



MANUAL DE INSTALACION

SISTEMA DE TOMA DE INVENTARIO

TABLA DE CONTENIDO

| | |
|---|----|
| INTRODUCCION | 3 |
| Objetivos del Sistema | 3 |
| Alcances del Sistema | 3 |
| instalacion de base de datos | 5 |
| Requisitos previos | 5 |
| Hardware mínimo: | 5 |
| Software: | 5 |
| Archivos necesarios | 5 |
| Instalación paso a paso | 5 |
| Paso 1: Ejecutar el Instalador | 5 |
| Paso 2: Seleccionar Características | 5 |
| Paso 3: Configurar Nombre de Instancia | 5 |
| Paso 4: Configurar Modo de Autenticación | 6 |
| Instalacion de SQL Server Management Studio | 6 |
| Configuracion de base datos | 7 |
| Creacion de Base de datos. | 7 |
| Creacion de Tablas | 7 |
| Creacion de Procedimiento Almacenado | 7 |
| Creacion de Link Server | 10 |
| Configuracion de iis | 11 |
| Activar Internet Information Service | 11 |
| Configurar backend en IIS. | 11 |
| Configuración de la Pagina Web. | 16 |

INTRODUCCION

Este Manual de Instalación del Sistema de Toma de Inventario del Proyecto Especial CORAH proporciona las instrucciones necesarias para la correcta implementación técnica del sistema en los equipos designados. Está dirigido al personal de soporte técnico, administradores de sistemas y responsables de TI encargados de realizar la instalación, configuración inicial y puesta en marcha del software.

El documento detalla los requisitos mínimos del sistema, las dependencias necesarias, los pasos para la instalación en distintos entornos (servidores, estaciones de trabajo, o dispositivos móviles si aplica), así como recomendaciones de seguridad y mantenimiento básico.

Una instalación adecuada garantiza el funcionamiento estable y seguro del sistema, permitiendo a los usuarios finales acceder a todas sus funcionalidades sin interrupciones ni errores.

Se recomienda seguir cada paso con atención y contar con los permisos adecuados para ejecutar procesos administrativos en los dispositivos involucrados.

Objetivos del Sistema

- Optimizar la gestión de inventario en almacenes centrales, campamentos operativos y brigadas móviles.
- Garantizar la trazabilidad de todos los bienes desde su ingreso hasta su destino final.
- Reducir pérdidas y extravíos mediante controles automatizados y auditorías periódicas.
- Agilizar la toma de decisiones logísticas con reportes actualizados y acceso rápido a la información.
- Apoyar la transparencia institucional, facilitando la supervisión y control de los recursos por parte de las autoridades competentes.

Alcances del Sistema

- Control de entradas y salidas de bienes en almacenes centrales, almacenes de campo y campamentos móviles.
- Registro detallado de equipos asignados al personal y su estado operativo.
- Gestión de lotes de herramientas, materiales de consumo, equipos de seguridad, ropa de campaña, medicamentos y repuestos.
- Generación de reportes logísticos por zona, brigada, tipo de recurso o periodo de tiempo.
- Control de inventario físico y digital con soporte para auditorías internas y externas.

- Acceso multiusuario con perfiles diferenciados (operativo, supervisor, administrador).
- Integración con otros sistemas institucionales, si corresponde, para facilitar la interoperabilidad.

Este manual está dirigido al personal logístico, encargados de almacén, supervisores de campo y usuarios administrativos. Contiene instrucciones detalladas para el uso del sistema, ejemplos prácticos y recomendaciones que aseguran una correcta implementación en las operaciones del Proyecto Especial CORAH.

INSTALACION DE BASE DE DATOS

El sistema de toma de inventario usa como motor de base de datos SQL Server, a continuación, se muestra los requisitos previos, archivos necesarios, instalación paso a paso e instalación de SSMS.

Requisitos previos

Hardware mínimo:

- Procesador: 1.4 GHz 64-bit
- Memoria RAM: 2 GB (4 GB recomendados)
- Espacio en disco: 6 GB mínimo (según características instaladas)

Software:

- Sistema operativo: Windows 10, Windows Server 2016 o superior
- .NET Framework 4.8 (instalado automáticamente si no está presente)
- Acceso de administrador en el sistema

Archivos necesarios

- **SQL Server Installer** (versión Express, Developer o Standard): Puedes descargarlo desde: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
- **SQL Server Management Studio (SSMS)** – Para administrar bases de datos (se instala por separado). <https://aka.ms/ssmsfullsetup>

Instalación paso a paso

Paso 1: Ejecutar el Instalador

- Abre el archivo descargado (SQL2019-SSEI-Dev.exe o similar).
- Selecciona la opción **Basic**, **Custom**, o **Download Media** (recomendado: *Custom*).

Paso 2: Seleccionar Características

Marca las características necesarias:

- **Database Engine Services** (obligatorio)
- **SQL Server Replication** (opcional)
- **Client Tools Connectivity**
- **Client Tools Backwards Compatibility** (opcional)

Paso 3: Configurar Nombre de Instancia

- Usa la opción **Default Instance** (recomendado: MSSQLSERVER)
- O asigna un nombre personalizado si tienes múltiples instancias

Paso 4: Configurar Modo de Autenticación

- Selecciona **Mixed Mode (SQL Server authentication and Windows authentication)**
- Crea una contraseña segura para el usuario sa
- Agrega el usuario actual como administrador de SQL Server

Paso 5: Verificación e Instalación

- Revisa el resumen de configuración
- Haz clic en **Install**
- Espera a que el proceso termine y luego haz clic en **Close**

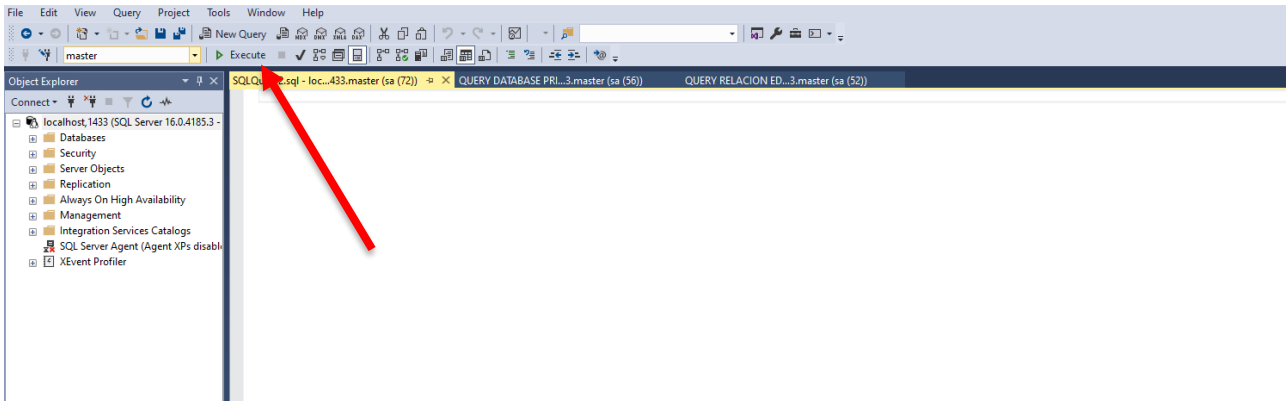
Instalacion de SQL Server Management Studio

1. Ejecuta el instalador descargado de SSMS
2. Acepta los términos y haz clic en Install
3. Una vez finalizado, abre SSMS y conéctate al servidor local (localhost o .\SQLEXPRESS, según configuración).

CONFIGURACION DE BASE DATOS

Creacion de Base de datos.

Ejecutaremos un nuevo query, se muestra imagen de referencia



Aquí ingresaremos los siguientes datos.

```
/*CREACION DE DATABASE GENERAL */  
CREATE DATABASE principal;
```

Creacion de Tablas

-- CREACION DE TABLAS

```
CREATE TABLE periodo(  
    id int identity primary key,  
    descripcion varchar(400),  
    nomBaseDatos varchar(400),  
    ano int,  
    fechaCreacion datetime,  
    estadoCargado bit,  
    estadoBackup bit,  
    estadoCierre bit  
);
```

```
CREATE TABLE usuario(  
    id int identity primary key,  
    nombre varchar(200),  
    usuario varchar(50),  
    password varchar(200)  
);
```

```
insert into usuario (nombre, usuario, password) values('administrador','admin','46-51-44-38-57-112-125-100-107');
```

Creacion de Procedimiento Almacenado

```
/*CREACION DE PROCEDIMIENTO ALMACENADO */  
--SP CREAR PERIODO  
CREATE PROCEDURE sp_InsertarPeriodo(  
    @descripcion varchar(400),  
    @nombreBaseDatos varchar(400),  
    @ano int,  
    @fechaCreacion datetime,  
    @estadoCargado bit,  
    @estadoBackup bit,  
    @estadocierre bit
```

```

)
AS
BEGIN
    INSERT INTO periodo (descripcion, nomBaseDatos, ano, fechaCreacion, estadoCargado, estadoBackup, estadoCierre) VALUES
    (@descripcion,@nombreBaseDatos,@ano,@fechaCreacion,@estadoCargado,@estadoBackup,@estadocierre)
END

--SP CREAR PERIODO
CREATE PROCEDURE sp_ActualizarPeriodo(
    @id int,
    @descripcion varchar(400)
)
AS
BEGIN
    UPDATE periodo set descripcion=@descripcion WHERE id=@id
END

--SP CREAR BASE DE DATOS
CREATE PROCEDURE sp_crearBaseDatos
    @nombre_base_datos NVARCHAR(128)
AS
BEGIN
    DECLARE @sql NVARCHAR(MAX)

    -- Verificar si la base de datos ya existe
    IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = @nombre_base_datos)
    BEGIN
        -- Construir el comando SQL para crear la base de datos
        SET @sql = 'CREATE DATABASE ' + QUOTENAME(@nombre_base_datos)

        -- Ejecutar el comando SQL
        EXEC sp_executesql @sql
        PRINT 'Base de datos creada: ' + @nombre_base_datos
    END
    ELSE
    BEGIN
        PRINT 'La base de datos ya existe: ' + @nombre_base_datos
    END
END

--SP ELIMINAR BASE DE DATOS
CREATE PROCEDURE sp_eliminarBaseDatos
    @nombre_base_datos NVARCHAR(128)
AS
BEGIN
    DECLARE @sql NVARCHAR(MAX)

    -- Verificar si la base de datos ya existe
    IF EXISTS (SELECT * FROM sys.databases WHERE name = @nombre_base_datos)
    BEGIN
        -- Construir el comando SQL para crear la base de datos
        SET @sql = 'DROP DATABASE IF EXISTS ' + QUOTENAME(@nombre_base_datos)

        -- Ejecutar el comando SQL
        EXEC sp_executesql @sql
        PRINT 'Base de datos eliminada: ' + @nombre_base_datos
    END
    ELSE
    BEGIN
        PRINT 'La base de datos no existe: ' + @nombre_base_datos
    END
END

```



```

        END
--SP CREAR TABLAS
CREATE PROCEDURE SP_TABLAS(
    @nombre_base_datos nvarchar(128),
    @nombreTabla nvarchar(128),
    @nombreColumn nvarchar(max)
)
AS
BEGIN
    DECLARE @sql NVARCHAR(MAX)
    EXEC sp_executesql @sql;
    SET @sql = 'USE ' + QUOTENAME(@nombre_base_datos) + '
                CREATE TABLE ' + QUOTENAME(@nombreTabla) +'(' + @nombreColumn + ')'
    EXEC sp_executesql @sql
END

--SP CREAR EXPORTACION TABLAS
CREATE PROCEDURE SP_InsertDatosDinamico
(
    @nombre_base_datos nvarchar(120),
    @nombreTabla nvarchar(128),
    @nombreDatosTabla nvarchar(max),
    @datosInsert nvarchar(max),
    @nombreSentencia nvarchar(max)
)
AS
BEGIN
    BEGIN TRANSACTION
    BEGIN TRY
        DECLARE @sql NVARCHAR(MAX)
        SET @sql = 'USE ' + QUOTENAME(@nombre_base_datos) + '
                INSERT INTO' + QUOTENAME(@nombreTabla) +'(' + @nombreDatosTabla + ')' + ' SELECT ' + @datosInsert + ' FROM
OPENQUERY(ServidorInventarioGeneral, ' + @nombreSentencia + ''')'
        EXEC sp_executesql @sql
        --INSERT INTO dbo.donacion(id,nombre)
        --SELECT * FROM OPENQUERY(ServidorInventarioGeneral, 'select * from EDBCORAH.dbo.edbACTBIEADQ');
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'ERROR AL EXPORTAR LOS DATOS: ' + ERROR_MESSAGE();
    END CATCH
END;

--SP ACTUALIZAR ESTADO DE CARGA PERIODO
CREATE PROCEDURE SP_UpdateEstadoCargaPeriodo(
    @id int
)
AS
BEGIN
    UPDATE periodo SET estadoCargado =1 WHERE id =@id
END

--SP CREAR TYPE TABLE DBO.IntList
CREATE PROCEDURE SP_TYPEINT(
    @nombre_base_datos nvarchar(128)
)
AS
BEGIN

```

```

        DECLARE @sql NVARCHAR(MAX)
        EXEC sp_executesql @sql;
        SET @sql = 'USE ' + QUOTENAME(@nombre_base_datos) + ';
                                CREATE TYPE dbo.IntList AS TABLE (Id INT)'

        EXEC sp_executesql @sql
END

--SP CREAM TYPE TABLE DBO.StringList
CREATE PROCEDURE SP_TYPESTRING(
        @nombre_base_datos nvarchar(128)
)
AS
BEGIN
        DECLARE @sql NVARCHAR(MAX)
        EXEC sp_executesql @sql;
        SET @sql = 'USE ' + QUOTENAME(@nombre_base_datos) + ';
                                CREATE TYPE dbo.StringList AS TABLE (Codigos varchar(100))'

        EXEC sp_executesql @sql
END

```

Creacion de Link Server

Creamos la conexion de la base de datos de EDBCORAH con la base de datos local.

```

EXEC sp_addlinkedserver
        @server = 'ServidorInventarioGeneral',    -- Nombre del Linked Server
        @provider = 'SQLNCLI',                    -- Proveedor (SQL Server Native Client)
        @datasrc = 'sia.corahperu.org\siaserver, 63359', -- Nombre o IP del servidor remoto
        @srvproduct = '';                          -- Producto (vacío para SQL Server);

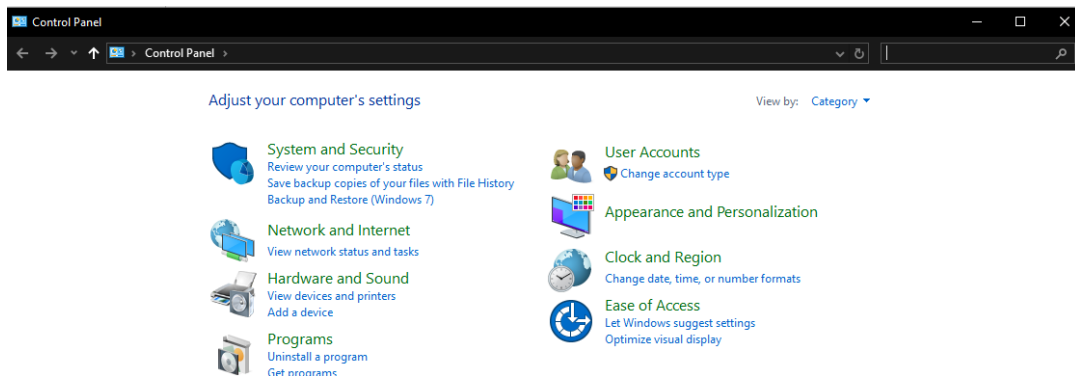
EXEC sp_addlinkedsrvlogin
        @rmtsrvname = 'ServidorInventarioGeneral', -- Nombre del Linked Server
        @useself = 'FALSE',                        -- Usar las credenciales del usuario local
        @rmtuser = 'sia',                          -- Nombre de usuario para acceder al servidor remoto
        @rmtpassword = 'w240q!#zWL7c*20i22*';    -- Contraseña del servidor remote

```

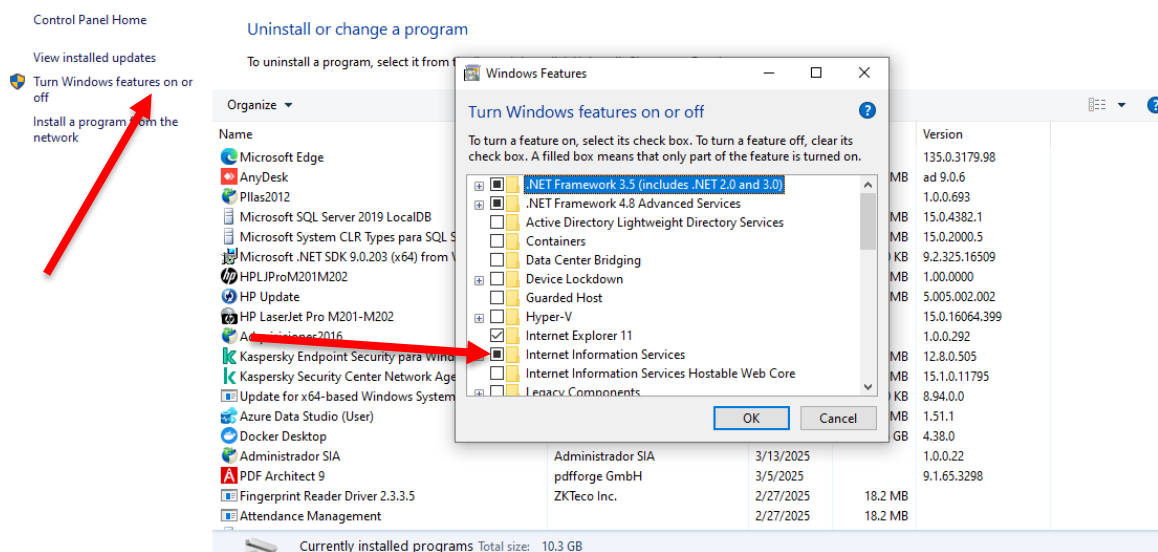
CONFIGURACION DE IIS

Activar Internet Information Service

- ✓ Ingresamos al panel de control



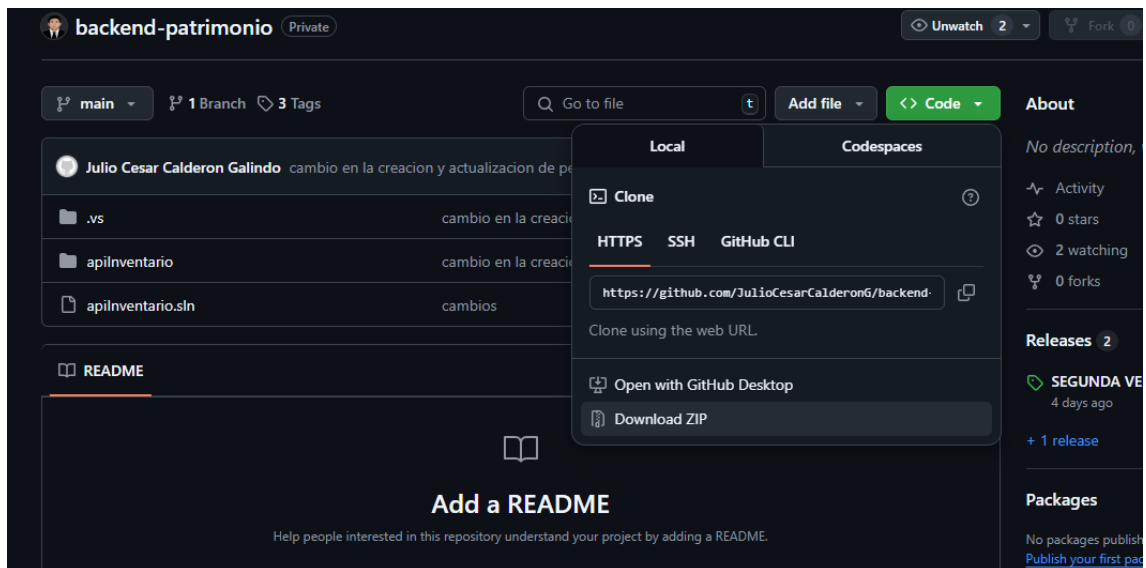
- ✓ Ingresamos a la opción Panel de Control>Programas>Programas y Funciones, y seleccionamos la opción de **Activar o desactivar las funciones de Windows** se abrirá una ventana emergente aca activamos la opción de de **Internet Information Service**, se adjunta imagen de referencia.



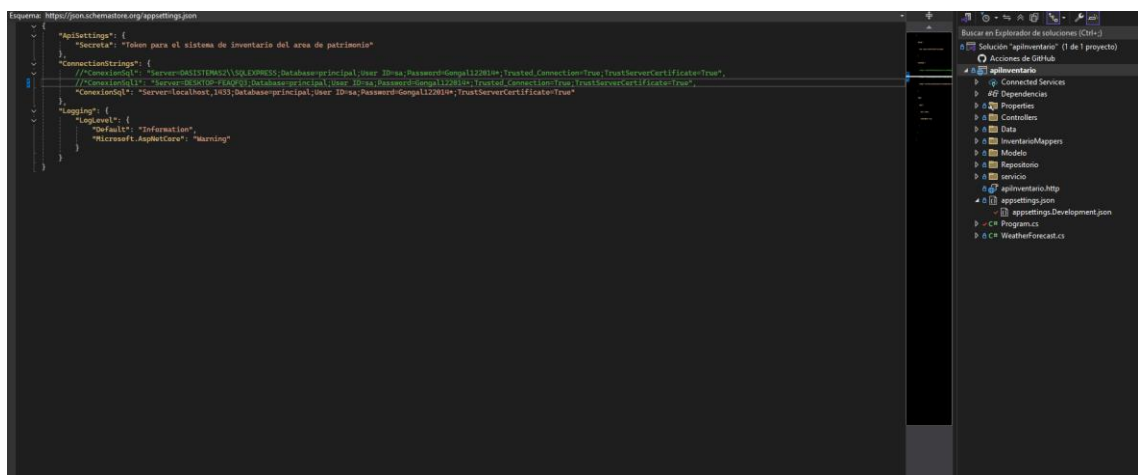
Configurar backend en IIS.

La opción de Internet Information Service se utilizará para configurar el Backend del sistema de Toma de Inventario.

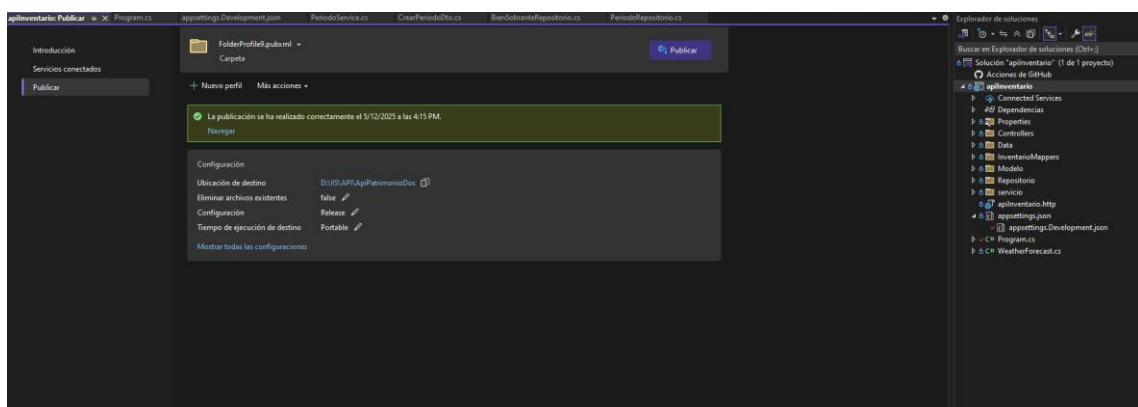
- ✓ Descargamos el proyecto que esta alojado en GitHub.



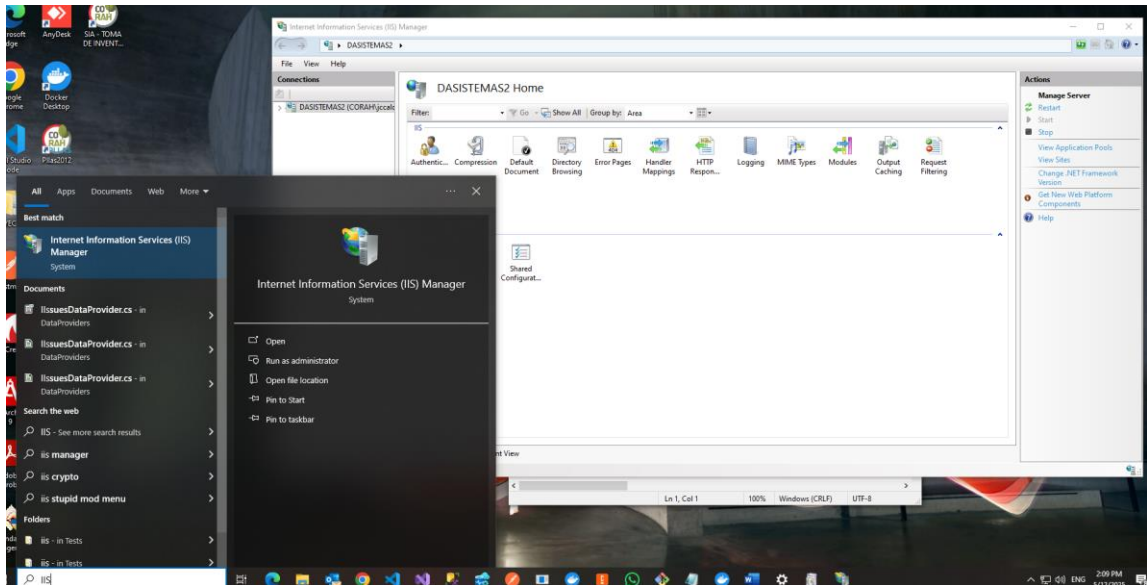
- ✓ Abrimos el proyecto con Visual Studio 2022, y verificamos que las variables de entorno coincidan con la base de datos.



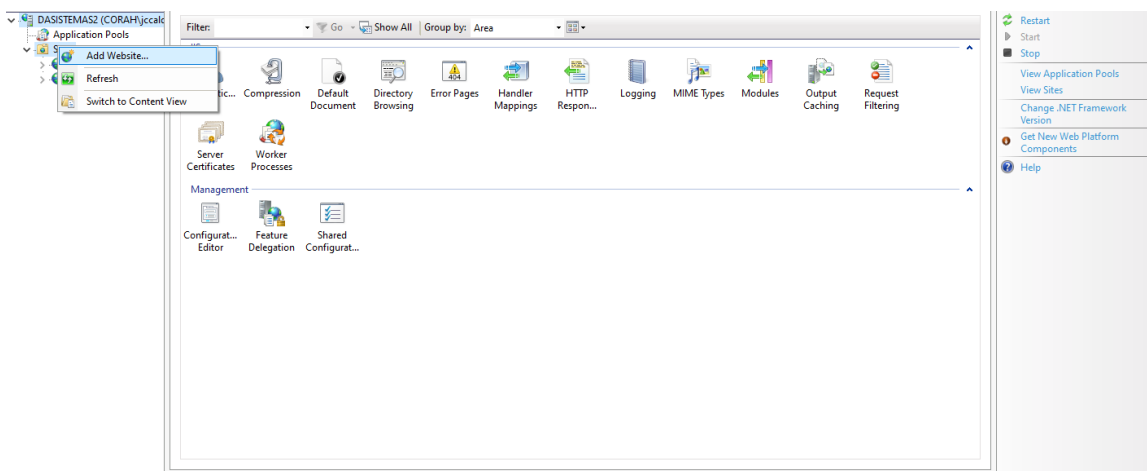
- ✓ Generamos la publicación del proyecto.



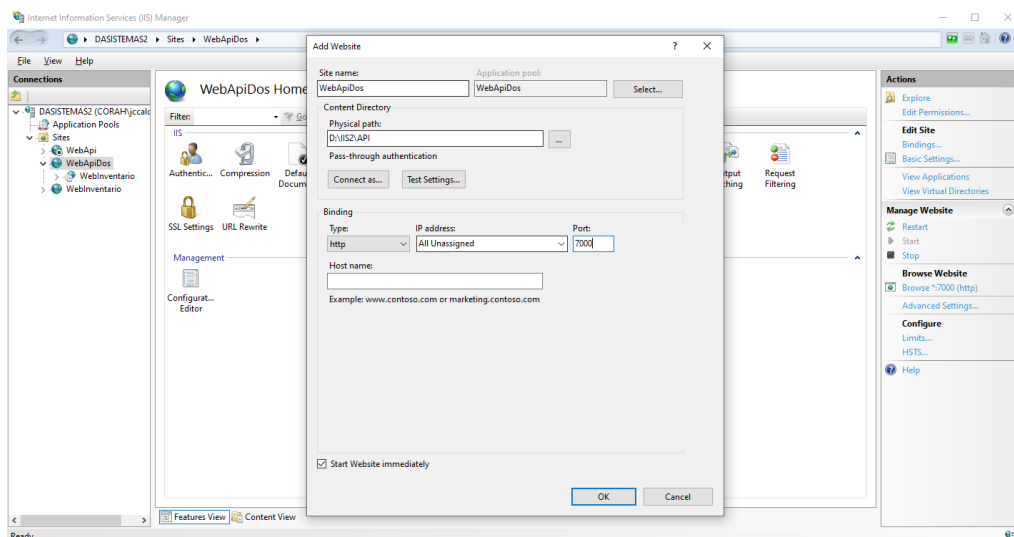
- ✓ Buscamos Internet Information Service y damos click en él, este nos abrirá la interfaz gráfica, se adjunta imagen de referencia.



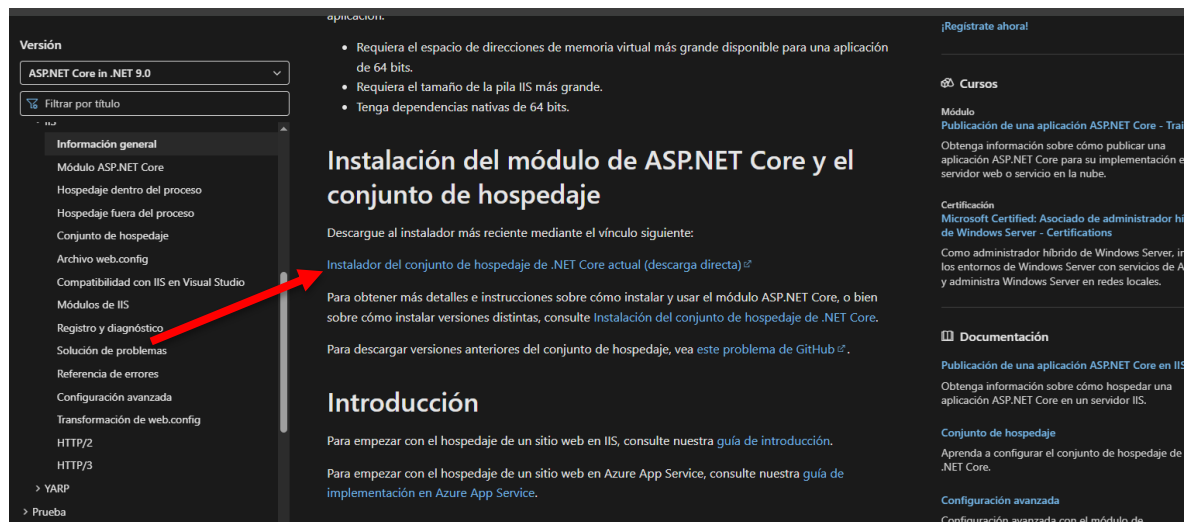
- ✓ Damos click derecho en sites y seleccionamos Add Website, se adjunta imagen de referencia.



- ✓ Completamos los datos que nos solicita, la carpeta que seleccionemos deberá contar con un archivo .html. se adjunta imagen de referencia.

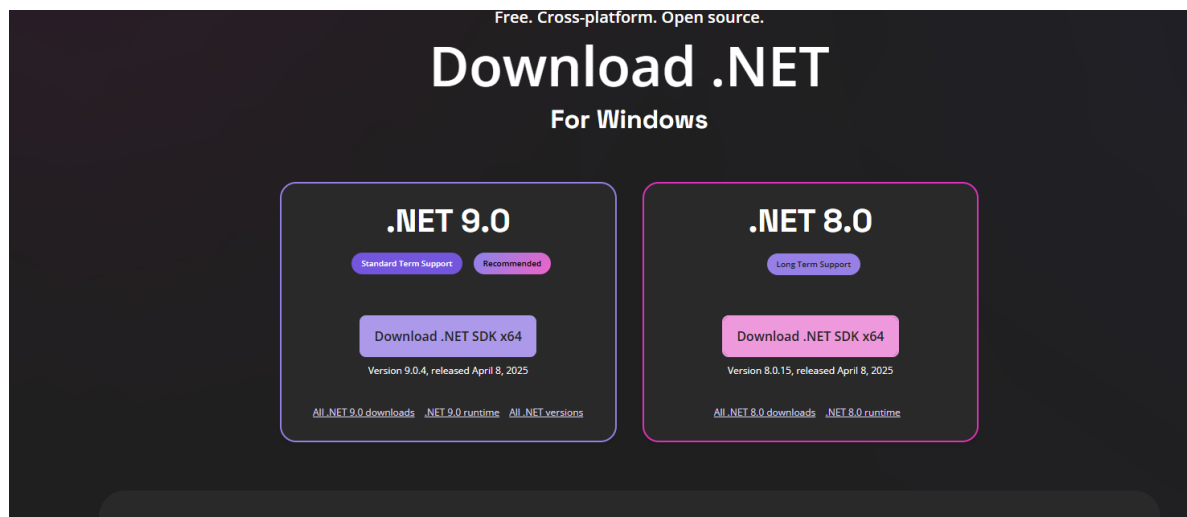


Enlace: <https://learn.microsoft.com/es-es/aspnet/core/host-and-deploy/iis/?view=aspnetcore-9.0>

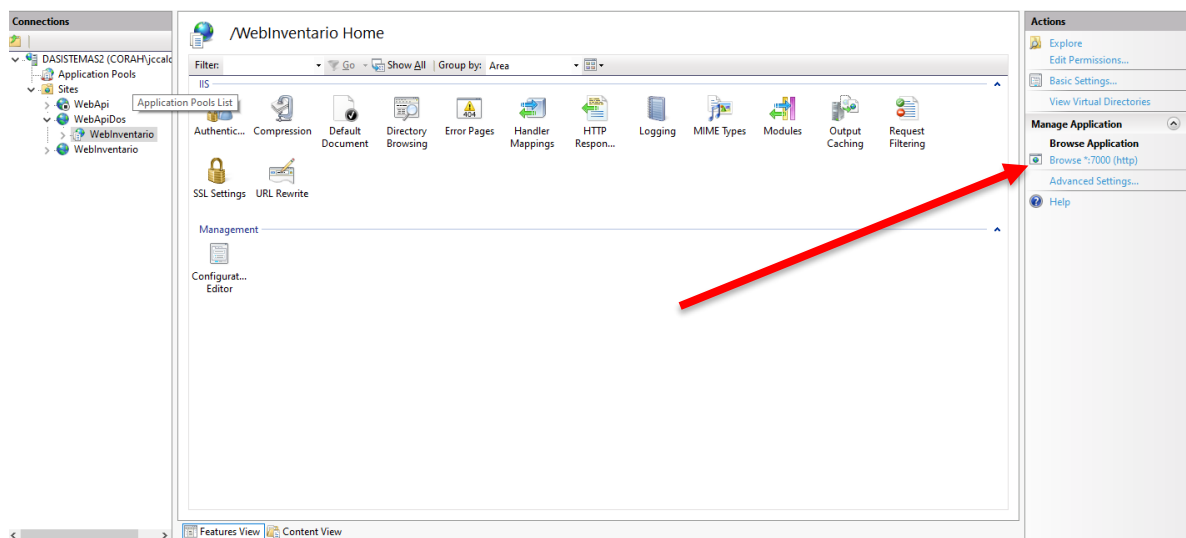


- ✓ Instalación del SDK de .net, la versión que corresponda al proyecto, en nuestro caso sera la versión 8

Enlace: <https://dotnet.microsoft.com/en-us/download>



- ✓ Abrimos la aplicación para comprobar que todo funciona correctamente.



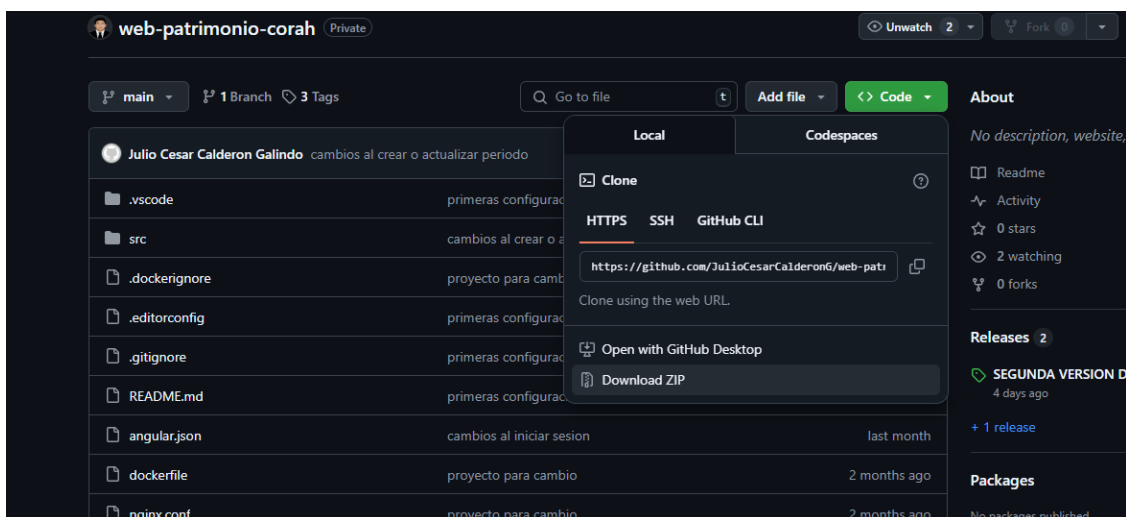
```
localhost:7000/WebInventario/api/periodo

{
  "id": 2009,
  "descripcion": "PERIODO INVENTARIO 2020 - V2",
  "nombreDatos": "20200908_12558",
  "ano": 2020,
  "fechaCreacion": "2020-05-08T11:25:58.833",
  "estadoCargado": true,
  "estadoBackup": false,
  "estadoCierre": true
},
{
  "id": 2004,
  "descripcion": "PRUEBA 2",
  "nombreDatos": "20200908_111749",
  "ano": 2020,
  "fechaCreacion": "2020-05-08T11:17:49.883",
  "estadoCargado": false,
  "estadoBackup": false,
  "estadoCierre": false
},
{
  "id": 2003,
  "descripcion": "PERIODO INVENTARIO 2020",
  "nombreDatos": "20200908_111710",
  "ano": 2020,
  "fechaCreacion": "2020-05-08T11:17:10.540",
  "estadoCargado": false,
  "estadoBackup": false,
  "estadoCierre": false
},
{
  "id": 2002,
  "descripcion": "PERIODO INVENTARIO 2020 - V4",
  "nombreDatos": "20200908_104508",
  "ano": 2020,
  "fechaCreacion": "2020-05-08T10:45:02.473",
  "estadoCargado": true,
  "estadoBackup": false,
  "estadoCierre": false
},
{
  "id": 1002,
  "descripcion": "PERIODO INVENTARIO 2020 - V2",
  "nombreDatos": "Inventario2020v2",
  "ano": 2020,
  "fechaCreacion": "2020-04-29T11:02:17.363",
  "estadoCargado": true,
  "estadoBackup": false
}
}
```

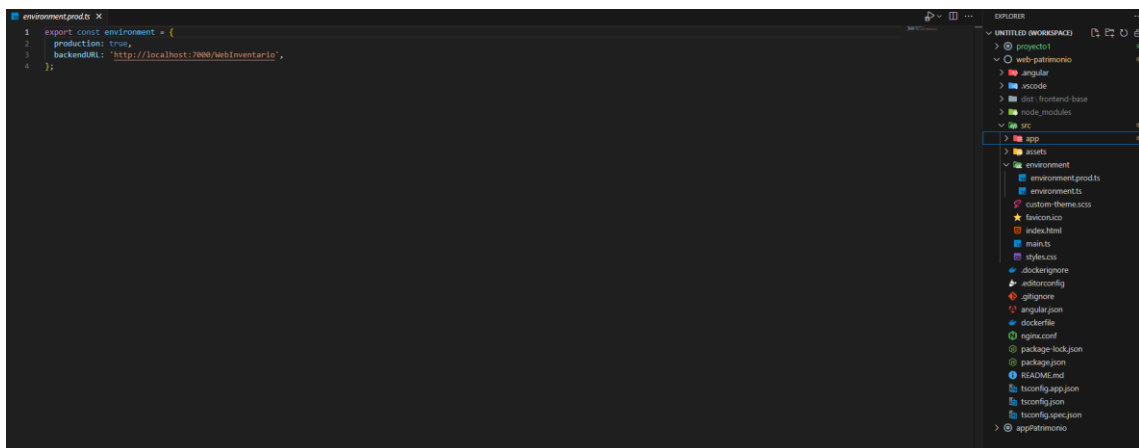
Configuración de la Pagina Web.

La opción de Internet Information Service se utilizará para configurar el frontend del sistema de Toma de Inventario.

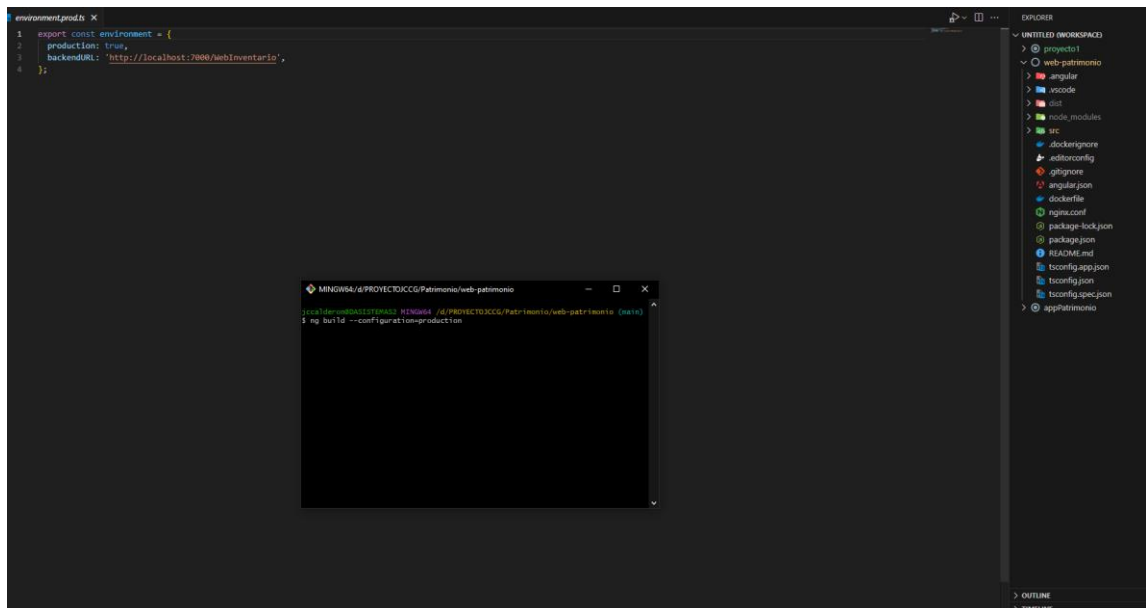
- ✓ Descargamos el proyecto de angular que esta alojado en GitHub.



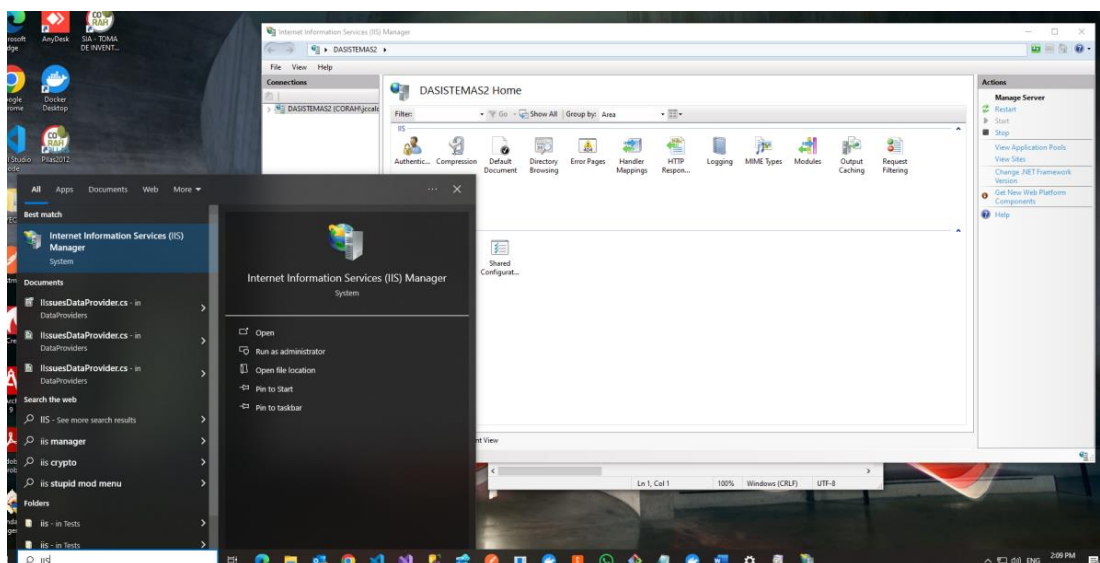
- ✓ Ingresamos a la variable de entorno del proyecto y verificamos que el enlace del backend sea el correcto.



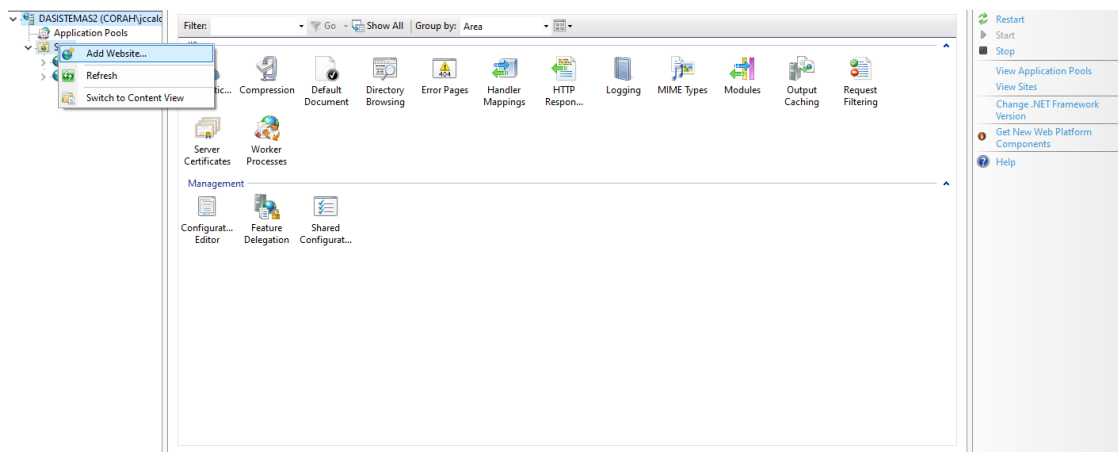
- ✓ Creamos la build del proyecto de angular, abrimos un CMD o GitBash y ejecutamos el comando “ng build --configuration=production” en la raíz del proyecto, este creara la carpeta **.dist/nombre-proyecto**, tendremos que copiar los archivos que están dentro de esa ubicación en otra carpeta fuera del proyecto de angular.



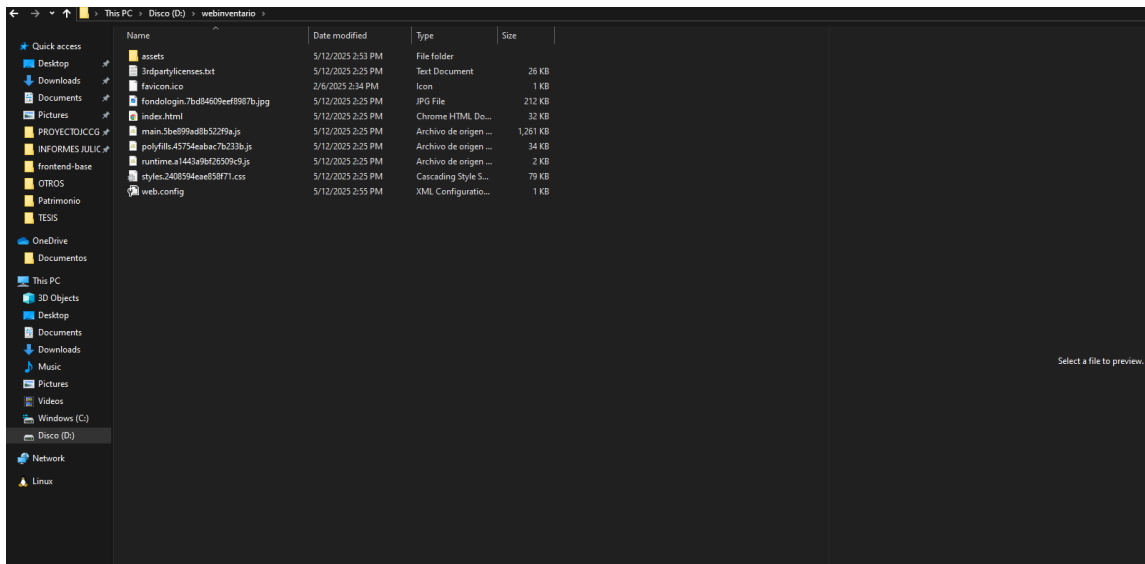
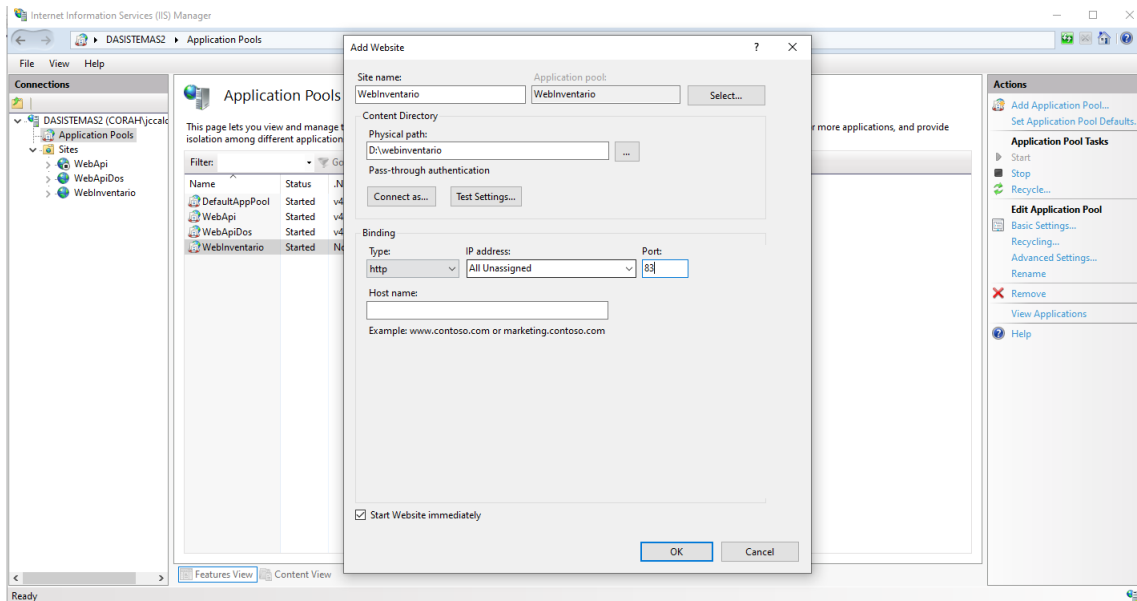
- ✓ Buscamos Internet Information Service y damos click en él, este nos abrirá la interfaz gráfica, se adjunta imagen de referencia.



- ✓ Damos click derecho en sites y seleccionamos Add Website, se adjunta imagen de referencia.



- ✓ Completamos los datos que nos solicita, la carpeta que seleccionemos deberá contar con la build del proyecto de angular se adjunta imagen de referencia.



- ✓ Para que las rutas del sistema de toma de inventario funcione correctamente debemos instalar la extensión de **URL Rewrite**, enlace: <https://www.iis.net/downloads/microsoft/url-rewrite>.
- ✓ En la carpeta donde hemos configurado nuestro proyecto crearemos un archivo web.config, con la siguiente información.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<configuration>
```

```
<system.webServer>
```

```
<rewrite>
```

```
<rules>
```

```
<rule name="Angular Routes" stopProcessing="true">
```

```
<match url=".*" />
```

```
<conditions logicalGrouping="MatchAll">
```

```
<add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
```

```
<add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
```

```
</conditions>
```

```
<action type="Rewrite" url="/index.html" />
```

```

</rule>

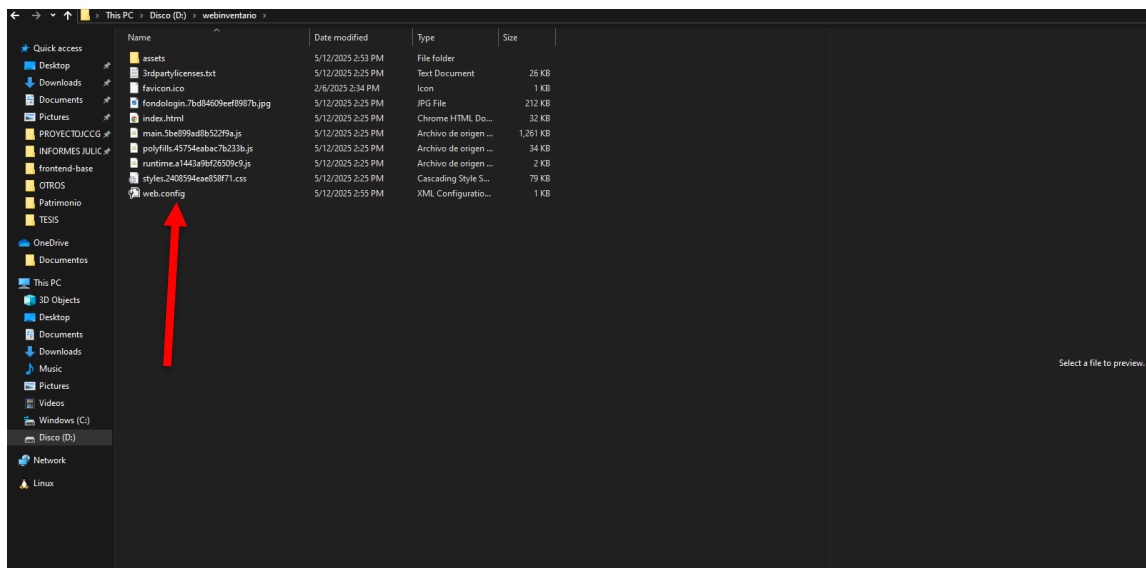
</rules>

</rewrite>

</system.webServer>

</configuration>

```



✓ Abriremos la aplicación para corroborar que todo esté funcionando correctamente.

