir carpeta en el Explorador de archivos', 'Abrir en terminal', 'Guardar como filtro de soluciones', 'Ocultar proyectos sin cargar', and 'Propiedades'.

Agregar un nuevo proyecto

Plantillas de proyecto recientes

- Aplicación web ASP.NET (.NET Framework)
- Aplicación móvil (Xamarin.Forms)
- Biblioteca de clases (.NET Framework)
- ASP.NET Core vacío
- Aplicación web de ASP.NET Core (Modelo-Vista-Controlador)
- Aplicación de Windows Forms (.NET Framework)

buscando biblioteca

Hablando: MARIO ANTONIO...

Todos los lenguajes Todas las plataformas Todos los tipos de proy...

Biblioteca de clases
Proyecto para crear una biblioteca de clases para .NET Standard o .NET Core.
C# Android Linux macOS Windows Biblioteca

Biblioteca de clases (.NET Framework)
Proyecto para crear una biblioteca de clases de C# (.dll).
C# Windows Biblioteca

Biblioteca de control personalizada de WPF (.NET Framework)
Biblioteca de controles personalizados de Windows Presentation Foundation.
C# XAML Windows Escritorio Biblioteca

Biblioteca de control de usuario de WPF (.NET Framework)
Biblioteca de controles de usuario de Windows Presentation Foundation.
C# XAML Windows Escritorio Biblioteca

Biblioteca de controles de Windows Forms (.NET Framework)
Proyecto para crear controles que se van a utilizar en aplicaciones de Windows Forms (WinForms).
C# Windows Escritorio Biblioteca

Biblioteca de servicios WCF
Proyecto para crear una biblioteca de clases de servicio WCF independiente del host (.dll).
C# Windows Biblioteca Web Servicio

Biblioteca de clases (.NET Framework)
Proyecto para crear una biblioteca de clases de VB (.dll).
Visual Basic Windows Biblioteca

Biblioteca de controles de Windows Forms (.NET Framework)
Proyecto para crear controles que se van a utilizar en aplicaciones de Windows Forms (WinForms).
Visual Basic Windows Escritorio Biblioteca

Biblioteca de control de usuario de WPF (.NET Framework)
Biblioteca de controles de usuario de Windows Presentation Foundation.
Visual Basic XAML Windows Escritorio Biblioteca

Biblioteca de control personalizada de WPF (.NET Framework)
Biblioteca de controles personalizados de Windows Presentation Foundation.

Siguiente

Está compartiendo la pantalla Dejar de compartir

Configure su nuevo proyecto

Hablando:

Biblioteca de clases (.NET Framework) C# Windows Biblioteca

Nombre del proyecto
pe.com.tiendita.dal

Ubicación
C:\Users\Mario\Desktop\net\laboratorio01\ServicioTiendita

Framework
.NET Framework 4.8

Atrás Crear

Está compartiendo la pantalla Dejar de compartir

Configure su nuevo proyecto

Hablando: MARIO ANTONIO...

Biblioteca de clases (.NET Framework) C# Windows Biblioteca

Nombre del proyecto

Ubicación

...

Framework

Atrás

Crear

Agregar un nuevo proyecto

Plantillas de proyecto recientes

- Biblioteca de clases (.NET Framework)
- Aplicación web ASP.NET (.NET Framework)
- Aplicación móvil (Xamarin.Forms)
- ASP.NET Core vacío
- Aplicación web de ASP.NET Core (Modelo-Vista-Controlador)
- Aplicación de Windows Forms (.NET Framework)

asp

Hablando: MARIO ANTONIO...

Todos los lenguajes Todas las plataformas Todos los tipos de proy...

Una plantilla de proyecto para crear una aplicación ASP.NET Core MVC. Esta plantilla también puede usarse para controladores y vistas de ASP.NET Core MVC.

C# Linux macOS Windows Nube Servicio Web

gRPC Servicio gRPC de ASP.NET Core
Plantilla de proyecto para crear un servicio gRPC de ASP.NET Core.

C# Linux macOS Windows Nube Servicio Web

ASP.NET Core con Angular
Plantilla de proyecto para crear una aplicación ASP.NET Core con Angular.

C# Linux macOS Windows Nube Servicio Web

ASP.NET Core con React.js
Plantilla de proyecto para crear una aplicación ASP.NET Core con React.js.

C# Linux macOS Windows Nube Servicio Web

ASP.NET Core con React.js y Redux
Plantilla de proyecto para crear una aplicación ASP.NET Core con React.js y Redux.

C# Linux macOS Windows Nube Servicio Web

Aplicación web ASP.NET (.NET Framework)
Plantillas de proyecto para crear aplicaciones ASP.NET. Puede crear aplicaciones ASP.NET Web Forms, MVC o Web API y agregar muchas otras características en ASP.NET.

C# Windows Nube Web

Aplicación web ASP.NET (.NET Framework)
Plantillas de proyecto para crear aplicaciones ASP.NET. Puede crear aplicaciones ASP.NET Web Forms, MVC o Web API y agregar muchas otras características en ASP.NET.

Visual Basic Windows Nube Web

ASP.NET Core vacío
Una plantilla de proyecto vacía para crear una aplicación ASP.NET Core. Esta plantilla no incluye ningún contenido.

F# Linux macOS Windows Nube Servicio Web

Aplicación web de ASP.NET Core (Modelo-Vista-Controlador)
Una plantilla de proyecto para crear una aplicación ASP.NET Core con controladores y vistas de ASP.NET Core MVC de ejemplo. Esta plantilla también puede usarse para servicios RESTful HTTP.

F# Linux macOS Windows Nube Servicio Web

ASP.NET Core Web API

Siguiente

Está compartiendo la pantalla Dejar de compartir

Configure su nuevo proyecto

Aplicación web ASP.NET (.NET Framework) C# Windows Nube Web

Hablando: MARIO ANTONIO...

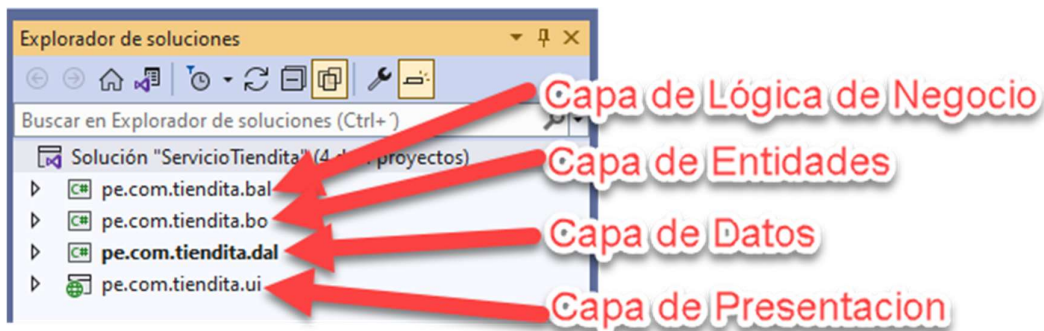
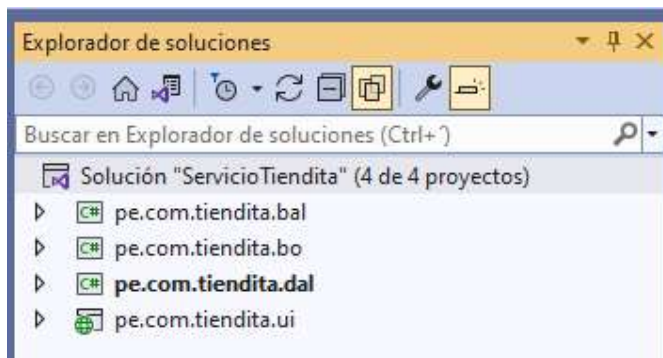
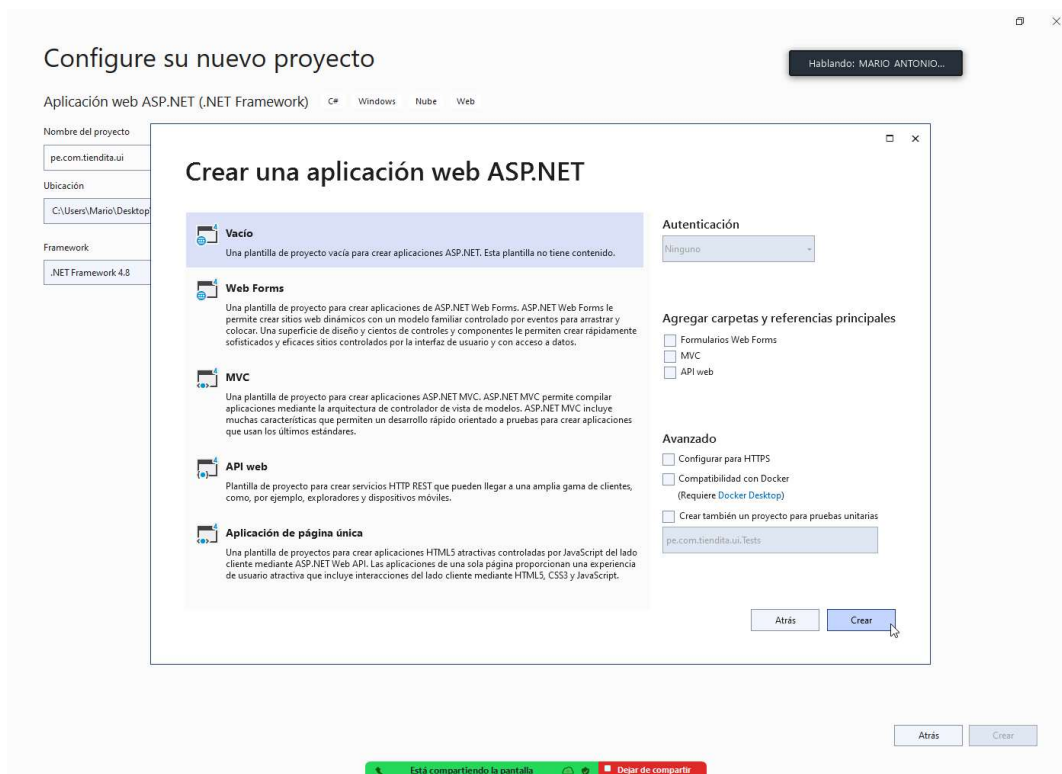
Nombre del proyecto
pe.com.tiendita.ui

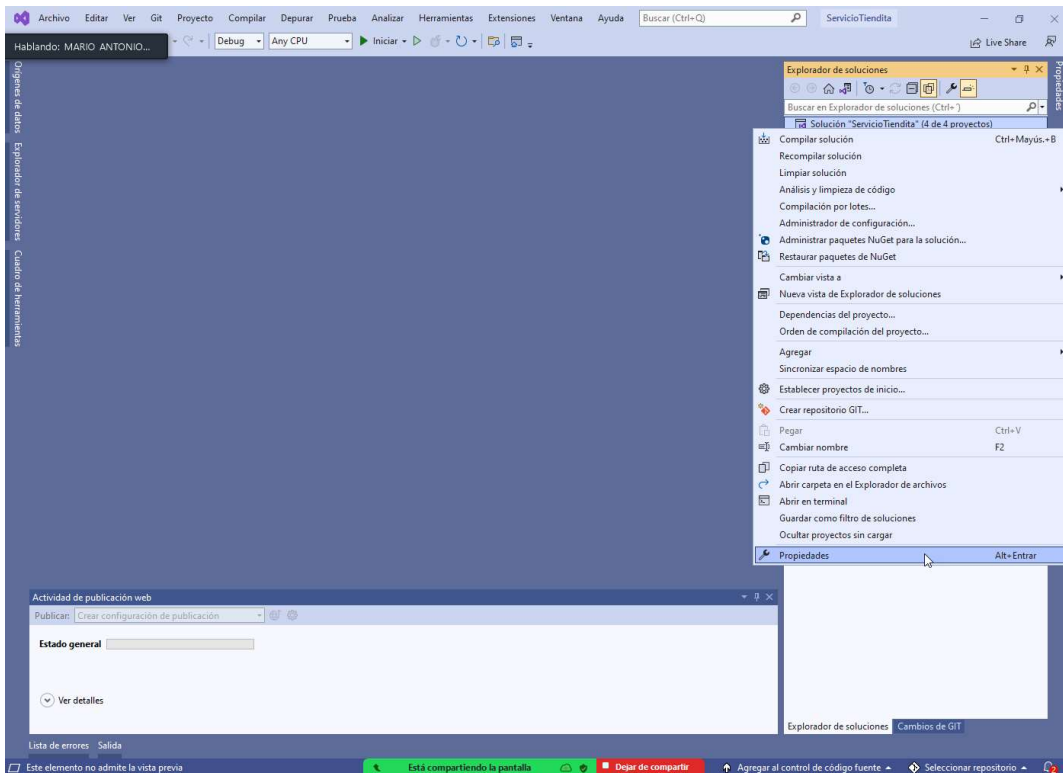
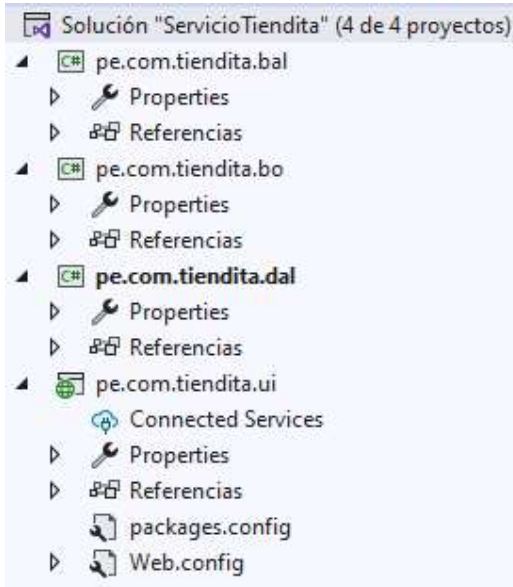
Ubicación
C:\Users\Mario\Desktop\net\laboratorio01\ServicioTiendita

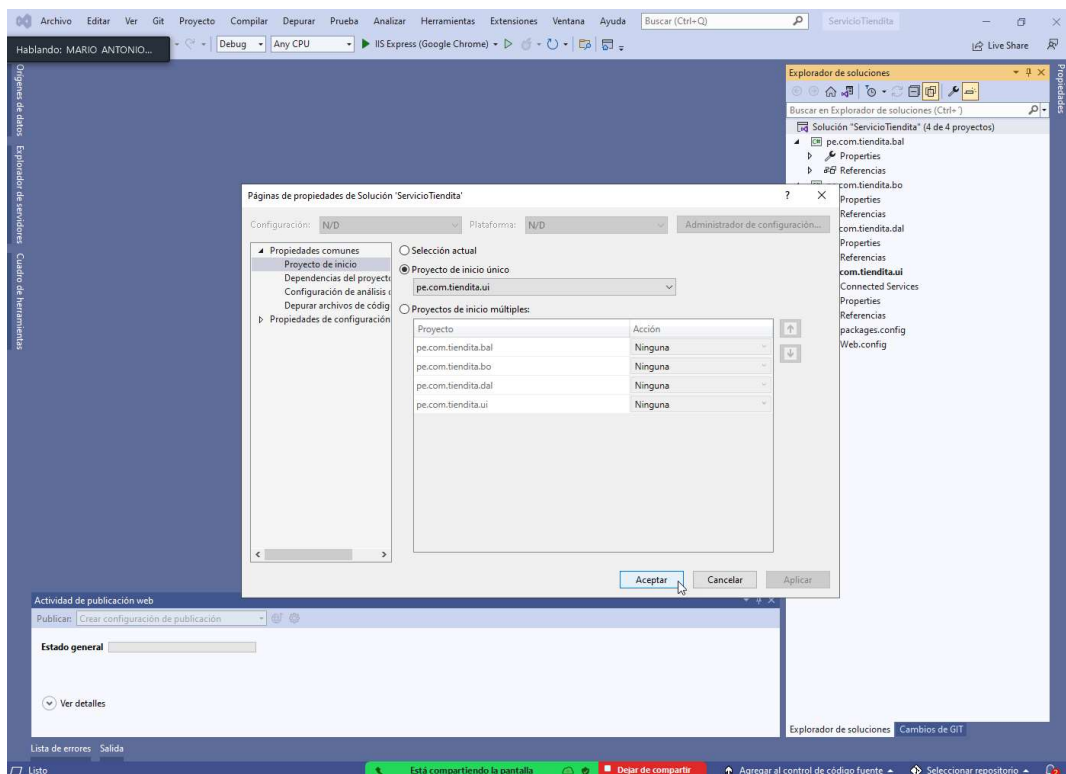
Framework
.NET Framework 4.8

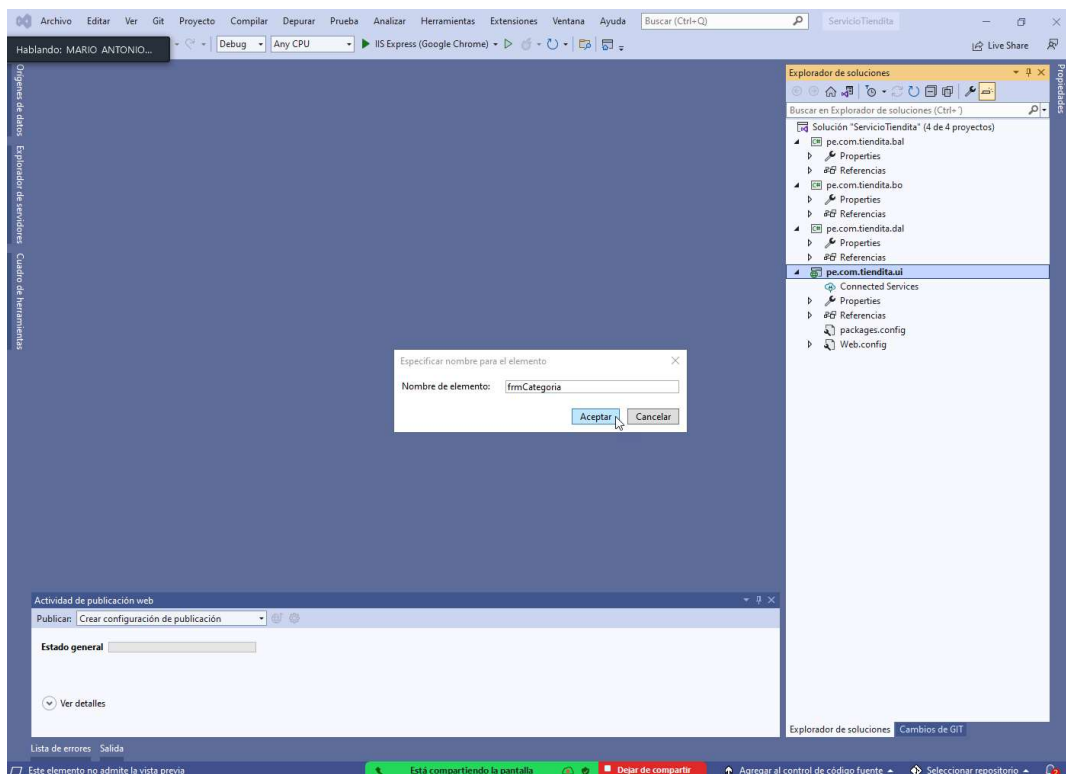
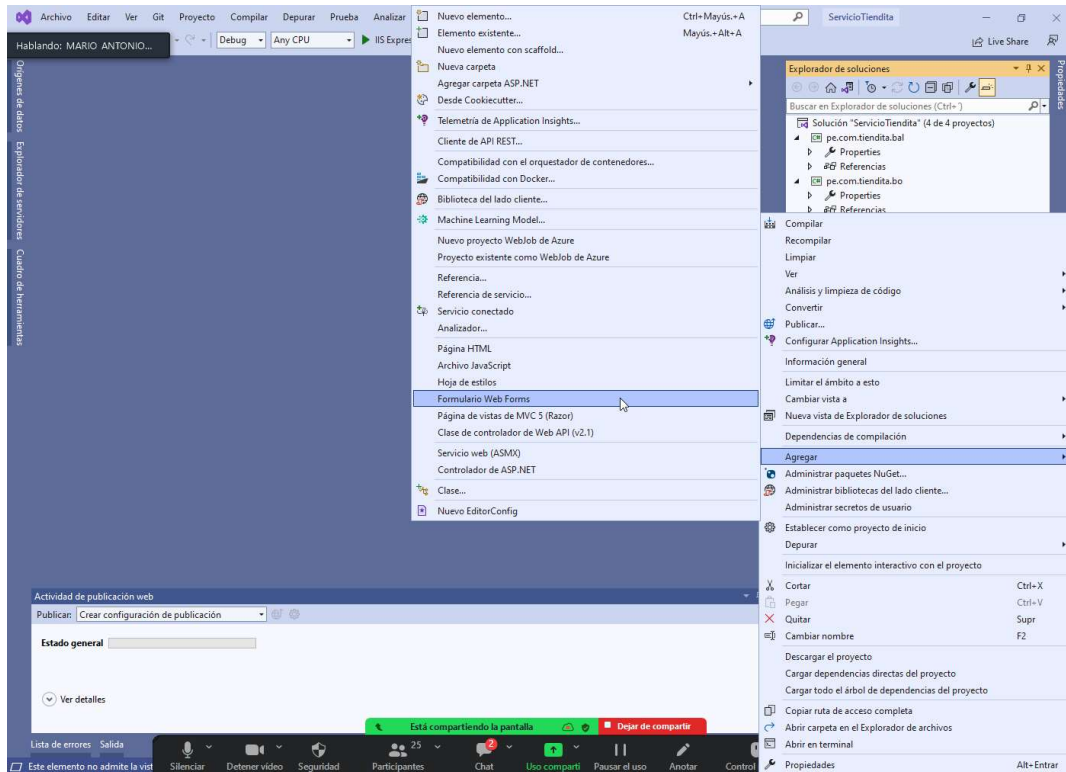
Atrás Crear

Está compartiendo la pantalla Dejar de compartir

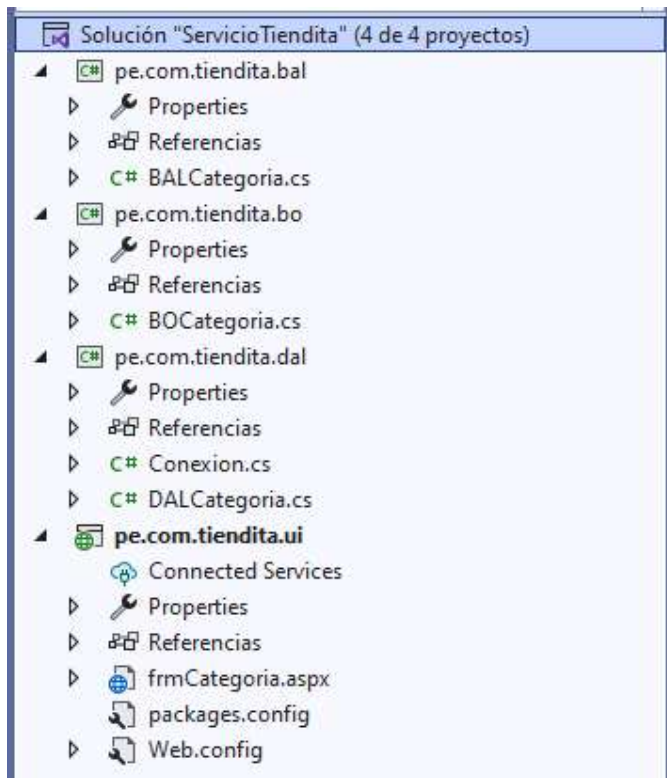








Bdtiendita.sql



Conexion.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace pe.com.tiendita.dal
{
    public class Conexion
    {
        //cadena de conexion
        private string cadena = "Data Source=.;Initial
Catalog=bdtiendita;Integrated Security=True";

        private SqlConnection xcon;

        public SqlConnection Conectar()
        {
            xcon = new SqlConnection(cadena);
            xcon.Open();
            return xcon;
        }

        public void CerrarConexion()
        {
            xcon.Close();
            xcon.Dispose();
        }
    }
}
```

```

    }
}

```

BOCategoria.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace pe.com.tiendita.bo
{
    public class BOCategoria
    {
        public int codigo { get; set; }
        public string nombre { get; set; }
        public bool estado { get; set; }

        public BOCategoria()
        {
        }
    }
}

```

DALCategoria.cs

```

using pe.com.tiendita.bo;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace pe.com.tiendita.dal
{
    public class DALCategoria
    {
        Conexion objconexion = new Conexion();

        private SqlCommand cmdcat;
        private SqlDataReader drcat;
        int res = 0;

        public List<BOCategoria> MostrarCategoria()
        {
            List<BOCategoria> categorias = new List<BOCategoria>();
            try
            {
                cmdcat = new SqlCommand();
                cmdcat.CommandType = CommandType.StoredProcedure;
                cmdcat.CommandText = "SP_MostrarCategoria";
                cmdcat.Connection = objconexion.Conectar();
                drcat = cmdcat.ExecuteReader();
                while (drcat.Read())
                {
                    BOCategoria objgen = new BOCategoria();

```

```

        objgen.codigo =
Convert.ToInt32(drcat["codcat"].ToString());
        objgen.nombre = drcat["nomcat"].ToString();
        objgen.estado =
Convert.ToBoolean(drcat["estcat"].ToString());
        categorias.Add(objgen);
    }
    return categorias;
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
    return null;
}
finally
{
    objconexion.CerrarConexion();
}
}

public List<BOCategoria> MostrarCategoriaTodo()
{
    List<BOCategoria> categorias = new List<BOCategoria>();
    try
    {
        cmdcat = new SqlCommand();
        cmdcat.CommandType = CommandType.StoredProcedure;
        cmdcat.CommandText = "SP_MostrarCategoriaTodo";
        cmdcat.Connection = objconexion.Conectar();
        drcat = cmdcat.ExecuteReader();
        while (drcat.Read())
        {
            BOCategoria objgen = new BOCategoria();
            objgen.codigo =
Convert.ToInt32(drcat["codcat"].ToString());
            objgen.nombre = drcat["nomcat"].ToString();
            objgen.estado =
Convert.ToBoolean(drcat["estcat"].ToString());
            categorias.Add(objgen);
        }
        return categorias;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.ToString());
        return null;
    }
    finally
    {
        objconexion.CerrarConexion();
    }
}

public bool RegistrarCategoria(BOCategoria bc)
{
    try
    {
        cmdcat = new SqlCommand();
        cmdcat.CommandType = CommandType.StoredProcedure;
        cmdcat.CommandText = "SP_RegistrarCategoria";
        cmdcat.Connection = objconexion.Conectar();

        cmdcat.Parameters.AddWithValue("@nombre", bc.nombre);
    }
}

```

```
cmdcat.Parameters.AddWithValue("@estado", bc.estado);

res = cmdcat.ExecuteNonQuery();

if (res == 1)
{
    return true;
}
else
{
    return false;
}

}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
    return false;
}
finally
{
    objconexion.CerrarConexion();
}
}

public bool ActualizarCategoria(BOCategoria bc)
{
    try
    {
        cmdcat = new SqlCommand();
        cmdcat.CommandType = CommandType.StoredProcedure;
        cmdcat.CommandText = "SP_ActualizarCategoria";
        cmdcat.Connection = objconexion.Conectar();

        cmdcat.Parameters.AddWithValue("@codigo", bc.codigo);
        cmdcat.Parameters.AddWithValue("@nombre", bc.nombre);
        cmdcat.Parameters.AddWithValue("@estado", bc.estado);

        res = cmdcat.ExecuteNonQuery();

        if (res == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.ToString());
        return false;
    }
    finally
    {
        objconexion.CerrarConexion();
    }
}

public bool EliminarCategoria(BOCategoria bc)
{

```

```

        try
        {
            cmdcat = new SqlCommand();
            cmdcat.CommandType = CommandType.StoredProcedure;
            cmdcat.CommandText = "SP_EliminarCategoria";
            cmdcat.Connection = objconexion.Conectar();

            cmdcat.Parameters.AddWithValue("@codigo", bc.codigo);

            res = cmdcat.ExecuteNonQuery();

            if (res == 1)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.ToString());
            return false;
        }
        finally
        {
            objconexion.CerrarConexion();
        }
    }
}

```

BALCategoria.cs

```

using pe.com.tiendita.bo;
using pe.com.tiendita.dal;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace pe.com.tiendita.bal
{
    public class BALCategoria
    {
        DALCategoria dalcat = new DALCategoria();

        public List<BOCategoria> MostrarCategoria()
        {
            return dalcat.MostrarCategoria();
        }

        public List<BOCategoria> MostrarCategoriaTodo()
        {
            return dalcat.MostrarCategoriaTodo();
        }

        public bool RegistrarCategoria(BOCategoria bc)
        {

```

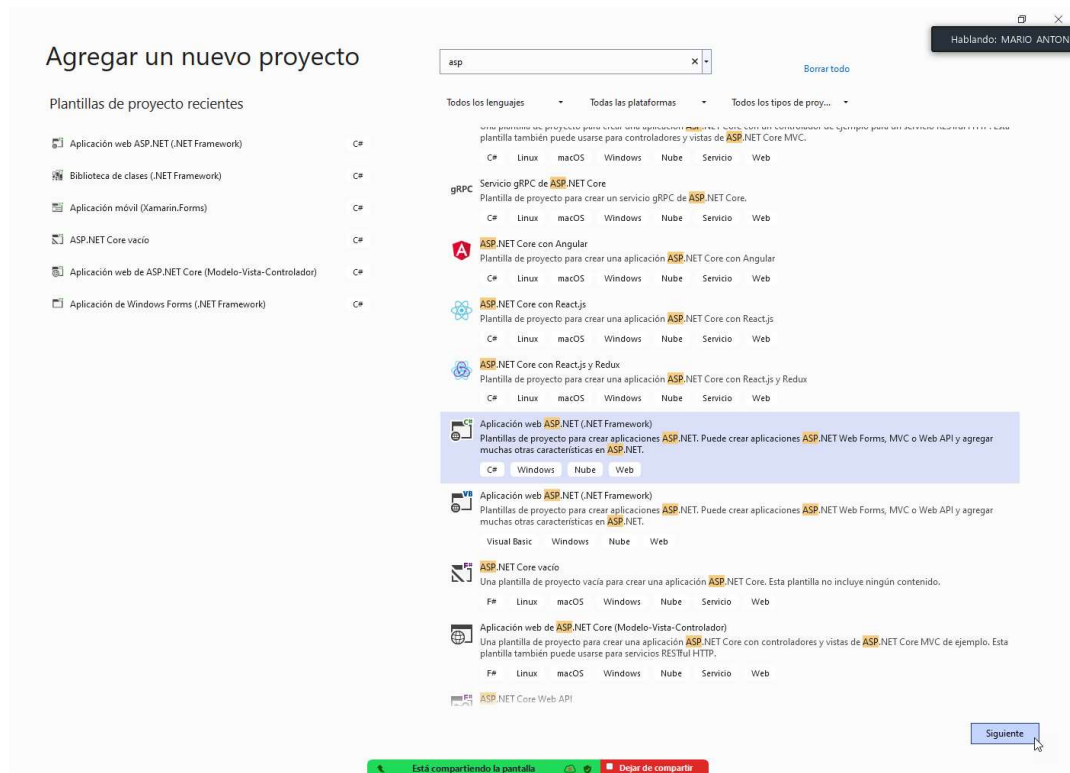
```

        return dalcot.RegistrarCategoria(bc);
    }

    public bool ActualizarCategoria(BOCategoria bc)
    {
        return dalcot.ActualizarCategoria(bc);
    }

    public bool EliminarCategoria(BOCategoria bc)
    {
        return dalcot.EliminarCategoria(bc);
    }
}
}
}

```



Configure su nuevo proyecto

Aplicación web ASP.NET (.NET Framework) C# Windows Nube Web

Nombre del proyecto
pe.com.tiendita.ws

Ubicación
C:\Users\Mario\Desktop\net\laboratorio01\ServicioTiendita

Framework
.NET Framework 4.8

Atrás Crear

Está compartiendo la pantalla Dejar de compartir

Configure su nuevo proyecto

Aplicación web ASP.NET (.NET Framework) C# Windows Nube Web

Nombre del proyecto
pe.com.tiendita.ws

Ubicación
C:\Users\Mario\Desktop

Framework
.NET Framework 4.8

Crear una aplicación web ASP.NET

Vacio
Una plantilla de proyecto vacía para crear aplicaciones ASP.NET. Esta plantilla no tiene contenido.

Web Forms
Una plantilla de proyecto para crear aplicaciones de ASP.NET Web Forms. ASP.NET Web Forms le permite crear sitios web dinámicos con un modelo familiar controlado por eventos para arrastrar y colocar. Una superficie de diseño y cientos de controles y componentes le permiten crear rápidamente sofisticados y eficaces sitios controlados por la interfaz de usuario y con acceso a datos.

MVC
Una plantilla de proyecto para crear aplicaciones ASP.NET MVC. ASP.NET MVC permite compilar aplicaciones mediante la arquitectura de controlador de vista de modelos. ASP.NET MVC incluye muchas características que permiten un desarrollo rápido orientado a pruebas para crear aplicaciones que usan los últimos estándares.

API web
Plantilla de proyecto para crear servicios HTTP REST que pueden llegar a una amplia gama de clientes, como, por ejemplo, exploradores y dispositivos móviles.

Aplicación de página única
Una plantilla de proyectos para crear aplicaciones HTML5 atractivas controladas por JavaScript del lado cliente mediante ASP.NET Web API. Las aplicaciones de una sola página proporcionan una experiencia de usuario atractiva que incluye interacciones del lado cliente mediante HTML5, CSS3 y JavaScript.

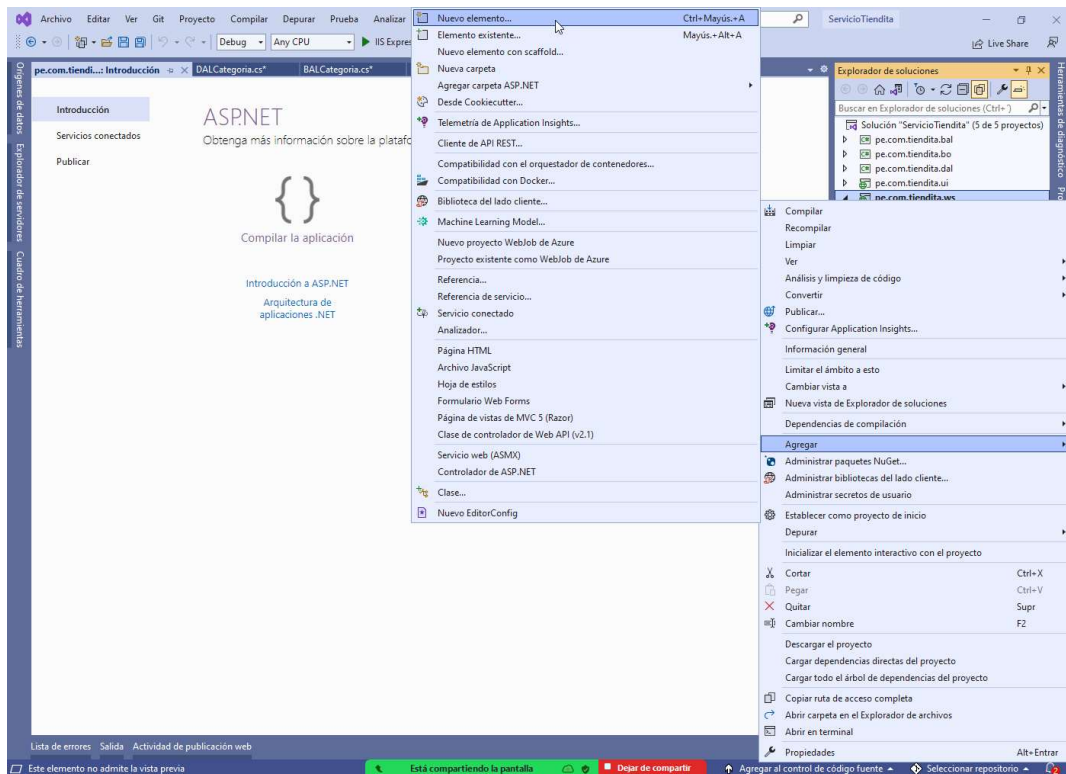
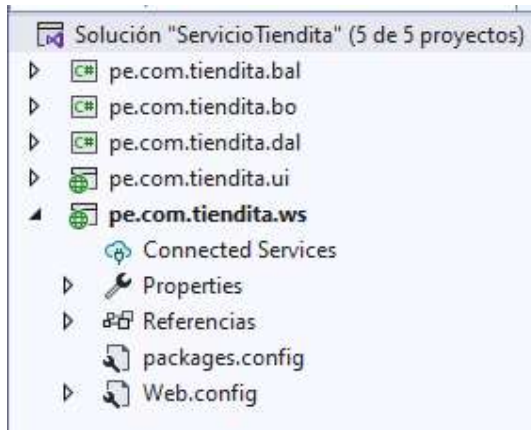
Autenticación
Ninguno

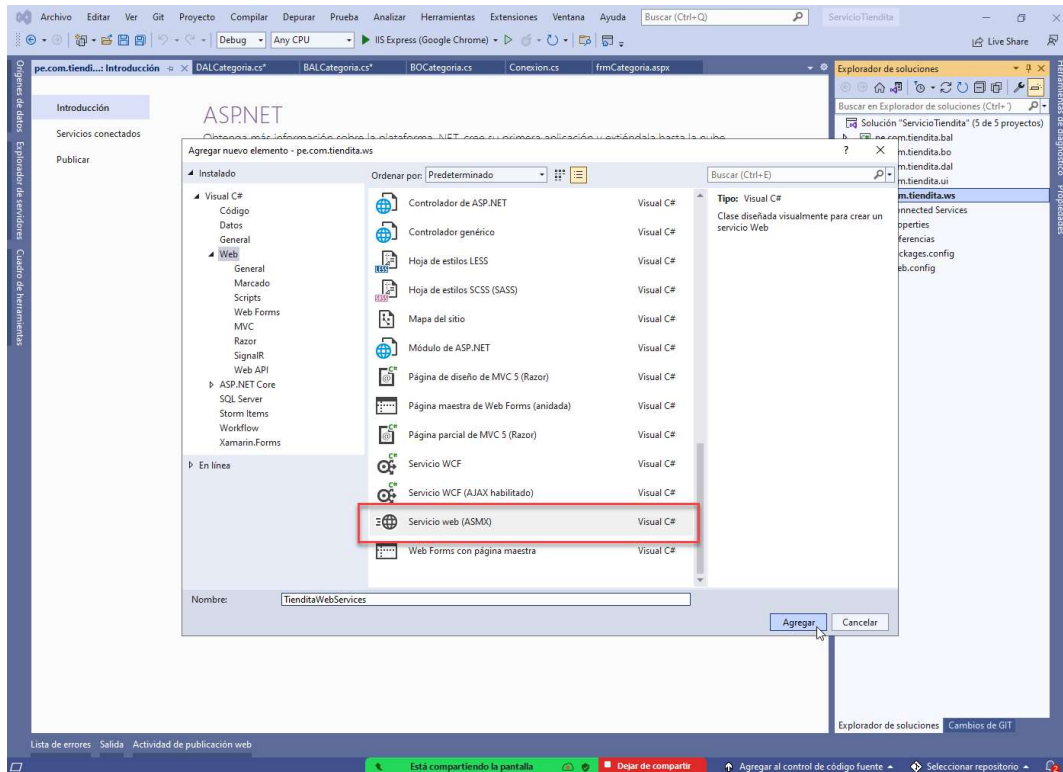
Agregar carpetas y referencias principales
☐ Formularios Web Forms
☐ MVC
☐ API web

Avanzado
☐ Configurar para HTTPS
☐ Compatibilidad con Docker (Requiere Docker Desktop)
☐ Crear también un proyecto para pruebas unitarias
 pe.com.tiendita.ws.Tests

Atrás Crear

Está compartiendo la pantalla Dejar de compartir





TienditaWebServices

```
using pe.com.tiendita.bal;
using pe.com.tiendita.bo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace pe.com.tiendita.ws
{
    /// <summary>
    /// Descripción breve de TienditaWebServices
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // Para permitir que se llame a este servicio web desde un script,
    usando ASP.NET AJAX, quite la marca de comentario de la línea siguiente.
    // [System.Web.Script.Services.ScriptService]
    public class TienditaWebServices : System.Web.Services.WebService
    {
        BOCategoria bocat = new BOCategoria();
        BALCategoria balcat = new BALCategoria();

        [WebMethod]
        public List<BOCategoria> MostrarCategoria()
        {
            List<BOCategoria> categorias =
            balcat.MostrarCategoria().ToList();
            return categorias;
        }
    }
}
```

```
    }

    [WebMethod]
    public List<BOCategoria> MostrarCategoriaTodo()
    {
        List<BOCategoria> categorias =
balcat.MostrarCategoriaTodo().ToList();
        return categorias;
    }

    [WebMethod]
    public bool RegistrarCategoria(BOCategoria bc)
    {
        return balcat.RegistrarCategoria(bc);
    }

    [WebMethod]
    public bool ActualizarCategoria(BOCategoria bc)
    {
        return balcat.ActualizarCategoria(bc);
    }

    [WebMethod]
    public bool EliminarCategoria(BOCategoria bc)
    {
        return balcat.EliminarCategoria(bc);
    }
}
}
```