



Me apasiona la Programacion Movil con Xamarin (C#, XAML), el Desarrollo Web (Frontend y Backend), la Programacion .Net, el Analisis de Datos con Excel, SQL, Power BI, RapidMiner R, Python, y Amo enseñar lo que aprendo cada día.



Santo Domingo, R.D.



Juancito.pena@gmail.com



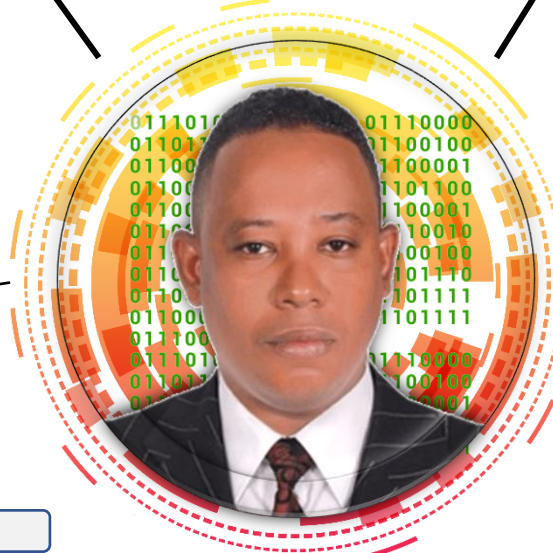
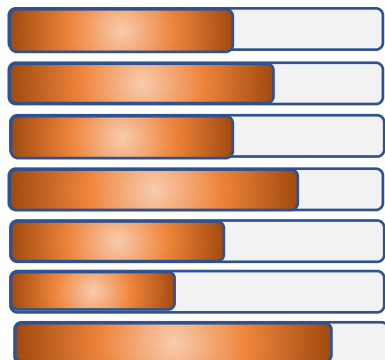
809-767-9290



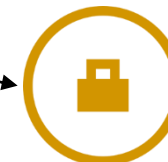
Ingeniero en Sistemas y Computación  
Post-Grado en Ingeniería de Software  
Maestría en Sistemas Mención Gerencial  
**Universidad Dominicana O&M**



- C#, SQL, Crystal Report.
- HTML, CSS, Javascript.
- PHP, MySQL.
- SQL, Power BI, Excel.
- RapidMiner, R, Python.
- XAMARIN, C#, XAML
- Macola, EasySales, MSeller



**Juancito**  
**Peña V.**



Instructor de Tecnología (2019-Hoy)  
**Universidad Dominicana O&M**



Enc. Soporte Tecnológico (2011-Hoy)  
**Cerveceria Vegana S.R.L.**



Consultor de Tecnologías (2019-Hoy)  
**Independiente Freelancer.**



Bloguer Tecnológico (2021)

<https://advisertecnology.com>



UNIVERSIDAD  
DOMINICANA

O&M

SABER  
PENSAR  
TRABAJAR

**Codigo: 506146**

**Materia: Programacion Dispositivos Moviles**

**Creditos: 4**

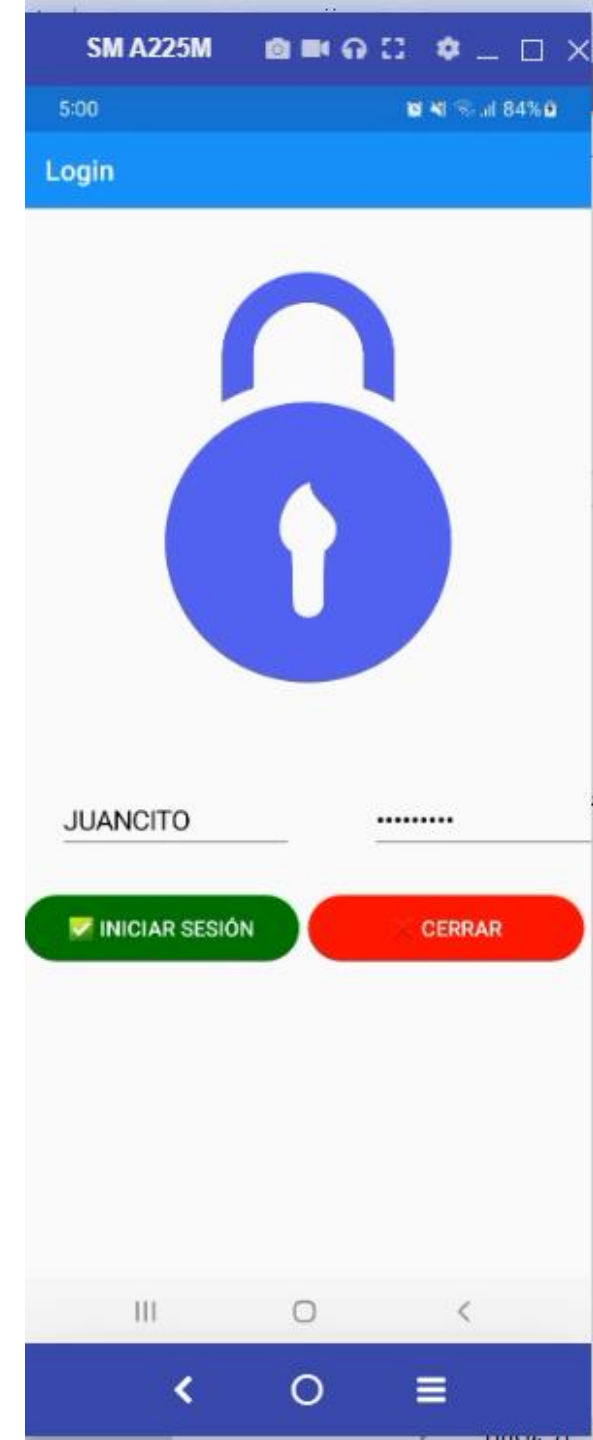
**Pre-Requisito: 506143-P00.**

**Horario: 11:30 PM a 1:00 PM.**

**Día: Martes. Seccion: 0742.**

**Tutor: Ing. Juancito Peña.**

Vamos a crear esta aplicación , la cual es un login con un usuario y un password, el cual me permite loguearme según los permisos, es decir, en una Base de datos, tendremos un usuario, el cual tendra un rol definido, que va desde Administrador, Supervisor y Vendedor, puedes cambiar según tu necesidad, esto esta en la base de datos, la cual te dejo el codigo.



Lo primero que necesitas es descargar e instalar el SQL Server Express, en cualquiera de estas versiones, te recomiendo que lo hagas con la version developer, que si mas adelante tienes otros proyectos o decides trabajar con esto, pues aquí lo tendrías instalado.

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>



### Developer

SQL Server 2022 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.

[Download now](#)



### Express

SQL Server 2022 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

[Download now](#)

Para poder administrar la base de datos de forma grafica tendrás que instalar el administrador de base de datos de SQL Server, el cual es el SQL Management Studio.

<https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

deploy, monitor, and upgrade the data-tier components used by your applications and build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are - on your local computer or in the cloud.

## Download SSMS

[Free Download for SQL Server Management Studio \(SSMS\) 19.0.2](#)

SSMS 19.0.2 is the latest general availability (GA) version. If you have a *preview* version of SSMS 19 installed, you should uninstall it before installing SSMS 19.0.2. If you have SSMS 19.x installed, installing SSMS 19.0.2 upgrades it to 19.0.2.

- Release number: 19.0.2
- Build number: 19.0.20209.0
- Release date: March 13, 2023

By using SQL Server Management Studio, you agree to its [license terms](#) and [privacy statement](#). If you have comments or suggestions or want to report issues, the best way to contact the SSMS team is at [SQL Server user feedback](#).

The SSMS 19.x installation doesn't upgrade or replace SSMS versions 18.x or earlier. SSMS 19.x installs alongside previous versions, so both versions are available for use. However,

**Aquí en este video puedes ver Cómo se instala y configuran  
ambas version de SQL SERVER paso a paso**



[https://www.youtube.com/watch?v=mA1qoWdNCOE&ab\\_channel=SergioAlejandroCampos-EXCELeINFO](https://www.youtube.com/watch?v=mA1qoWdNCOE&ab_channel=SergioAlejandroCampos-EXCELeINFO)

Una vez has instalado tu programa de SQL Server, ahora abres una nueva consulta, en la cual vas a copiar y pegar este código, el cual es la base de datos, la tabla, la inserción de datos, y algunas consultas que te ayudaran a ver esos registros.

```
--crear una base de datos SECCION-PDM:  
create database LOGIN_PDM  
GO
```

```
--USAR ESA BASE DE DATOS:  
USE LOGIN_PDM  
GO
```

```
-- Creación de tabla de roles  
CREATE TABLE roles (  
    id_rol INT PRIMARY KEY,  
    nombre_rol VARCHAR(50) NOT NULL  
);
```

```
select * from roles
```

-- Creación de tabla de usuarios

```
CREATE TABLE usuarios (  
    id_usuario INT PRIMARY KEY,  
    nombre_usuario VARCHAR(50) NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    id_rol INT NOT NULL,  
    FOREIGN KEY (id_rol) REFERENCES roles(id_rol)  
);
```

```
select * from usuarios
```

-- Insertando registros en tabla de roles

```
INSERT INTO roles (id_rol, nombre_rol) VALUES (1, 'administrador');  
INSERT INTO roles (id_rol, nombre_rol) VALUES (2, 'supervisor');  
INSERT INTO roles (id_rol, nombre_rol) VALUES (3, 'vendedor');
```

-- Insertando registros en tabla de usuarios

```
INSERT INTO usuarios (id_usuario, nombre_usuario, password, id_rol) VALUES (1, 'JUANCITO', 'ADMIN@123', 1);  
INSERT INTO usuarios (id_usuario, nombre_usuario, password, id_rol) VALUES (2, 'DARIEL', 'DARIEL@123', 2);  
INSERT INTO usuarios (id_usuario, nombre_usuario, password, id_rol) VALUES (3, 'DANIELA', 'DANIELA@123', 3);  
INSERT INTO usuarios (id_usuario, nombre_usuario, password, id_rol) VALUES (4, 'MARIA', 'MARIA@123', 1);  
INSERT INTO usuarios (id_usuario, nombre_usuario, password, id_rol) VALUES (5, 'YENNEFER ', 'YENNEFER@123', 2);
```



-- Consultas básicas

-- Seleccionar todos los registros de la tabla de roles

```
SELECT * FROM roles;
```

-- Seleccionar todos los registros de la tabla de usuarios

```
SELECT * FROM usuarios;
```

-- Seleccionar un registro específico de la tabla de roles

```
SELECT * FROM roles WHERE id_rol = 2;
```

-- Seleccionar un registro específico de la tabla de usuarios

```
SELECT * FROM usuarios WHERE id_usuario = 4;
```

-- Consultas avanzadas

-- Seleccionar todos los usuarios y sus roles

```
SELECT usuarios.*, roles.nombre_rol
```

```
FROM usuarios
```

```
INNER JOIN roles ON usuarios.id_rol = roles.id_rol;
```

-- Seleccionar los usuarios que tengan un rol específico

```
SELECT * FROM usuarios WHERE id_rol = 3;
```

```
-- Contar la cantidad de usuarios por cada rol
SELECT roles.nombre_rol, COUNT(usuarios.id_usuario) AS cantidad_usuarios
FROM usuarios
INNER JOIN roles ON usuarios.id_rol = roles.id_rol
GROUP BY roles.nombre_rol;

-- Actualizar un registro en la tabla de roles
UPDATE roles SET nombre_rol = 'administrador principal' WHERE id_rol = 1;

-- Eliminar un registro en la tabla de usuarios
DELETE FROM usuarios WHERE id_usuario = 8;

SELECT * FROM usuarios;
SELECT * FROM roles;
```

-- Creación de la tabla usuario

```
CREATE TABLE usuario (  
    id_usuario INT NOT NULL,  
    nombre_user VARCHAR(20) NOT NULL,  
    telefono VARCHAR(30) NOT NULL,  
    email VARCHAR(30) NOT NULL  
);
```

```
SELECT * FROM usuario
```

-- Insertar datos en la tabla usuario

```
INSERT INTO usuario VALUES (1, 'JUANCITO', '809-555-8789', 'JUANCITO@GMAIL.COM');
```

-- Creación de la tabla Articulos

```
CREATE TABLE Articulos (  
    IDArticulo INT PRIMARY KEY,  
    Nombre NVARCHAR(100),  
    Precio DECIMAL(10, 2)  
);
```

-- Insertar datos en la tabla Articulos

```
INSERT INTO Articulos (IDArticulo, Nombre, Precio)
VALUES
    (1, 'Laptop Legion', 850.50),
    (2, 'Mouse Gamer', 30.25),
    (3, 'Monitor Gamer 32', 250.75),
    (4, 'Bocinas', 12.90),
    (5, 'UPS de 1200 kw', 120.00);
```

-- Creación de la tabla Ventas

```
CREATE TABLE Ventas (
    IDVenta INT IDENTITY(1, 1) PRIMARY KEY,
    IDArticulo INT,
    Cantidad INT,
    Precio DECIMAL(18, 2)
);
```

-- Consulta de la tabla Ventas

```
SELECT * FROM Ventas;
```

-- Insertar 10 artículos tecnológicos en la tabla Articulos

```
INSERT INTO Articulos (IDArticulo, Nombre, Precio)
VALUES
```

```
(6, 'Smartphone', 500.00),
(7, 'Laptop Legion', 1000.00),
(8, 'Tablet', 300.00),
(9, 'Smartwatch', 150.00),
(10, 'Gafas de Realidad Virtual', 200.00),
(11, 'Auriculares Inalámbricos', 100.00),
(12, 'Altavoz Inteligente', 80.00),
(13, 'Cámara de Acción', 250.00),
(14, 'Monitor Curvo', 300.00),
(15, 'Consola de Videojuegos', 400.00);
```


-- Consulta de la tabla Articulos

```
SELECT * FROM Articulos;
```

# Crear un proyecto


## Plantillas de proyecto recientes


-  Aplicación móvil (Xamarin.Forms) C#
-  Aplicación de Windows Forms (.NET Framework) Visual Basic
-  Aplicación de Windows Forms (.NET Framework) C#
-  Aplicación de Windows Forms C#
-  Integration Services Project
-  Aplicación de Crystal Reports Visual Basic


Buscar plantillas (Alt+S) 


Borrar todo


C# Todas las plataformas Móvil


 **Aplicación móvil (Xamarin.Forms)**  
Plantilla de varios proyectos para compilar aplicaciones para iOS y Android con Xamarin y Xamarin.Forms.  
C# Android iOS Windows Móvil

 **Aplicación de Android (Xamarin)**  
Plantillas de proyecto para crear aplicaciones para teléfono y tableta de Android con Xamarin.  
C# Android Móvil

 **Aplicación de iOS (Xamarin)**  
Plantillas de proyecto para crear aplicaciones de iOS para iPhone y iPad con Xamarin.  
C# iOS Móvil

 **Aplicación Android Wear (Xamarin)**  
Proyecto para crear una aplicación Android Wear con Xamarin.  
C# Android Móvil

 **Aplicación watchOS (Xamarin)**  
Proyecto para crear una aplicación de watchOS con Xamarin.  
C# iOS Móvil

 **Biblioteca de clases de Android (Xamarin)**  
Proyecto de biblioteca de clases de Xamarin.Android.  
C# Android Móvil

Siguiente

# Nueva aplicación móvil

Seleccionar una plantilla para la aplicación

## Control flotante

Una aplicación con un menú lateral que se puede contraer en pequeñas pantallas.

## Con pestañas

Una aplicación que usa pestañas para navegar entre las secciones.

## En blanco

Una aplicación vacía con una sola pantalla inicial.



Tengo previsto desarrollar:

- ☒ Android
- ☒ iOS
- ☐ Windows (UWP) 

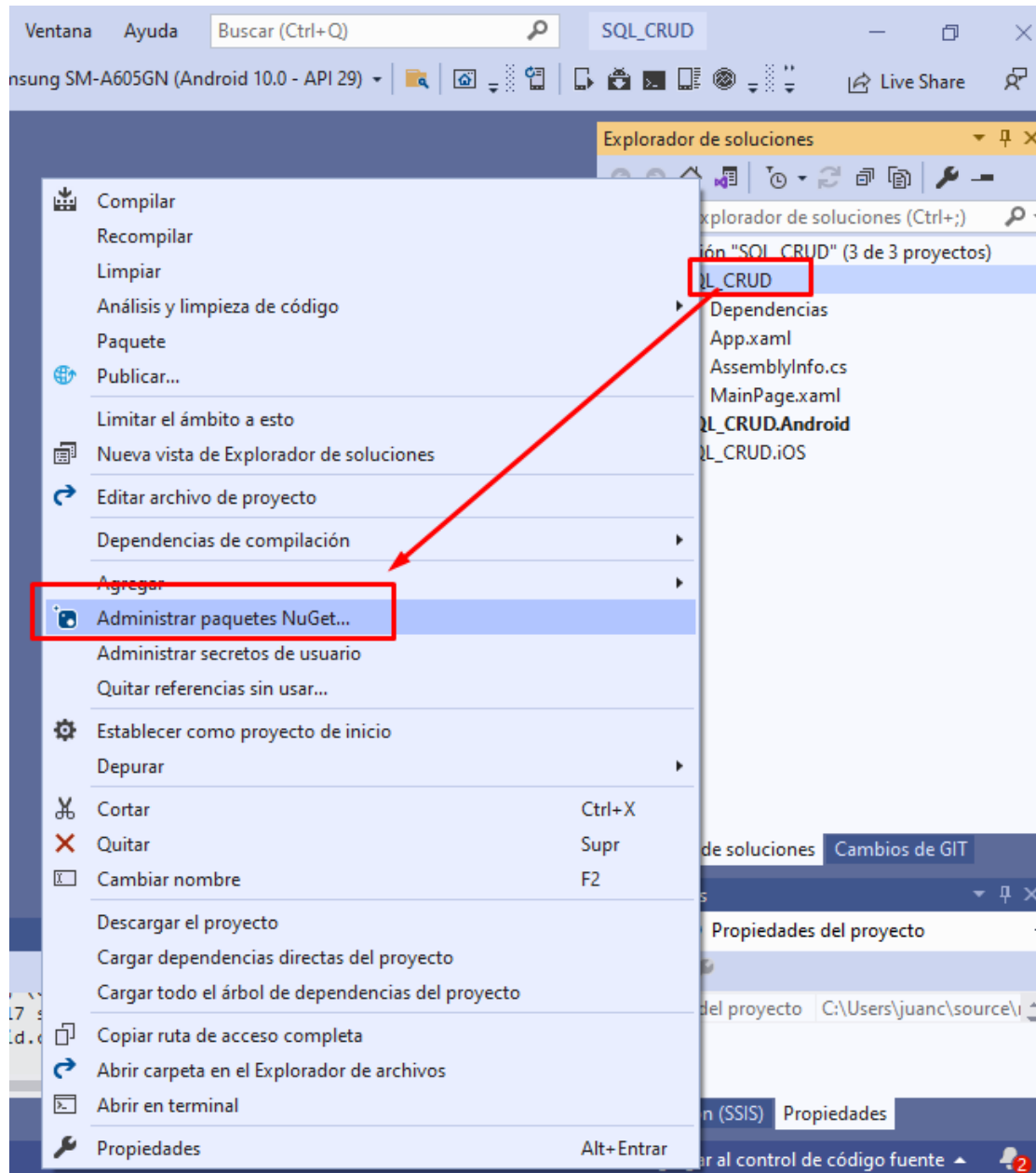
Atrás

Crear

Microsoft Visual Studio

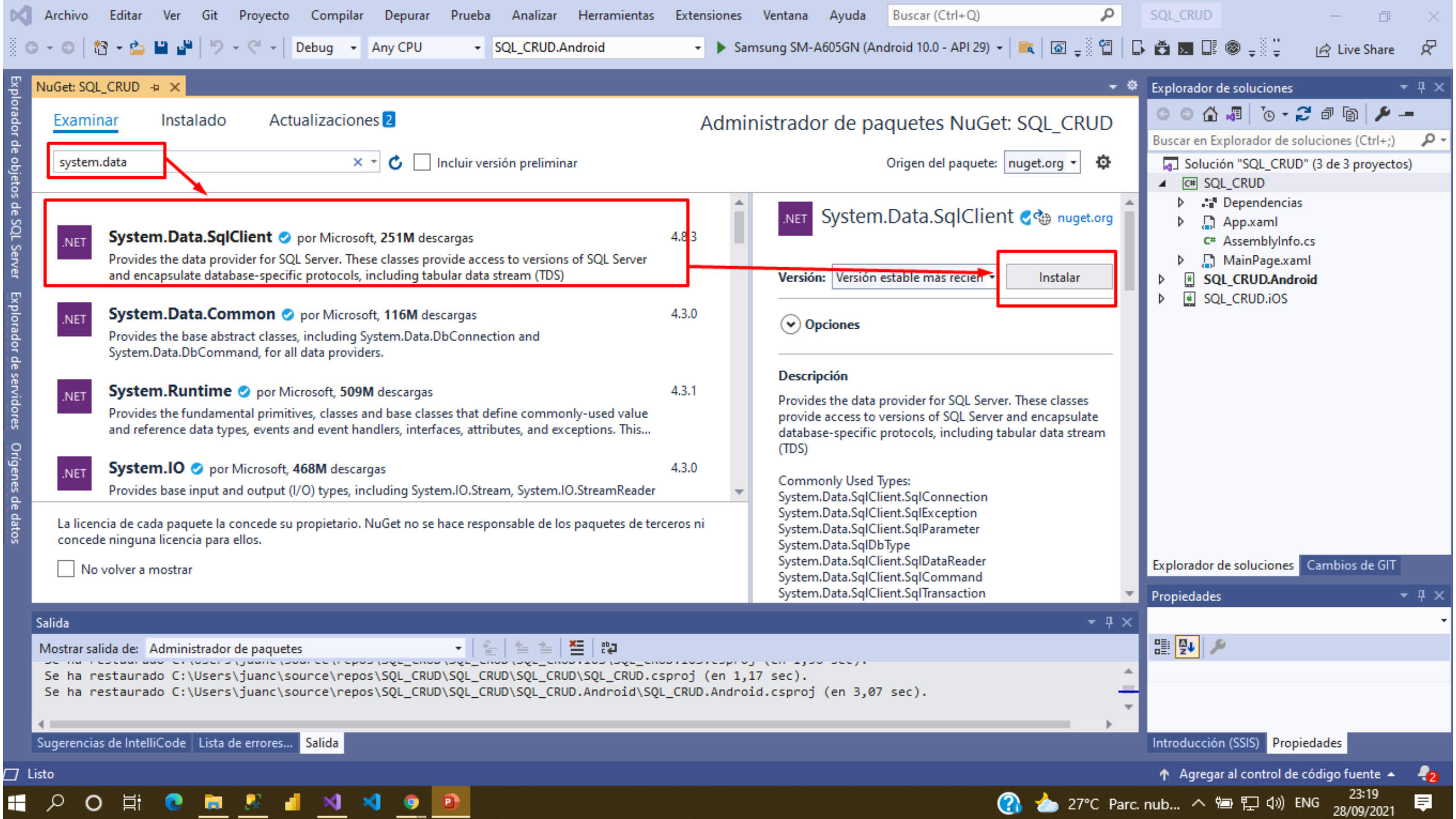
Creando el proyecto...

Cancelar

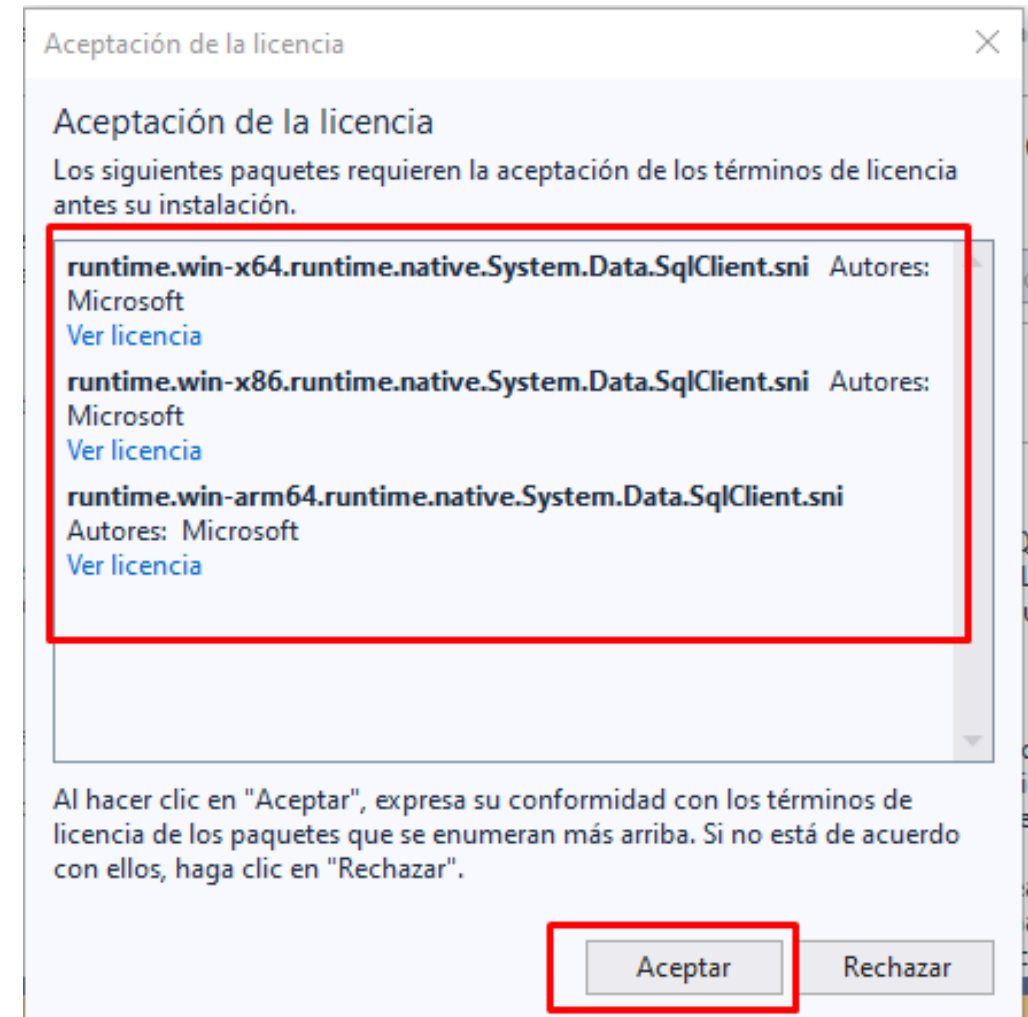
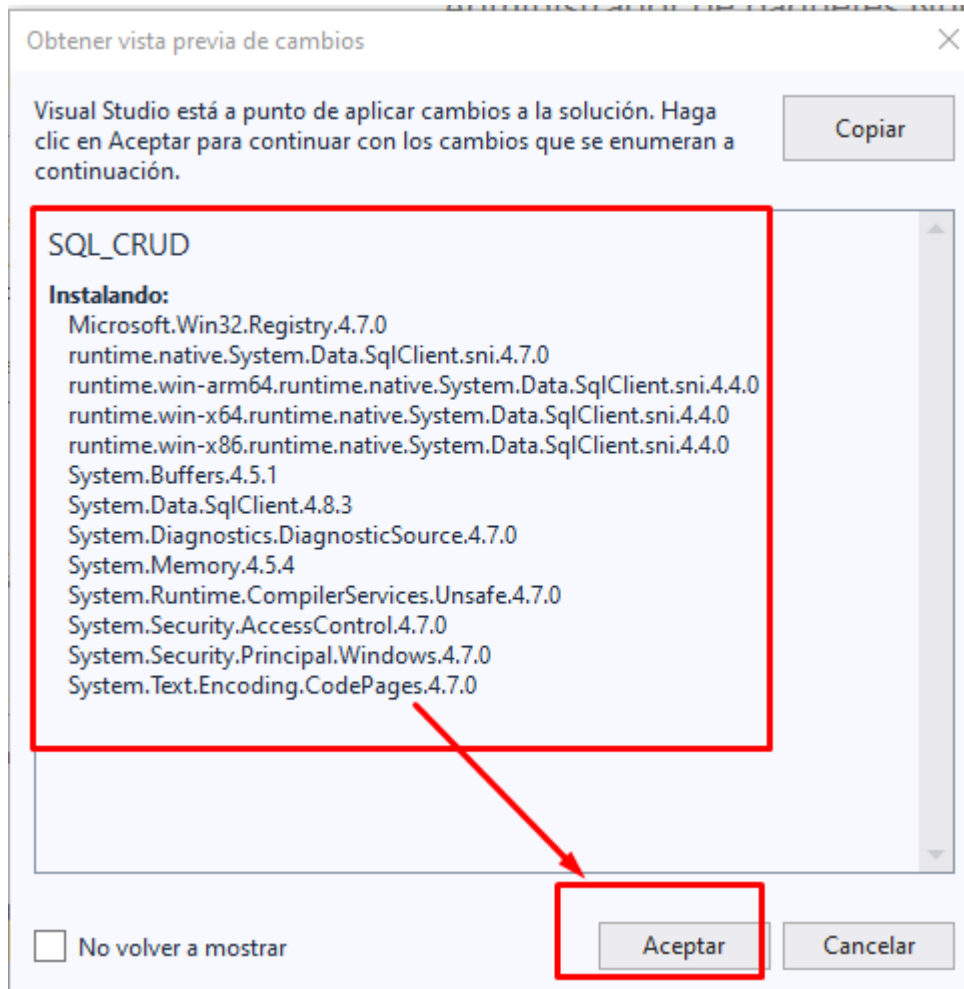


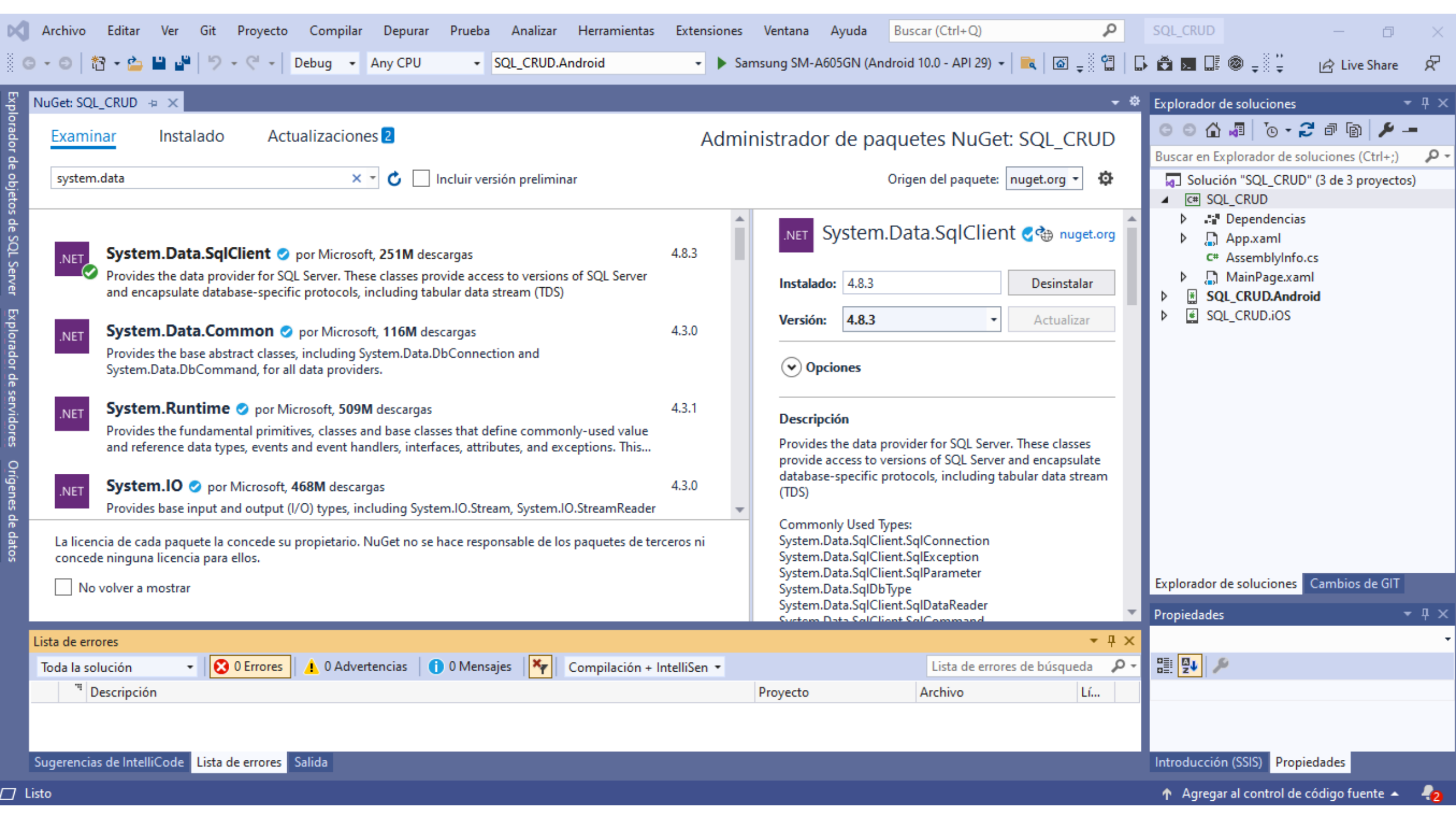
Vamos a instalar el plugin que nos permitira configurar nuestra cadena de conexión con el servidor y nuestra base de datos, la cual será la que validará nuestros roles en la aplicación.





# Aceptamos en ambos casos.





Aqui cuando se instala veremos el check en verde.

Cuadro de herramientas Cuadro de herramientas de SSIS Esquema del documento Orígenes de datos





NuGet: SECCION\_PDM curd\_User.xaml.cs curd\_User.xaml MainPage.xaml.cs MainPage.xaml


Examinar Instalado Actualizaciones 2

Buscar (Ctrl+L)  ☒ Incluir versión preliminar

Administrador de paquetes NuGet: SECCION\_PDM

Origen del paquete:

	<b>NETStandard.Library</b> por Microsoft	2.0.3
A set of standard .NET APIs that are prescribed to be used and supported together. 18a36291e48808fa7ef2d00a764ceb1ec95645a5		
	<b>System.Data.SqlClient</b> por Microsoft	4.8.5
Provides the data provider for SQL Server. These classes provide access to versions of SQL Server and encapsulate database-specific protocols, including tabular data stream (TDS)		
	<b>Xamarin.Essentials</b> por Microsoft	1.6.1
Xamarin.Essentials: a kit of essential API's for your apps		
	<b>Xamarin.Forms</b> por Microsoft	5.0.0.2012
Build native UIs for iOS, Android, UWP, macOS, Tizen and many more from a single, shared C# codebase		

**System.Data.SqlClient**  [nuget.org](#)

Instalado:

Versión:

☒ Opciones

**Descripción**

Provides the data provider for SQL Server. These classes provide access to versions of SQL Server and encapsulate database-specific protocols, including tabular data stream (TDS)

Commonly Used Types:

- System.Data.SqlClient.SqlConnection
- System.Data.SqlClient.SqlException
- System.Data.SqlClient.SqlParameter
- System.Data.SqlDbType
- System.Data.SqlClient.SqlDataReader
- System.Data.SqlClient.SqlCommand
- System.Data.SqlClient.SqlTransaction
- System.Data.SqlClient.SqlParameterCollection
- System.Data.SqlClient.SqlClientFactory

When using NuGet 3.x this package requires at least version 3.4.

Versión: 4.8.5

Autores: Microsoft

Licencia: MIT

Ubicamos nuestra IP de nuestro que servira como un Servidor para nuestra base de datos. Para ellos abrimos la Ventana de ejecutar, presionando las Teclas (**Windows + R**), y escribimos dentro cmd y presionamos enter. Una vez hacemos nos sale una pantalla Negra en la cual vamos a validar nuestra ip con el comando **ipconfig**, y nos mostrara la IP que tiene, en mi caso fue **10.0.0.4**, en el caso de ustedes seria la que le corresponde a su proveedor ISP.

```
Seleccionar C:\WINDOWS\system32\cmd.exe

Adaptador de LAN inalámbrica Conexión de área local* 10:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . :
Dirección IPv6 . . . . . : fd00:9acd:1947:9900:48de:8a37:c6a4:4c2b
Dirección IPv6 temporal. . . . . : fd00:9acd:1947:9900:9539:95fa:7d41:9a1f
Vínculo: dirección IPv6 local. . . : fe80::48de:8a37:c6a4:4c2b%23
Dirección IPv4. . . . . : 10.0.0.4
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 10.0.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

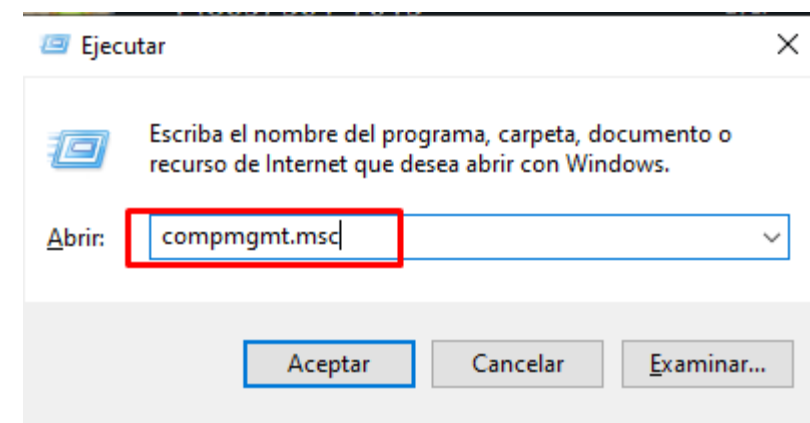
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet vEthernet (Default Switch):

Sufijo DNS específico para la conexión. . :
Vínculo: dirección IPv6 local. . . : fe80::684e:ad8e:8ee0:30f1%52
Dirección IPv4. . . . . : 172.29.144.1
Máscara de subred . . . . . : 255.255.240.0
Puerta de enlace predeterminada . . . . . :

C:\Users\juanc>
```

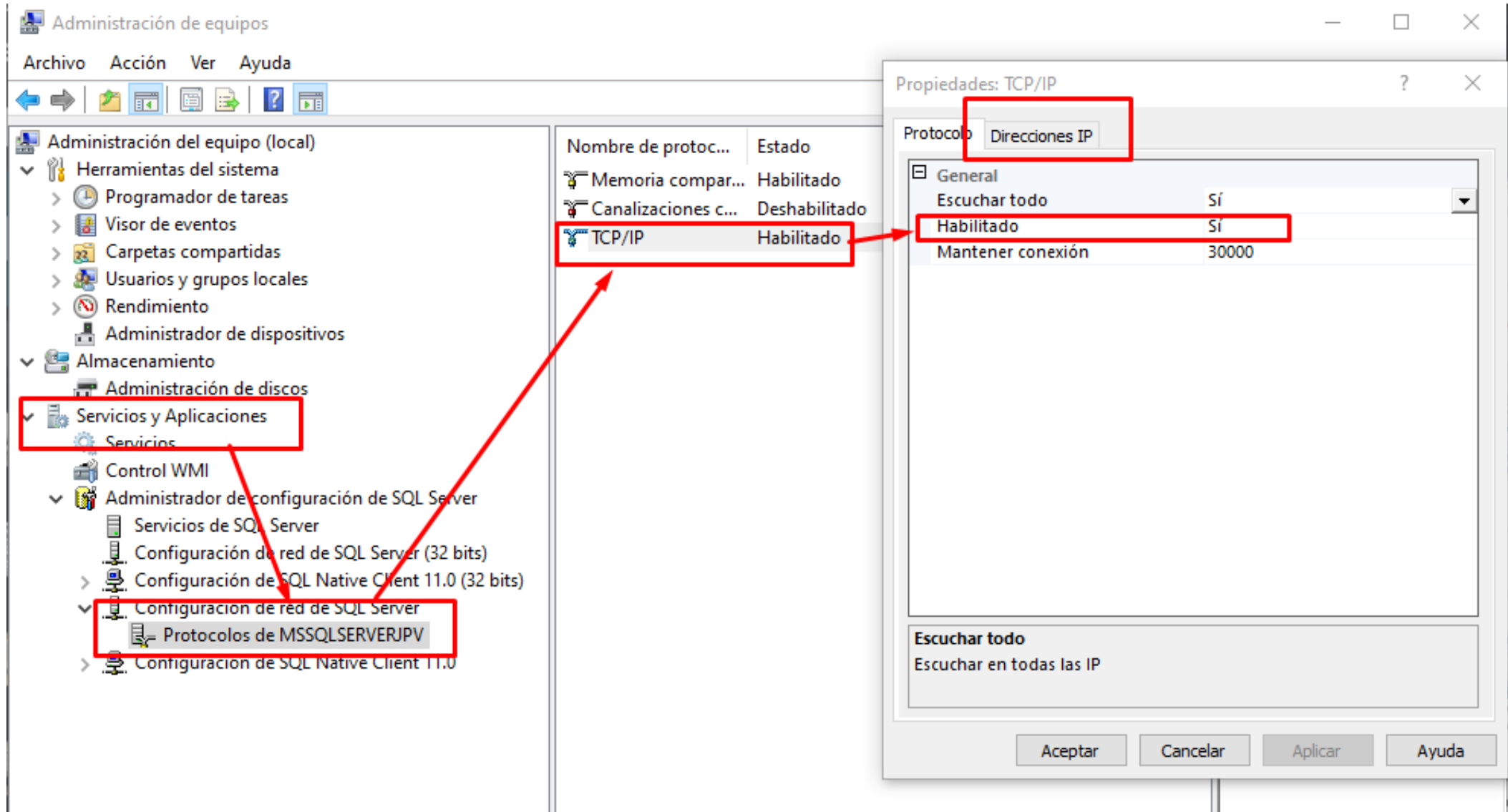
**compmgmt.msc**



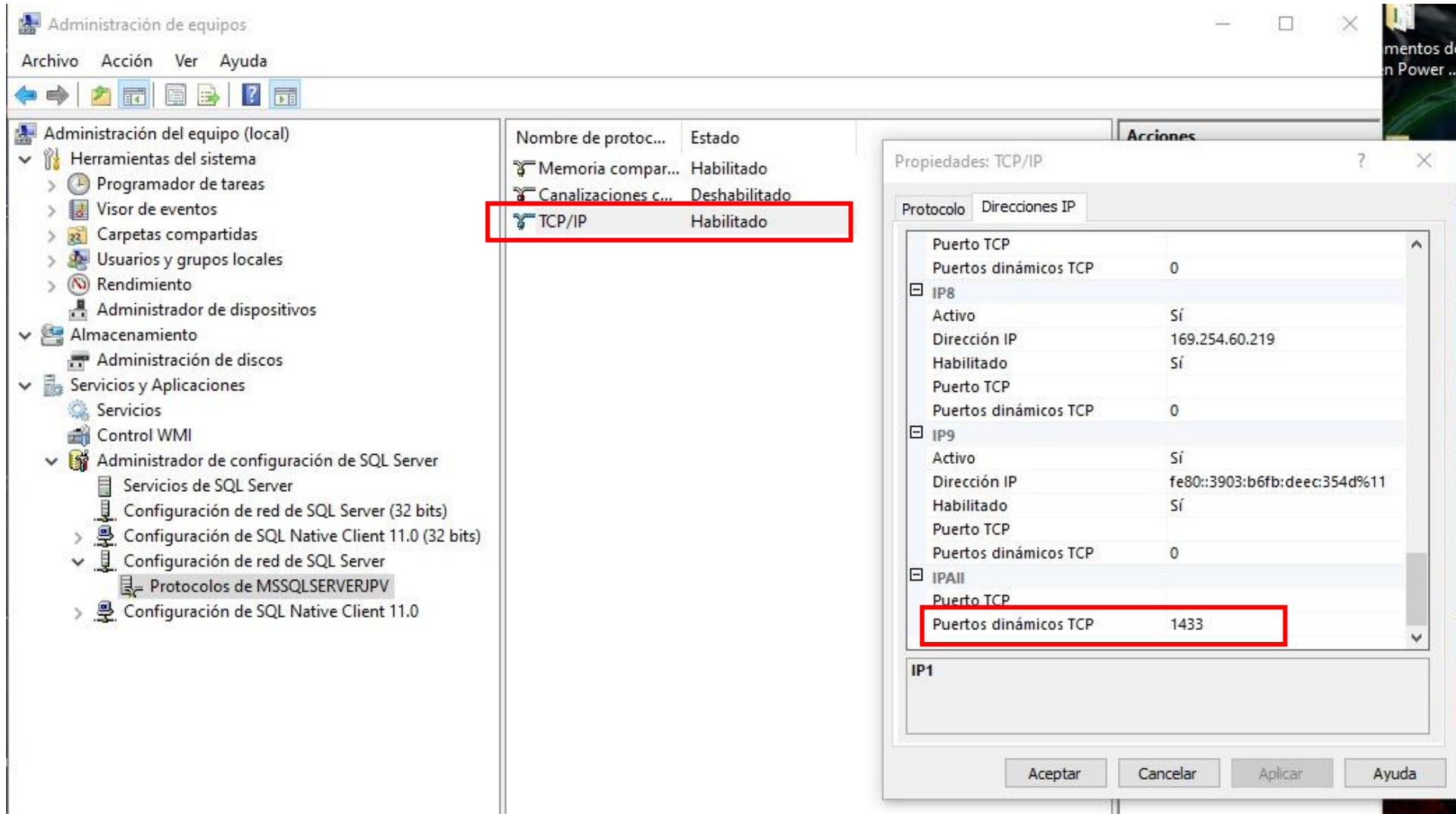
Luego hacemos lo mismo, y escribimos lo siguiente: **compmgmt.msc**, y vamos a realizar lo siguiente.



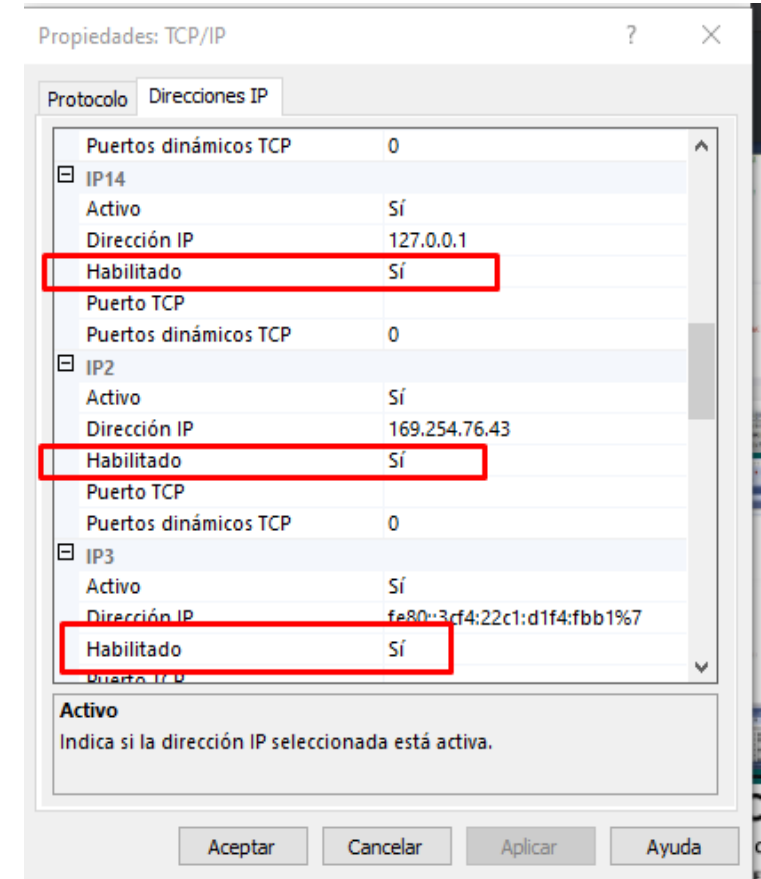
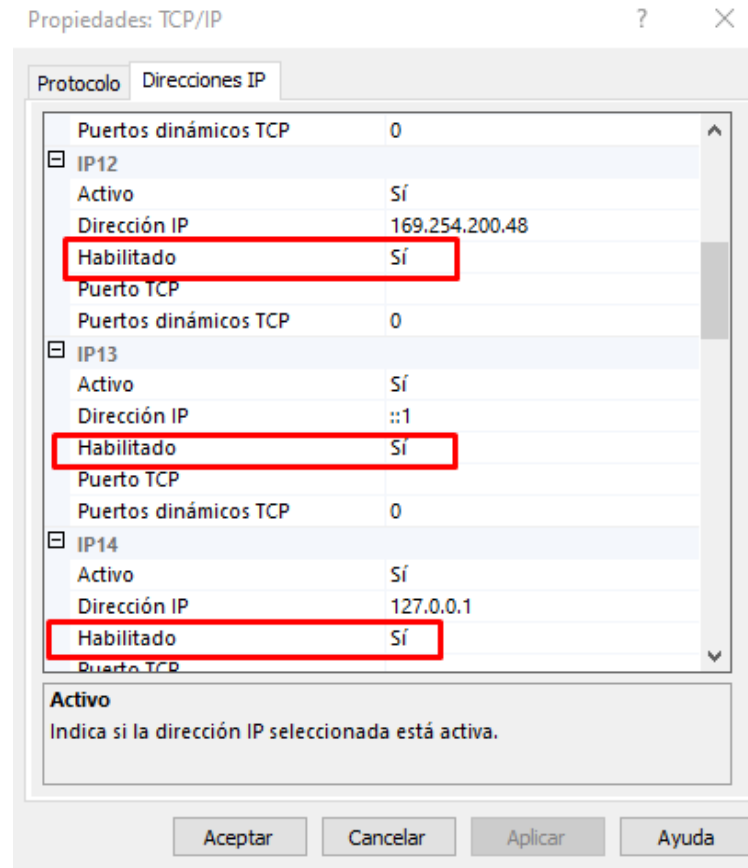
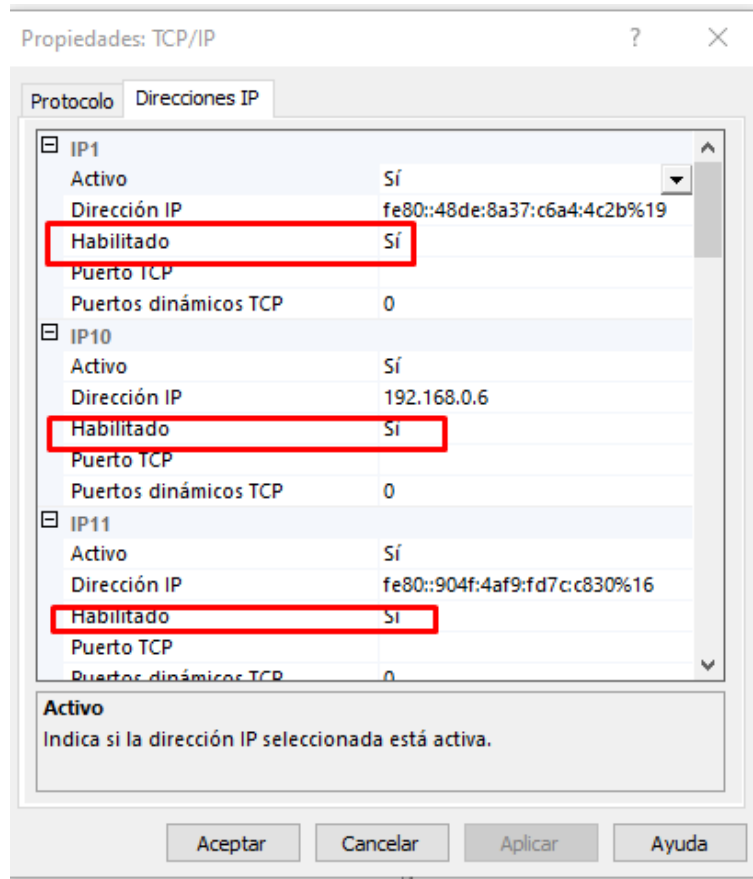
**Necesitamos Habilitar el protocolo TCP/IP sobre la dirección IP del server:** Abrir en el servidor el programa **Administrador de configuración de SQL Server (Sql Server Configuration Manager)**, desde el menú de aplicaciones de SQL Server, o buscando el programa en el buscador del server.



Una vez abierto, en el menú de la izquierda, en '*Configuración de red de SQL Server*', Seleccionar '*Protocolos de [Nombre de la instancia]*', y hacer doble click sobre el Nombre de protocolo TCP/IP para acceder a sus propiedades y seleccionar '*Si*' en '*Habilitado*', y en la pestaña de Direcciones IP, seleccionar también '*Si*' en '*Habilitado*' de la IP del server, este caso 192.168.1.116, y asegurarse de que el puerto es correcto (en este caso 1433 ya estaba bien porque es el puerto por defecto, pero si no hay que modificarlo)



Verifico que todos los puertos ip esten habilitados, sino los habilito.





y asegurarse de que el puerto es correcto (en este caso 1433 ya estaba bien porque es el puerto por defecto, pero si no hay que modificarlo)

Propiedades: TCP/IP

Protocolo Direcciones IP

IP4	
Activo	Sí
Dirección IP	169.254.251.177
Habilitado	Sí
Puerto TCP	
Puertos dinámicos TCP	0
IP5	
Activo	Sí
Dirección IP	fe80::b07b:2690:7371:6bf%14
Habilitado	Sí
Puerto TCP	
Puertos dinámicos TCP	0
IP6	
Activo	Sí
Dirección IP	169.254.6.191
Habilitado	Sí
Puerto TCP	
Puertos dinámicos TCP	0

**Activo**  
Indica si la dirección IP seleccionada está activa.

Aceptar Cancelar Aplicar Ayuda

Propiedades: TCP/IP

Protocolo Direcciones IP

Activo		Sí
Dirección IP		fe80::b8da:c68c:49b:3cdb%15
Habilitado		Sí
Puerto TCP		
Puertos dinámicos TCP		0
IP8		
Activo		Sí
Dirección IP		169.254.60.219
Habilitado		Sí
Puerto TCP		
Puertos dinámicos TCP		0
IP9		
Activo		Sí
Dirección IP		fe80::3903:b6fb:deec:354d%11
Habilitado		Sí
Puerto TCP		
Puertos dinámicos TCP		0
IPAll		
Activo		

**Activo**  
Indica si la dirección IP seleccionada está activa.

Aceptar Cancelar Aplicar Ayuda

Propiedades: TCP/IP

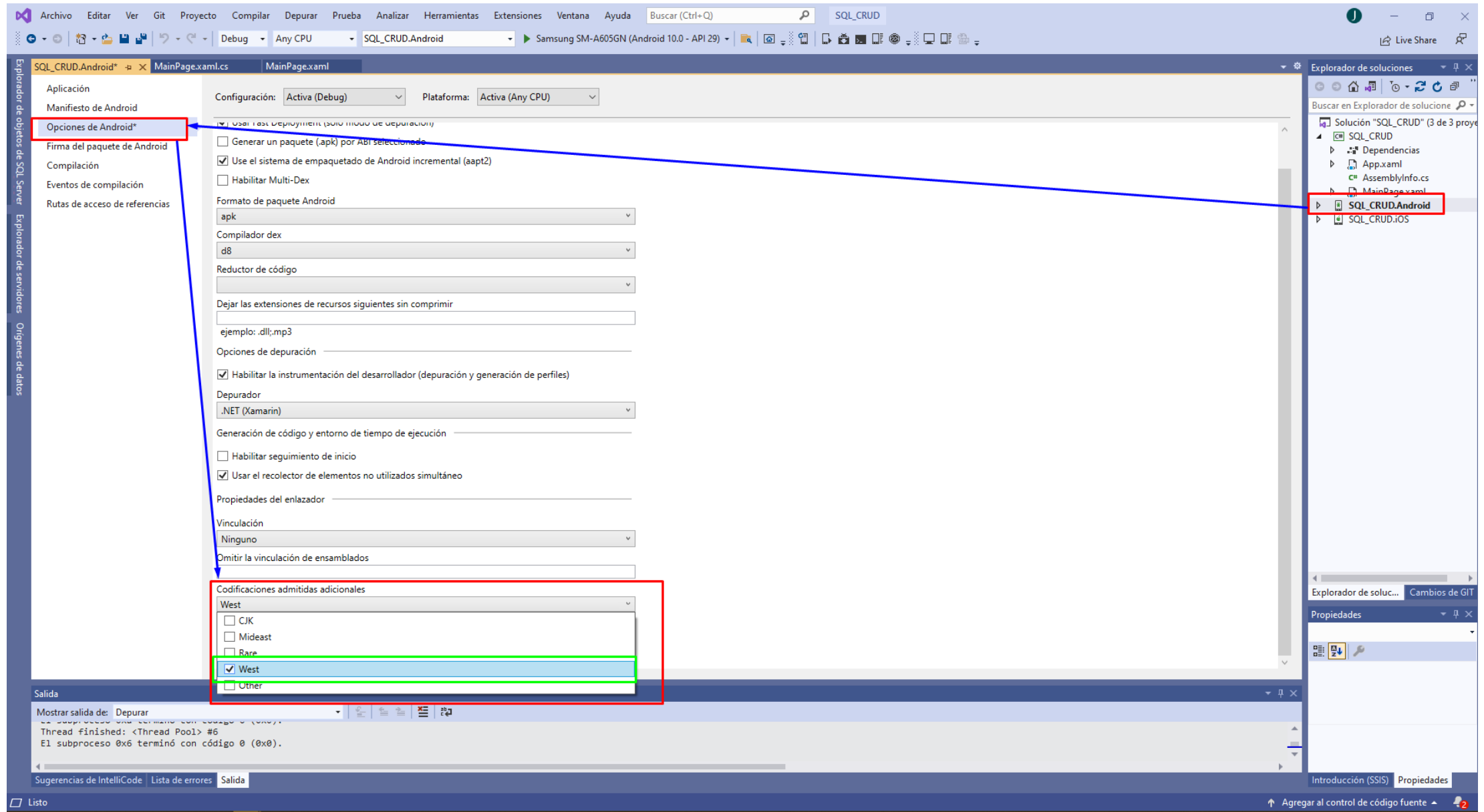
Protocolo Direcciones IP

Puerto TCP		
Puertos dinámicos TCP		0
IP8		
Activo		Sí
Dirección IP		169.254.60.219
Habilitado		Sí
Puerto TCP		
Puertos dinámicos TCP		0
IP9		
Activo		Sí
Dirección IP		fe80::3903:b6fb:deec:354d%11
Habilitado		Sí
Puerto TCP		
Puertos dinámicos TCP		0
IPAll		
Puerto TCP		
Puertos dinámicos TCP		1433

**Activo**  
Indica si la dirección IP seleccionada está activa.

Aceptar Cancelar Aplicar Ayuda

El siguiente Paso es ir al proyecto de Android, clic derecho, propiedades, ir a opciones de android, ir a codificaciones admitidas adicionales, marcamos West.

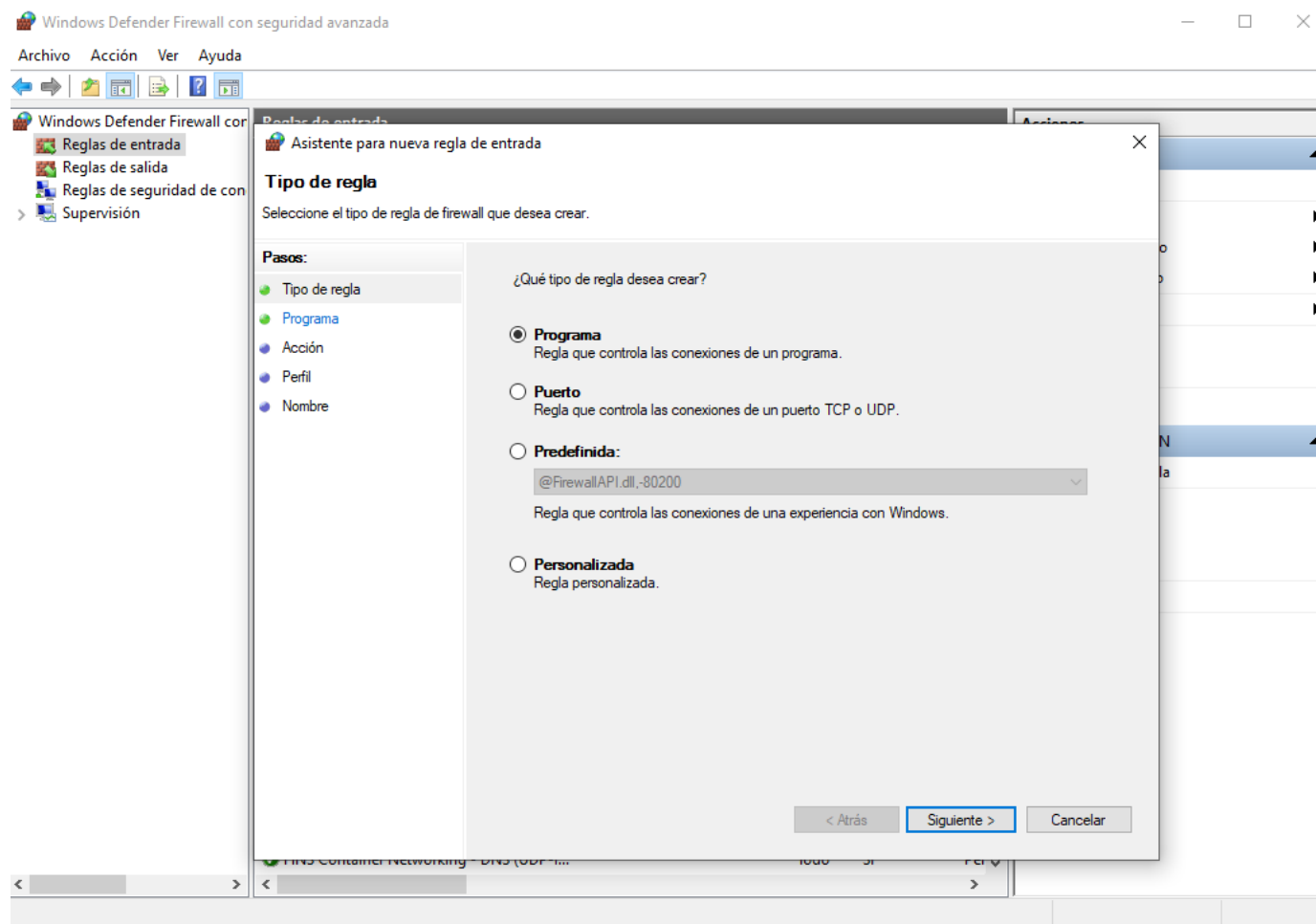


En el contexto de las opciones de codificación admitidas adicionales en **Android Studio**, "**West**" se refiere a la codificación occidental (también conocida como codificación **ISO-8859-1**), que es un estándar para representar caracteres en los idiomas de Europa occidental. Esta codificación cubre la mayoría de los idiomas europeos, incluyendo el **inglés, francés, español, alemán, italiano, portugués**, entre otros.

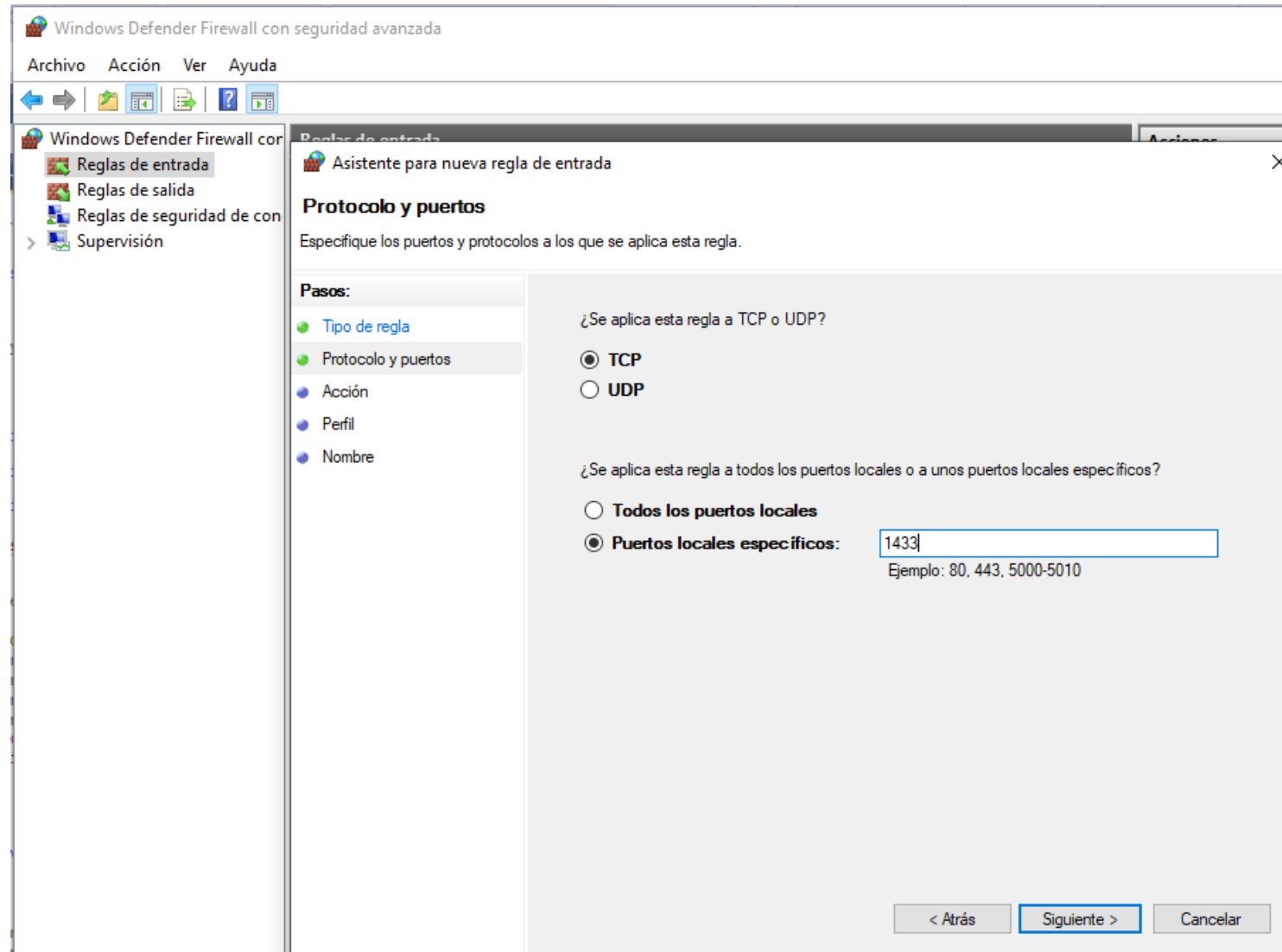
# Abrir el puerto TCP desde el Firewall de Windows Server

El siguiente paso es abrir la aplicación Firewall de Windows, desde el panel de control (Panel de control\Sistema y seguridad\Firewall de Windows), o buscando Firewall en el buscador de aplicaciones.

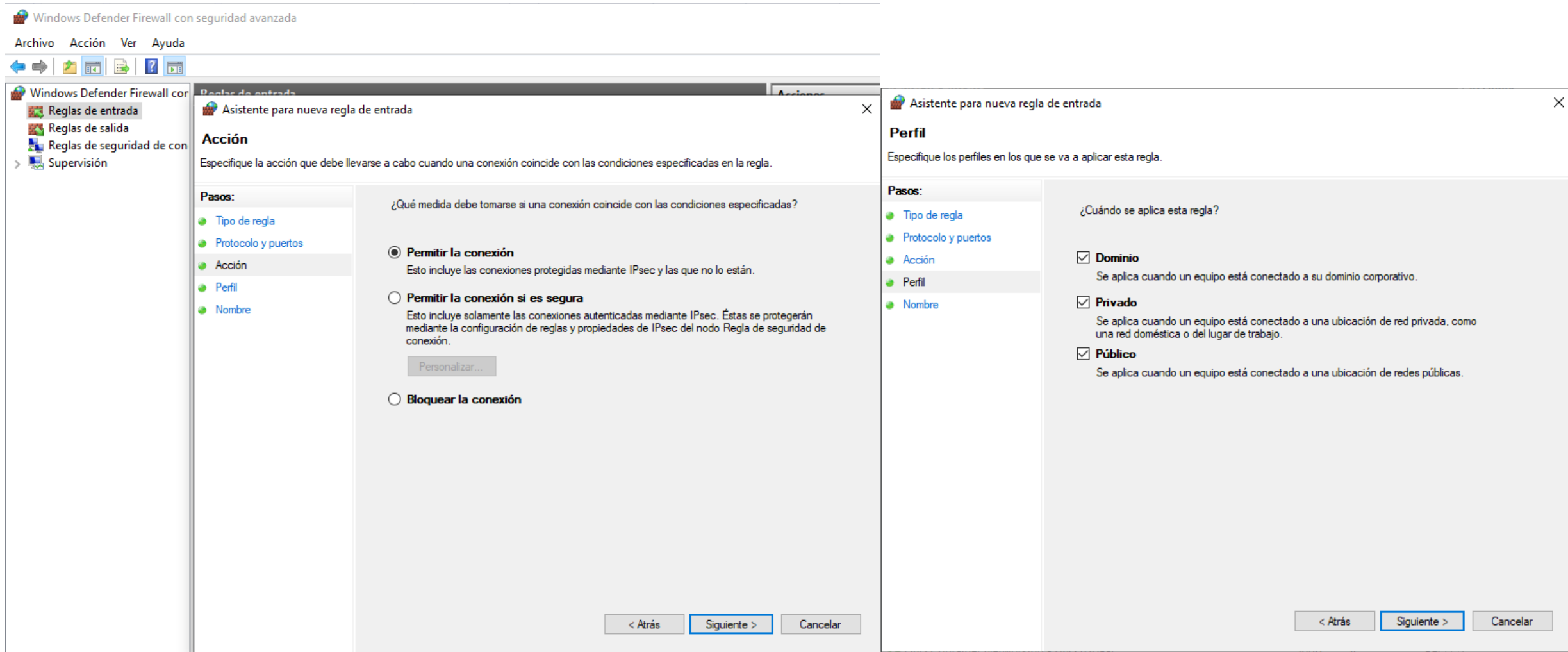
Una vez abierto, seleccionar '*Configuración avanzada*' en el menú de la izquierda, y después, en el menú hacer click con el botón derecho sobre '*Reglas de entrada*', y seleccionar '*Nueva Regla*' (También se puede hacer desde la opción de menú '*Acción*')



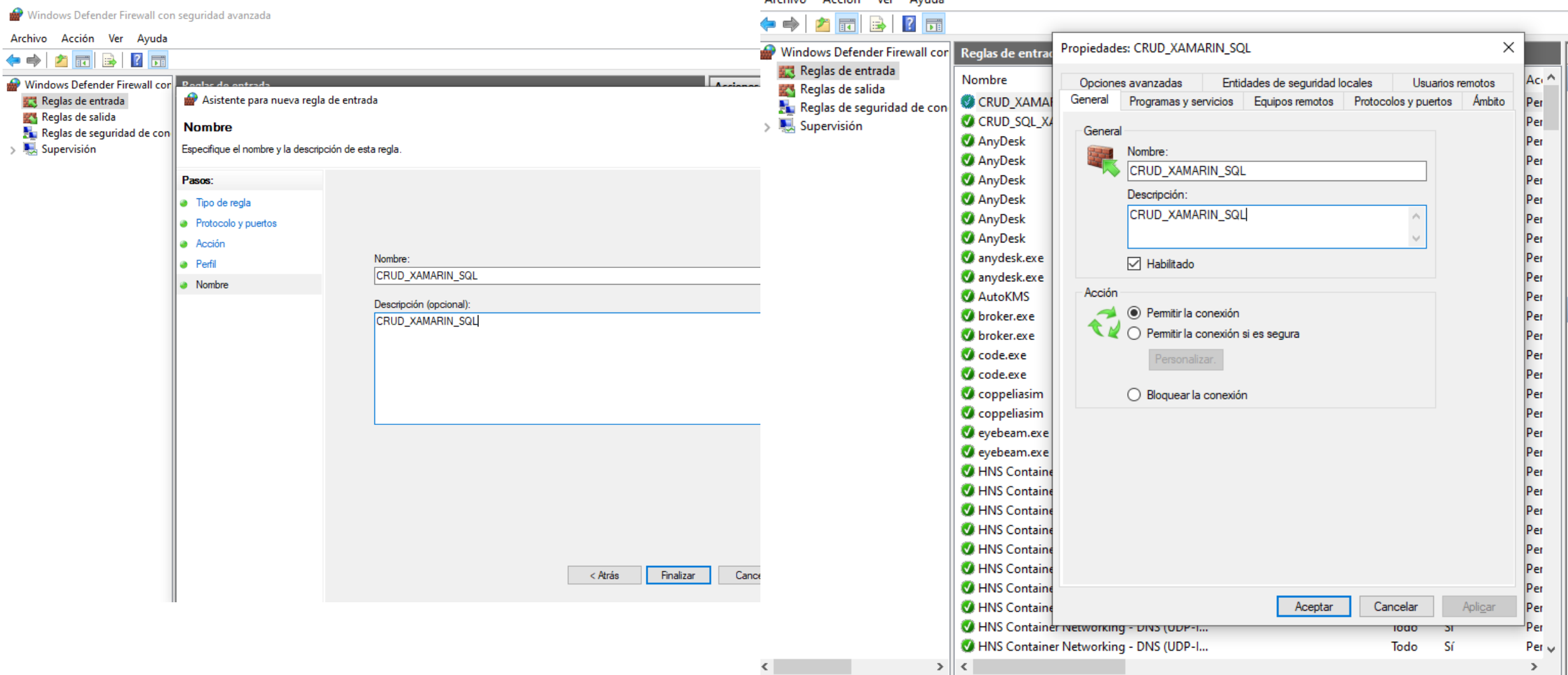
Seleccionar regla tipo '*Puerto*', y después '*TCP*', y escribir 1433 en la opción '*Puerto específico local*'. Esta es la configuración más sencilla, y sirve si esta es la única instancia de SQL Server instalada en el Servidor Windows.



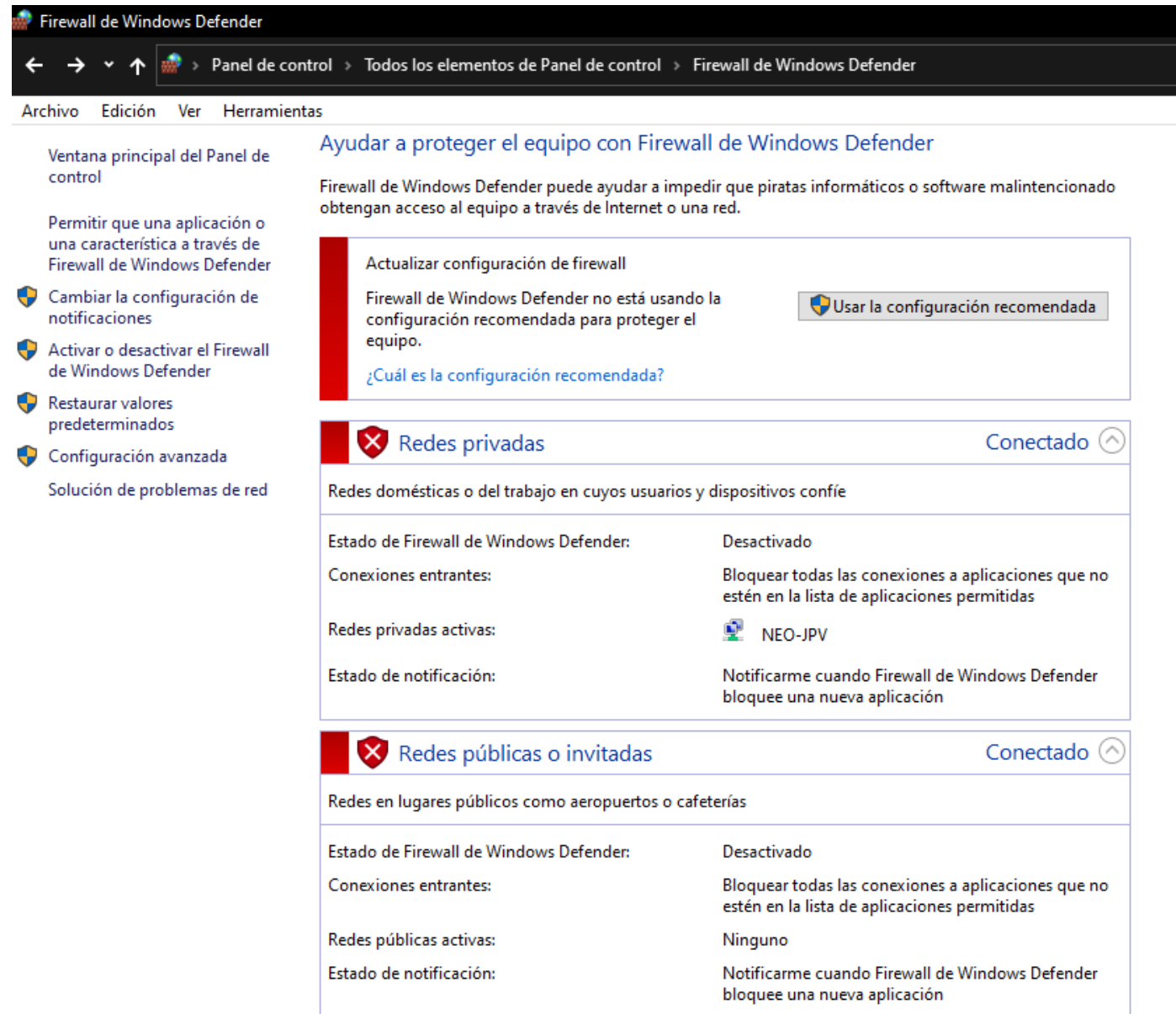
Si hubiera más instancias sería necesario abrir más puertos, activar el servicio **SQL Server Browser**, y abrir otro puerto UDP, y también crear una nueva regla personalizada en el firewall para el servicio. Seleccionando las siguientes opciones (normalmente las que vienen por defecto ya van bien) se finaliza la creación de la regla del firewall, y el telnet por el puerto **1433** desde la máquina cliente ya debería responder:



EN este le asignamos un nombre a nuestra regla en el Firewall, que puede ser el mismo de nuestro proyecto.



Como es un proyecto Local, simulando en nuestro caso el comportamiento de Servidor, vamos a deshabilitar el Firewall si nos ocasiona problemas, solo por el uso de la aplicación. **Nota: Esto es porque es un proyecto que nosotros estamos realizando, no se hace esto en la vida real, ya que estarías permitiendo que cualquier aplicación maliciosa entre a tu pc.**

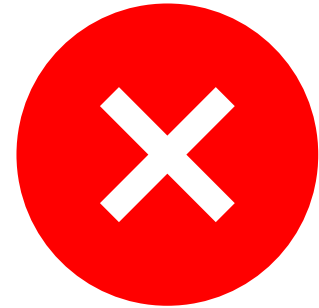




## SI TE DA ERROR EN LA EJECUCION DEL PROGRAMA CUANDO ESTE LISTO CON EL CODIGO C#. SIGUE ESTOS.

- 1) **Abre el Administrador de servicios:** Puedes buscar "services.msc" en el menú Inicio o en la barra de búsqueda de Windows y abrir el "Administrador de servicios".
- 2) **Encuentra el servicio de SQL Server:** En la lista de servicios, busca los servicios relacionados con SQL Server. Por lo general, verás servicios con nombres como "SQL Server (Nombre de la instancia)".
- 3) **Reinicia el servicio de SQL Server:** Haz clic derecho en el servicio de SQL Server que deseas configurar y selecciona "Reiniciar". Esto reiniciará el servicio y aplicará cualquier cambio en la configuración.
- 4) **Configura el Firewall:** Además de reiniciar el servicio, asegúrate de configurar el firewall para permitir conexiones entrantes en el puerto utilizado por SQL Server (generalmente el puerto 1433). Puedes crear una regla de entrada en el Firewall de Windows para permitir el tráfico en ese puerto.

Ten en cuenta que estos pasos pueden variar dependiendo de la versión de SQL Server que estés utilizando y de la configuración específica de tu entorno. Si sigues teniendo problemas para habilitar conexiones remotas, te recomiendo consultar la documentación de SQL Server específica para tu versión o buscar ayuda en foros de soporte técnico relacionados con SQL Server.



Como mencione anteriormente, muchas veces necesitamos reiniciar el servicio de SQL Server, ya que a veces se para según las acciones que hemos realizado.

The screenshot shows the Windows Services console. The 'Services (Local)' window is open, displaying a list of services. The 'SQL Server (MSSQLSERVERJPV)' service is selected, and its sub-services are listed in the right pane. A red box highlights the 'Stop the service', 'Pause the service', and 'Restart the service' links in the left pane. Another red box highlights the 'SQL Server (MSSQLSERVERJPV)' service in the list. A third red box highlights the 'Status' column in the table, showing that the service is 'Running'.

Name	Description	Status	Startup Type	Log On As
Servicio de usuario del portapapeles_448913c3	This user ser...	Running	Manual	Local Syste...
Shared PC Account Manager	Manages pr...	Disabled	Disabled	Local Syste...
Shell Hardware Detection	Provides no...	Running	Automatic	Local Syste...
Sincronizar host_448913c3	This service ...	Running	Automatic (...)	Local Syste...
Smart Card	Manages ac...	Manual (Trig...	Manual (Trig...	Local Service
Smart Card Device Enumeration Service	Creates soft...	Manual (Trig...	Manual (Trig...	Local Syste...
Smart Card Removal Policy	Allows the s...	Manual	Manual	Local Syste...
SNMP Trap	Receives tra...	Manual	Manual	Local Service
Software Protection	Enables the ...	Automatic (...)	Automatic (...)	Network S...
Spatial Data Service	This service ...	Manual	Manual	Local Service
Special Administration Console Helper	Allows adm...	Manual	Manual	Local Syste...
Spot Verifier	Verifies pote...	Manual (Trig...	Manual (Trig...	Local Syste...
SQL Full-text Filter Daemon Launcher (MSSQLSERVERJPV)	Servicio que...	Running	Automatic	NT Service...
SQL Server (MSSQLSERVERJPV)	Proporcion...	Running	Automatic	NT Service...
SQL Server Analysis Services (MSSQLSERVERJPV)	Proporcion...	Running	Automatic	NT Service...
SQL Server Browser	Proporcion...	Running	Automatic	Local Service
SQL Server Launchpad (MSSQLSERVERJPV)	Servicio que...	Running	Automatic	NT Service...
SQL Server VSS Writer	Proporcion...	Running	Automatic	Local Syste...
SSDP Discovery	Discovers n...	Running	Manual	Local Service
State Repository Service	Provides re...	Running	Manual	Local Syste...
Still Image Acquisition Events	Launches a...	Manual	Manual	Local Syste...
Storage Service	Provides en...	Running	Automatic (...)	Local Syste...
Storage Tiers Management	Optimizes t...	Manual	Manual	Local Syste...

# Codigo a pegar en el MainPage.xaml

```
<StackLayout>

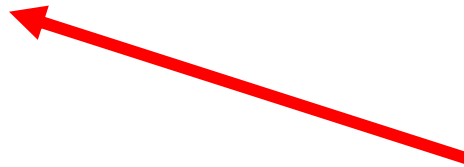
    <Image Source="https://ps.w.org/login-customizer/assets/icon-256x256.png?rev=2455454" WidthRequest="300" Margin="45"/>
    <StackLayout>
        <StackLayout Orientation="Horizontal">
            <Entry Placeholder="Usuario" x:Name="txtUsuario" WidthRequest="170" TextColor="Black" Margin="25" />
            <Entry Placeholder="Contraseña" x:Name="txtPassword" IsPassword="True" WidthRequest="200" TextColor="Black"
Margin="25" />
        </StackLayout>

        <StackLayout Orientation="Horizontal">
            <Button Text="✓ Iniciar sesión" BackgroundColor="Green" WidthRequest="200" TextColor="White" CornerRadius="25"
x:Name="btnLogin" Clicked="btnLogin_Clicked" />
            <Button Text="✗ Cerrar" BackgroundColor="Red" WidthRequest="200" TextColor="White" CornerRadius="25"
x:Name="btnSalir" Clicked="btnSalir_Clicked" />
        </StackLayout>
    </StackLayout>
</StackLayout>
```

Recordar las imágenes o Iconos para nuestro proyecto, que debemos colocar en la carpeta de android, en la carpeta de Resources, en la carpeta Drawable.

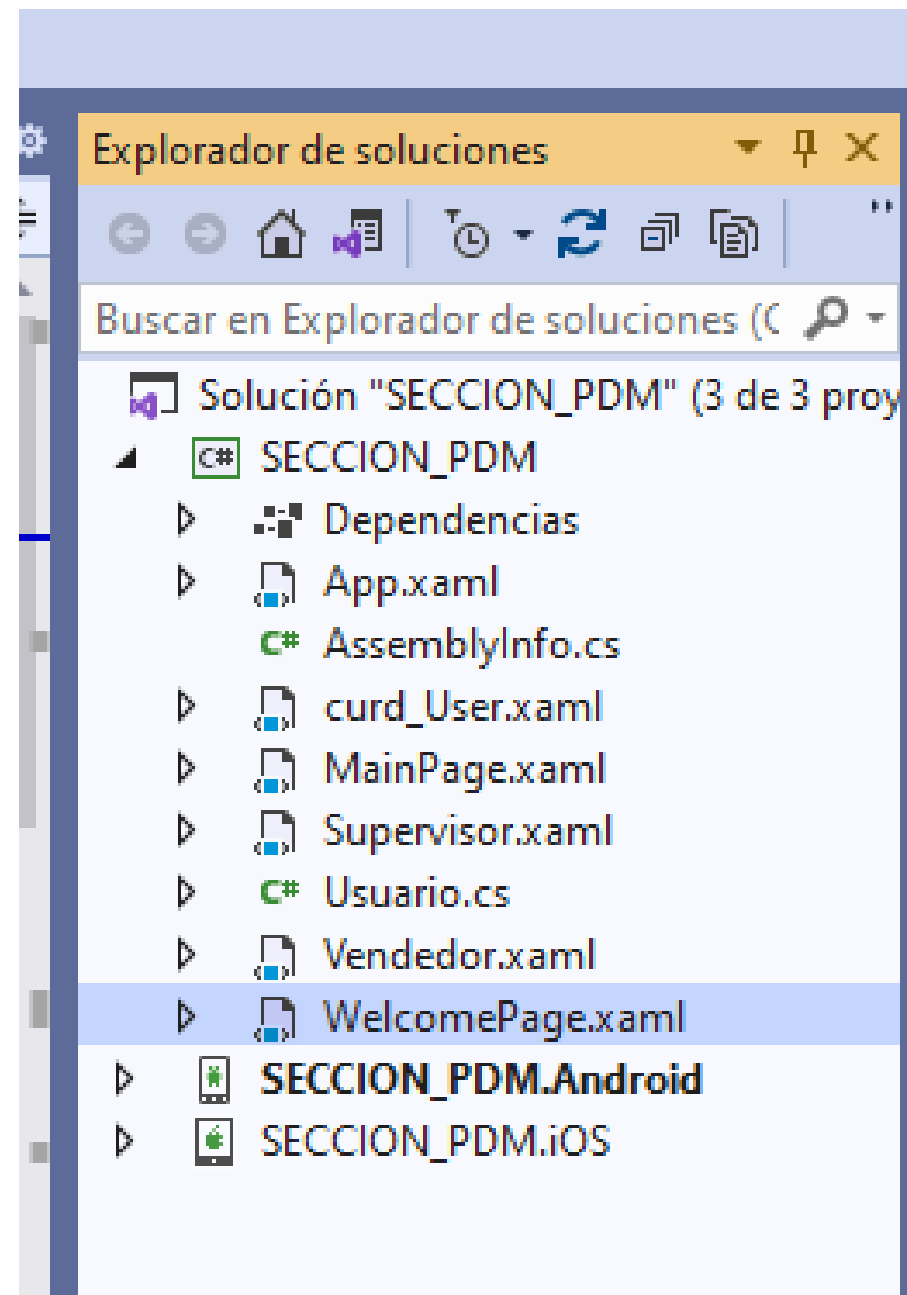
En el código MainPage.xaml.cs, vamos a agregar la librería o plugin que hemos instalado.

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



**Este plugin.**

Vamos a crear 3 archivos, como página de contenidos, la cual será la Pagina **WelcomePage**, la cual puedes ponerle **Administrador**, la página **Vendedor**, y la Pagina **Supervisor**, en las cuales podremos los datos que nos permitan visualizar según se loguean esos datos.



# Vamos a indicarle al Sistema como se comportará el MainPage.xaml.cs

The screenshot shows the Visual Studio IDE with the VENTAS\_2023 project open. The Explorer on the right shows the project structure, with **C# App.xaml.cs** highlighted. The code editor shows the `App` class with the `MainPage` assignment highlighted in yellow.

```
1 using System;
2 using Xamarin.Forms;
3 using Xamarin.Forms.Xaml;
4
5 namespace VENTAS_2023
6 {
7     6 referencias
8     public partial class App : Application
9     {
10         2 referencias
11         public App()
12         {
13             InitializeComponent();
14             MainPage = new NavigationPage( new MainPage());
15         }
16
17         0 referencias
18         protected override void OnStart()
19         {
20         }
21
22         0 referencias
23         protected override void OnSleep()
24         {
25         }
26
27         0 referencias
28         protected override void OnResume()
29         {
30         }
31     }
32 }
```

**MainPage = new NavigationPage( new MainPage());**

Escribe tal cual esto que tenemos en amarillo, para la navegacion.

# Codigo para la Pagina MainPage.xaml.cs

```
// Evento del Boton Login
private async void btnLogin_Clicked(object sender, EventArgs e)
{
    // Declaramos dos Variables, y guardamos lo que escribe el usuario:
    string usuario = txtUsuario.Text;
    string password = txtPassword.Text;

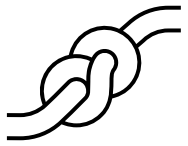
    // Validamos con la Funcion IsNull, que no esten vacios el Usuario y el Password:
    if (string.IsNullOrEmpty(usuario) || string.IsNullOrEmpty(password))
    {
        // Si los Datos estan vacios muestra este mensaje:
        await DisplayAlert("Error", "Ingrese un usuario y contraseña válidos", "OK");
        return;
    }
    // Establecemos la Cadena de Conexion con el SQL Server y Nuestros datos:
    string connectionString = "Server=10.0.0.5,1433;Database=LOGIN_2023;User Id=JUANCITO; Password=123456;";

    // conectar a la base de datos
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        await connection.OpenAsync();

        // crear un comando para seleccionar un usuario con el nombre de usuario y contraseña especificados
        using (SqlCommand command = new SqlCommand("SELECT id_usuario, id_rol FROM usuarios WHERE nombre_usuario = @usuario AND password = @password", connection))
        {
            command.Parameters.AddWithValue("@usuario", usuario);
            command.Parameters.AddWithValue("@password", password);

            // ejecutar el comando y recibir el resultado

            using (SqlDataReader reader = await command.ExecuteReaderAsync())
            {
                // verificar si se obtuvo algún resultado
                if (reader.HasRows)
                {
                    // leer los datos del usuario
                    reader.Read();
                    int idRoles = reader.GetInt32(1);
                    // validar si el usuario es administrador
                    if (idRoles == 1)
                    { // Si es válido abrir la página de Administrador
                        await Navigation.PushAsync(new WelcomePage());
                    }
                    else if (idRoles == 2)
                    { // Si es válido abrir la página de Supervisor
                        await Navigation.PushAsync(new Supevisor());
                    }
                    else if (idRoles == 3)
                    { // Si es válido abrir la página vendedora
                        await Navigation.PushAsync(new Vendedor());
                    }
                    else
                    { // Si no es válido enviar este mensaje de Error:
                        await DisplayAlert("Error", "Usuario o contraseña incorrectos", "Intentar nuevamente");
                    }
                }
                else
                { // Si no es válido enviar este mensaje de Error:
                    await DisplayAlert("Error", "Usuario o contraseña incorrectos", "Intentar nuevamente");
                }
            }
        }
    }
}
```



**Cadena de conexión **válida** para nuestro caso, existen otras como crear una clase y que podemos utilizarla en nuestro código llamándola en algún método o función .**

```
string connectionString = "Server=xxx.xxx.xxx.xxx,1433;Database=SECCION_PDM;User Id=JUANCITO;Password=123456;"
```

Es una cadena de conexión a una base de datos SQL Server. Esta cadena de conexión es utilizada en el código de una aplicación para establecer una conexión con la base de datos.

A continuación, se explica el significado de cada una de las partes de la cadena de conexión:

- **"Server=10.0.0.4,1433"**: especifica la dirección IP del servidor de base de datos y el número de puerto utilizado por el servidor.
- **"Database=SECCION\_PDM"**: indica el nombre de la base de datos a la que se desea conectar.
- **"User Id=JUANCITO; Password=123456"**: son las credenciales de inicio de sesión que se utilizarán para conectarse a la base de datos. En este caso, el nombre de usuario es "JUANCITO" y la contraseña es "123456".





```
// Evento del Boton Salir:
private async void btnSalir_Clicked(object sender, EventArgs e)
{    // Si el usuario le da a salir, le saldra este menseje con las dos opciones Salir (Si o No) Si es No lo devuelve al form
:
    var respuesta = await DisplayAlert("Confirmación", "¿Seguro que deseas cerrar la aplicación?", "Si", "No");

    if (respuesta)
    {    // Si la Repuesta es que si, Cierra la aplicacion:
        System.Diagnostics.Process.GetCurrentProcess().CloseMainWindow();
    }
}
```

Existen muchas formas de crear un boton para Salir, te dejare más adelante algunas opciones.

Vamos a crear ahora las interfaces basicas, las cuales luego modificaremos según las opciones que necesitamos agregar en nuestro proyecto, como quien tiene acceso a ver y hacer los CRUD, quien solo puedo buscar y listar datos, y quien tiene acceso a otras tablas como por ejemplo ventas y hacer los insert.

## Codigo del Archivo WelcomePage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="SECCION_PDM.WelcomePage"
             Title="Bienvenido Admin">

    <ContentPage.Content>
        <StackLayout BackgroundColor="LightGray">
            <Label Text="¡Bienvenido! Administrador"
                  FontAttributes="Bold"
                  FontSize="36"
                  TextColor="Black"
                  VerticalTextAlignment="Center"
                  HorizontalTextAlignment="Center" />

            <StackLayout BackgroundColor="LightGray">
                <Image Source="https://advisertecnology.com/wp-content/uploads/2021/08/portadajpv.png" Aspect="Fill"/>
            </StackLayout>
            <Button Text="Administrar Usuarios" TextColor="Black"
                   HorizontalOptions="Center"
                   BackgroundColor="GreenYellow"
                   WidthRequest="250"
                   HeightRequest="50"
                   BorderWidth="2"
                   BorderColor="Red"
                   CornerRadius="25"
                   />
        </StackLayout>

    </ContentPage.Content>
</ContentPage>
```

# Codigo del Archivo Supervisor.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="SECCION_PDM.Supervisor">
    <ContentPage.Content>
        <StackLayout BackgroundColor="LightGray">
            <Label Text="¡Bienvenido! Supervisor"
                  FontAttributes="Bold"
                  FontSize="36"
                  TextColor="Black"
                  VerticalTextAlignment="Center"
                  HorizontalTextAlignment="Center" />

            <StackLayout BackgroundColor="LightGray">
                <Image Source="https://cdn.icon-icons.com/icons2/966/PNG/128/Users_icon-icons.com_74706.png" Aspect="Fill"/>
            </StackLayout>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

# Codigo del Archivo Vendedor.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="SECCION_PDM.Vendedor">
    <ContentPage.Content>
        <StackLayout BackgroundColor="LightGray">
            <Label Text="¡Bienvenido! Vendedor"
                  FontAttributes="Bold"
                  FontSize="36"
                  TextColor="Black"
                  VerticalTextAlignment="Center"
                  HorizontalTextAlignment="Center" />

            <StackLayout BackgroundColor="LightGray">
                <Image Source="https://cdn-icons-png.flaticon.com/512/2133/2133124.png" Aspect="AspectFit"/>
            </StackLayout>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Object Explorer

JUANCITO\MSSQLSE...M - dbo.usuarios

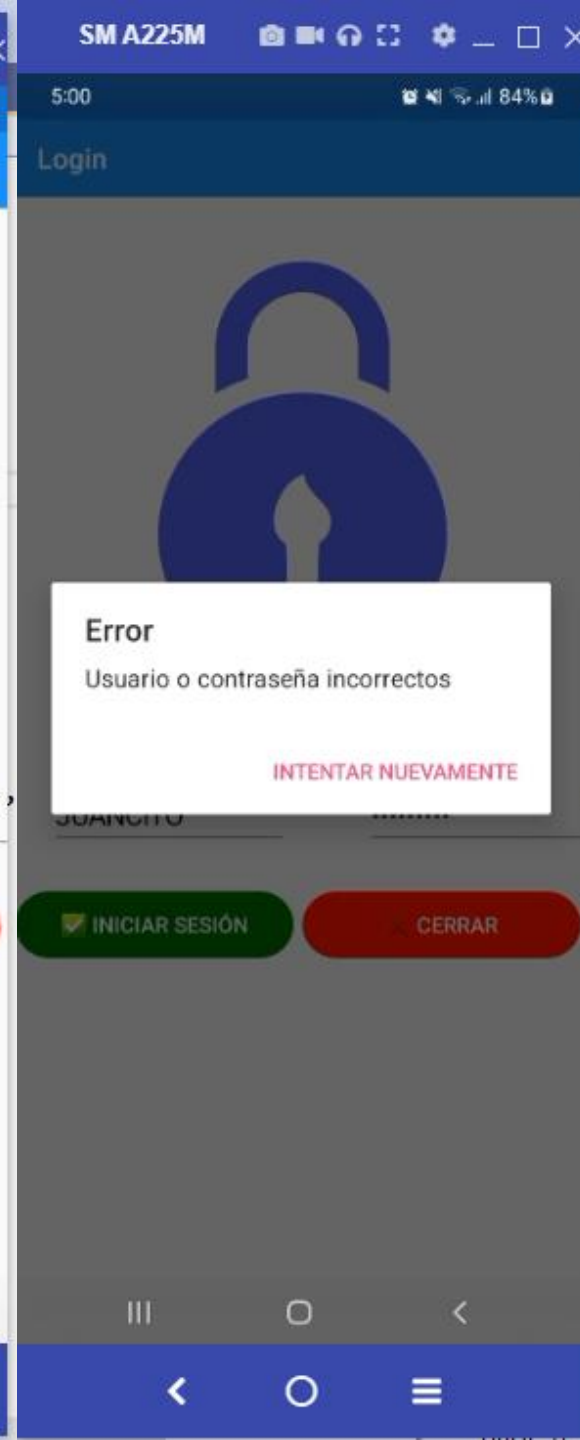
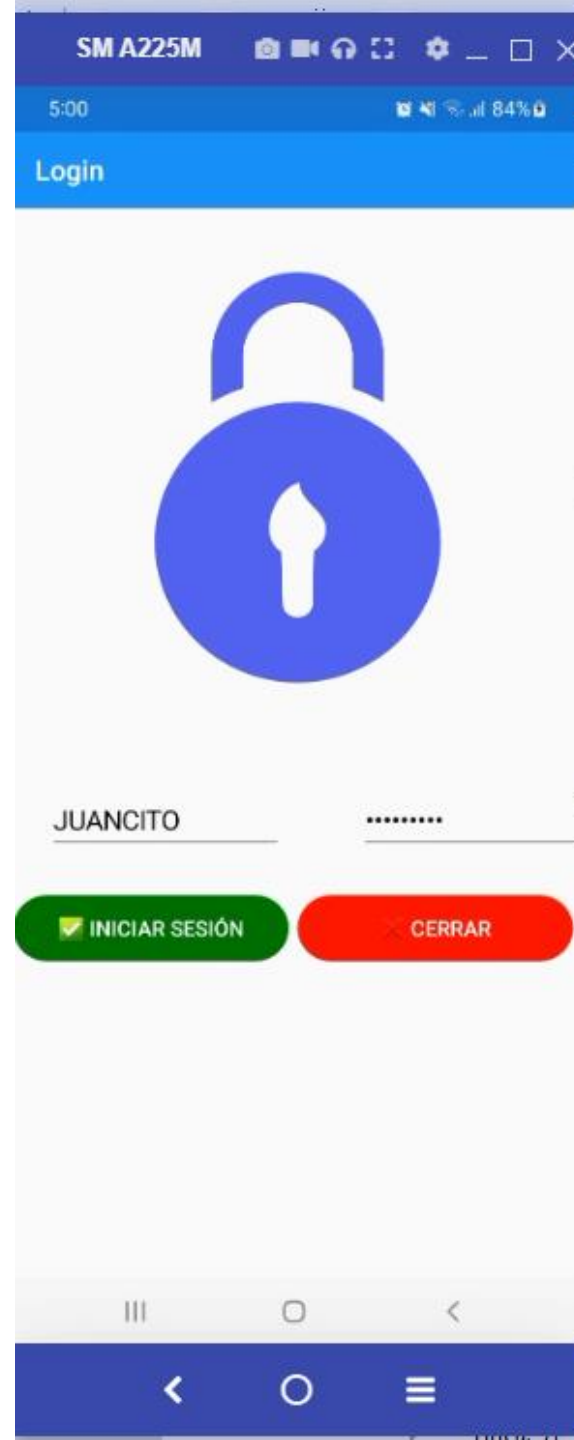
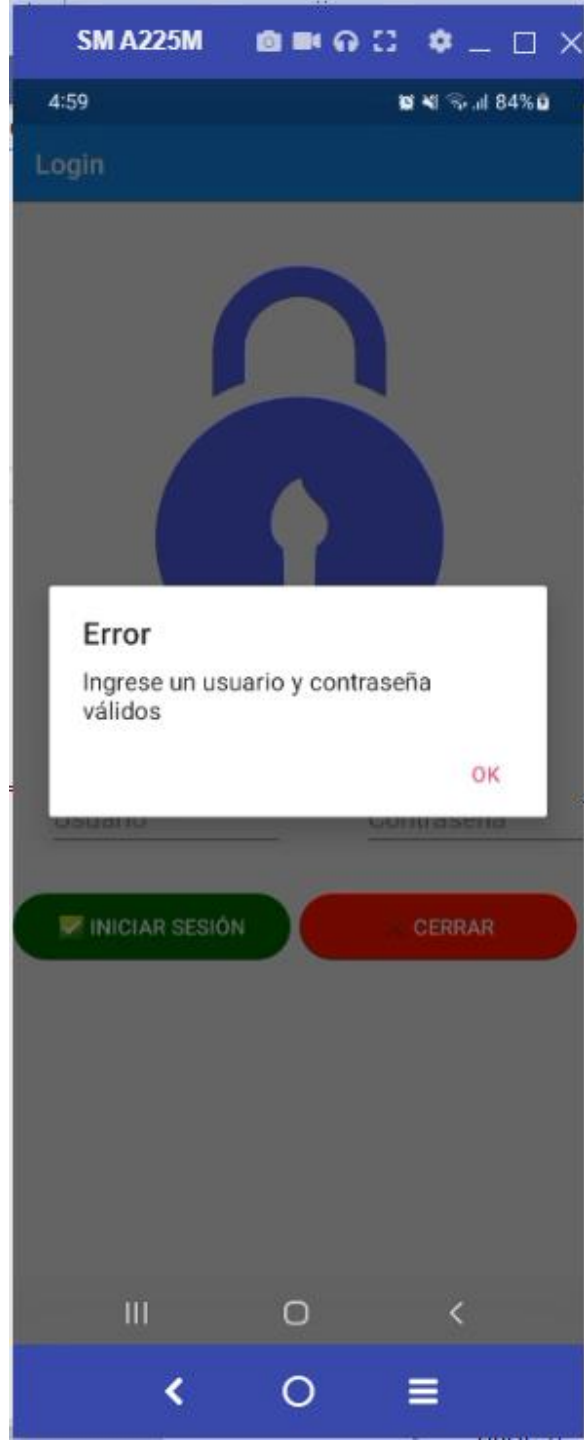
	id_usuario	nombre_usuario	password	id_rol
	1	JUANCITO	ADMIN@123	1
▶	2	usuario2	password2	2
	3	usuario3	password3	3
	4	usuario4	password4	1
	5	usuario5	password5	2
	6	juancito	123456	1
	7	Dariel	123456	2
•	NULL	NULL	NULL	NULL

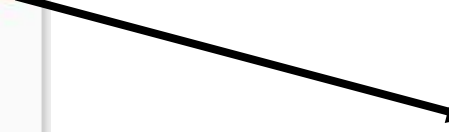
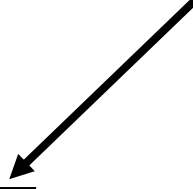
Object Explorer

JUANCITO\MSSQLSE...\_PDM - dbo.roles

	id_rol	nombre_rol
▶	1	administrador ...
	2	supervisor
	3	vendedor
•	NULL	NULL

**Aquí podemos ver los registros que tenemos en nuestra base de datos, la cual me permite ver y usarlos ahora en mi aplicación. Podemos intentar acceder con los usuarios y probar en cada caso.**







Te toca a ti.

**Hacer esta aplicacion, y con lo que han aprendido a nivel de la Interfaz vamos Aplicarlo aqui nuevamente y con codigo xaml y C# darle funcionalidad.**



**Creando la Presentacion del Profesor en el menu professor, y en estuantes la lista de Estudiantes con sus fotografias y en la pagina Principal en el MainPage.xaml agregar una descripcion de la Universidad y su materia de PDM-742**

