

Instalacion de ambiente de trabajo

Introduccion [↗](#)

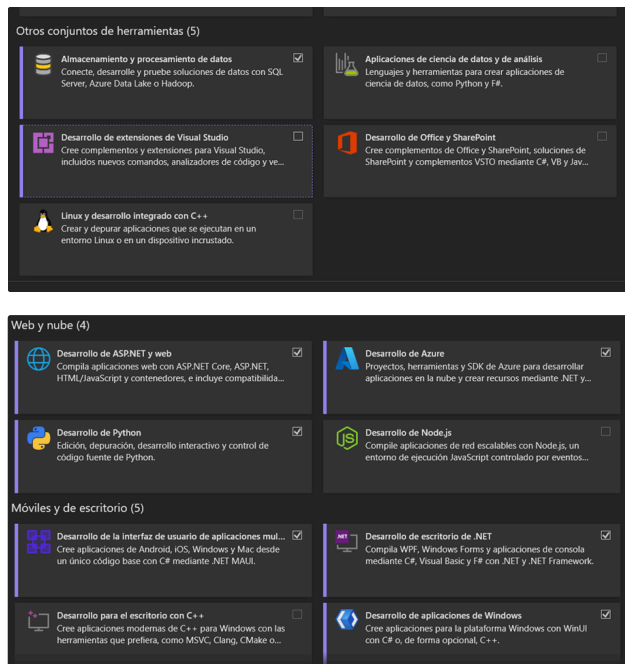
Este documento proporciona una guía detallada para la instalación y configuración del entorno de trabajo para el proyecto para Bravo Izquierdo, el proyecto consiste en el desarrollo de una aplicación web que consiste en un proyecto modelo de capas y otro en modelo vista controlador por lo que se utilizarán las siguientes herramientas/programas para el desarrollo web.

Programas necesarios - instalacion y configuracion [↗](#)

Visual Studio community 2022 [↗](#)

Ejecutar el archivo .exe que se descarga a través de la página e instalar de manera normal, luego se nos llevará al apartado de Visual Studio installer donde deberemos seleccionar Visual Studio Community 2022 y seleccionamos instalar.

Ahora se nos llevará al siguiente paso que consiste en instalar las herramientas de desarrollo que tendrá nuestro Vs 2022, son las siguientes:



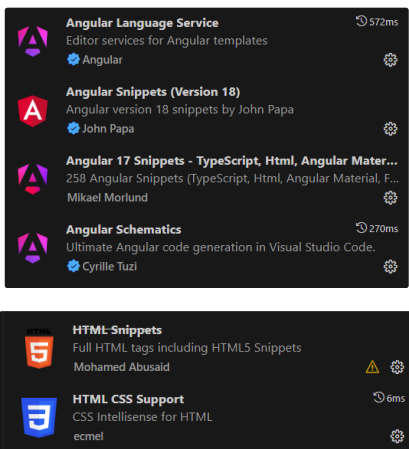
Ya seleccionadas las herramientas continuamos con la instalación, con eso termina la instalación de Vs community 2022.

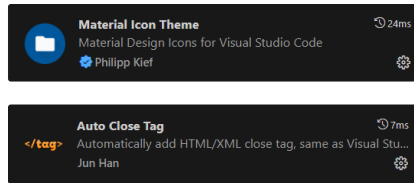
En esta plataforma se verá

- Desarrollo de ASP.NET y web.
- Desarrollo de aplicaciones .NET Core.

Visual studio code [↗](#)

Entramos a la página de Vs code que está enlazada arriba y descargamos la versión necesaria para nuestro sistema operativo, se ejecuta el archivo .exe y se siguen los pasos de la instalación no se requiere hacer nada más. Ya dentro de Visual Studio code serán requeridas las siguientes extensiones:



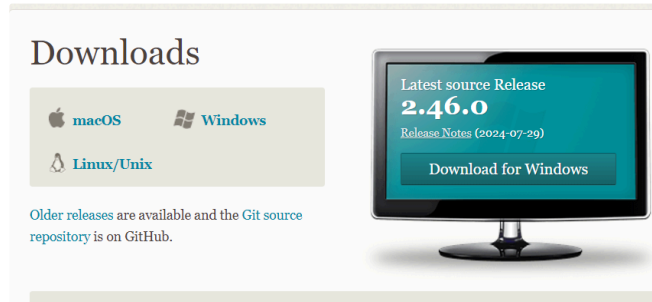


NodeJS [↗](#)

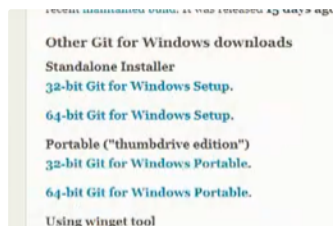
Entramos a la pagina y le damos a descargar Node.js (LTS). Instalamos normalmente Node.js y rechazamos herramientas las que ofrece solo una instalación normal. Si ocurre un error por las versiones desactualizadas o incompatibilidad [revisar aquí](#).

Instalación GIT [↗](#)

Instalamos la versión más reciente:



Seleccionamos la versión standalone dependiendo si nuestro sistema es de 64-bit o 32-bit



Y seguimos la instalación normal.

Angular [↗](#)

La instalación de Angular se realiza de la siguiente manera abrimos CMD como administrador y escribimos los siguientes comandos para la instalación de la versión más reciente de Angular.

1. En el cmd ejecutamos el siguiente comando:

```
1 npm install -g @angular/cli
```

```
C:\Windows\System32>npm install -g @angular/cli
added 225 packages in 23s
41 packages are looking for funding
  run 'npm fund' for details
C:\Windows\System32>ng --version
18.0.2
```

2. Revisamos la versión que se instaló con el siguiente comando:

```
1 ng version
```

```
C:\Windows\System32>ng version

Angular CLI
Angular CLI: 18.0.2
Node: 20.14.0
Package Manager: npm 10.8.1
OS: win32 x64

Angular:
+---+
|   |
+---+

Package                                  Version
-----
@angular-devkit/architect                0.1800.2 (cli-only)
@angular-devkit/core                     18.0.2 (cli-only)
@angular-devkit/schematics               18.0.2 (cli-only)
@schematics/angular                     18.0.2 (cli-only)
```

Generar nuevo proyecto en angular [↗](#)

```
1
2 ng new nameapp --no-standalone --routing --ssr=false
```

Si hay algun problema de incompatibilidad de versiones de angular por tener una version especifica [revisar aqui](#).

Hay que tener precaucion ya que nuestro proyecto usara app.module y las versiones posterires a angular 17 no vienen con este conteido necesario [revisar aqui](#) para ver como se previno.

Instalacion de angular material para nuestro proyecto ya creado no situamos en el y a travez de una consola de comandos puede ser tanto por Vs code, navegar por CMD hasta la ubicacion o abrir una consola de comandos desde la carpeta del proyecto ejecutamos el siguiente codigo,

```
1 ng add @angular/material
```

```
C:\Proyecto_HRD_Bravo_Izquierdo_constructora\HRD_BI_Frontend>ng add @angular/material
i Using package manager: npm
✓ Found compatible package version: @angular/material@18.0.1.
✓ Package information loaded.

The package @angular/material@18.0.1 will be installed and executed.
Would you like to proceed? (Y/n) y
```

```
The package @angular/material@18.0.1 will be installed and executed.
Would you like to proceed? Yes
✓ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: (Use arrow keys)
> Azure/Blue      [Preview: https://material.angular.io/theme=azure-blue]
  Rose/Red        [Preview: https://material.angular.io/theme=rose-red]
  Magenta/Violet  [Preview: https://material.angular.io/theme=magenta-violet]
  Cyan/Orange     [Preview: https://material.angular.io/theme=cyan-orange]
  Custom
```

```
The package @angular/material@18.0.1 will be installed and executed.
Would you like to proceed? Yes
✓ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Rose/Red
? Set up global Angular Material typography styles? (Y/n) y
```

```
The package @angular/material@18.0.1 will be installed and executed.
Would you like to proceed? Yes
✓ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Rose/Red
? Set up global Angular Material typography styles? Yes
? Include the Angular animations module? (Use arrow keys)
> Include and enable animations
  Include, but disable animations
  Do not include
```

```
✓ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Rose/Red
? Set up global Angular Material typography styles? Yes
? Include the Angular animations module? Include and enable animations
UPDATE package.json (1112 bytes)
✓ Packages installed successfully.
UPDATE src/app/app.module.ts (651 bytes)
UPDATE angular.json (3157 bytes)
UPDATE src/index.html (535 bytes)
UPDATE src/styles.css (182 bytes)

C:\Proyecto_HRD_Bravo_Izquierdo_constructora\HRD_BI_Frontend>
```

Por si no esta instalado (es para poder ejecutar los comandos ng en angular ejemplo ng s)

```
1 npm install
```

Generar un componente ne angular y otros comandos de utilidad: [↗](#)

Crear un componente sin archivo specs (el archivo specs es para pruebas y no es necesario)

```

1 ng generate component nombre-del-componente --skip-tests
2
3 Tambien se puede abreviar
4 ng g c nombre-del-componente --skip-tests
5

```

SQL server 20 [↗](#)

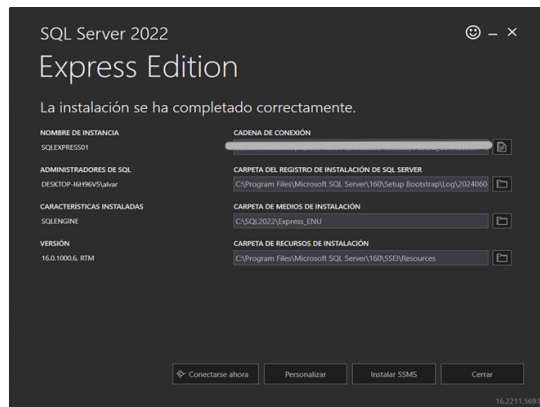
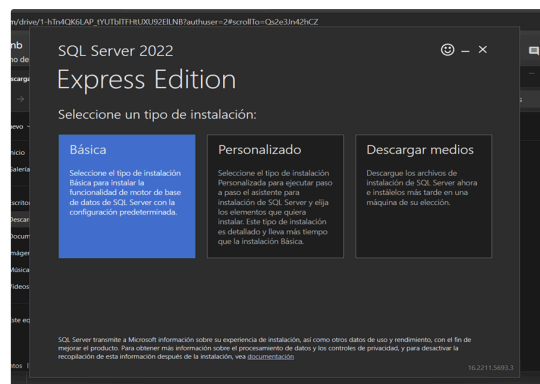
Entramos en la pagina enlazada y seleccionamos la version Express de SQL server. Luego ejecutamos el archivo que se nos descargo y seleccionamos instalacion basica.



Express

SQL Server 2022 Express es una edición gratuita de SQL Server, que es ideal para el desarrollo y la producción, para aplicaciones de escritorio, Internet y pequeños servidores.

[Descargar ahora](#)



nota: en esta pestaña al final de la instalacion se nos dabara un link para el siguiente paso que es instalar SSMS y a deas se jos dara la cadena de conexion, pero el servidor que utilizaremos para hacer la conexion de la base de datos con nuestro proyecto modelo de por capaz en Vs community aparecera en el siguiente al finalizar el siguiente paso en el SSMS.

SQL Management Studio 20 [↗](#)

Debido a como esta conformada la pagina de microsoft es un poco complicado encontrar el link de descarga para SSMS 20. Le daremos click en descargar e iniciara la bajada del ejecutable, ya descargado se hace una ejecucion e instalacion normal.

Descarga de SSMS

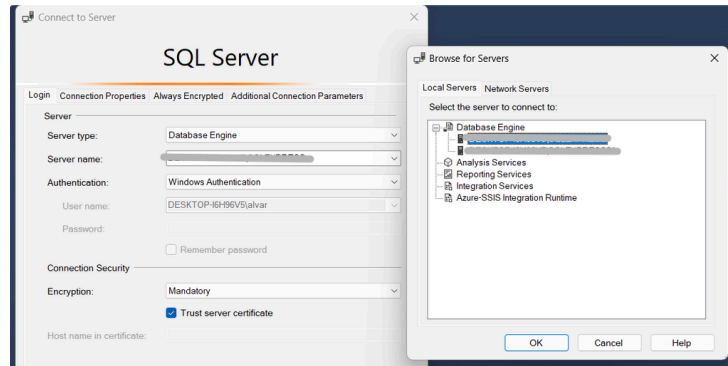
[Descargar SQL Server Management Studio \(SSMS\) 20.2](#)

SSMS 20.2 es la última versión de disponibilidad general (GA). Si tiene instalada una *versión preliminar* de SSMS 20, desinstálela antes de instalar SSMS 20.2. La instalación de SSMS 20.2 no actualiza ni reemplaza las versiones 19.x de SSMS ni las anteriores.

- Número de versión: 20.2
- Número de compilación: 20.2.30.0
- Fecha de publicación: 9 de julio de 2024

Ya instalado el SSMS se debe establecer la conexión de nuestro proyecto modelo por capaz a la base de datos a través de una cadena de conexión. A continuación se mostrará el nombre del servidor y la DataBase engine que se usarán.

Seleccionamos browse for more en el apartado de server name y podremos escoger la Database engine con la que trabajaremos. en este caso será así.



Se creó una base de datos de prueba para el uso remoto y no en servidor, la base de datos se compartirá a través de un script que tiene el contenido para crear la base de datos, crear las tablas y poblar las tablas.

SEGUIR ADJUNTANDO PROCESO DE LINKEO DE BASE DE DATOS A VS COMMUNITY

Lenguajes de programación

Los lenguajes de programación necesarios para el desarrollo de la aplicación web (HRD) son los siguientes.

En el IDE de Visual Studio code se utilizará:

- HTML
- CSS
- TS

En el IDE de Visual Studio Community 2022

- C#

Para las bases de datos en SQL Management studio

- SQL

GitHub comandos

1. Inicializa el Git en la carpeta del proyecto se ejecuta en la consola de PowerShell:

```
1 git init
```

2. El URL es el del repositorio:

```
1 git remote add origin https://github.com/Ophanim666/HDR_frontend-responsive.git
```

3. Descargará el contenido remoto sin modificar el estado del repositorio local:

```
1 git fetch origin
```

4. Configuración git inicial:

```
1 git checkout -b main origin/main
2 git config --global user.email "iv.diazl@duocuc.cl"
3 git config --global user.name "ivan diaz"
```

5. Para hacer un commit:

```
1 git commit -m "Guardar cambios antes de cambiar a la rama main"
```

Si salta un error de historias no relacionadas se [resuelve así](#).

6. Para juntar la rama que uno crea con el main en la rama nueva que uno creó:

```
1 git merge main
```

7. Para enviar al remoto se usa:

```
1 git push origin <nombre-de-la-rama>
```

8. Para consultar las ramas en el remoto (la página de GitHub):

```
1 git ls-remote --heads origin
```

9. Para actualizar/traer los cambios del remoto al local:

```
1 git pull
```

10. Consultar nombre de la rama actual en la que se encuentra

```
1 git branch --show-current
```

11. Ver lista de ramas (este solo mostrara las ramas locales)

```
1 git branch
```

Si también quieres ver las ramas remotas junto con las locales, puedes usar:

```
1 git branch -a
```

12. Traer ramas loc

```
1 git fetch --all
2 git branch -r
3 --ESTE ES PARA CAMBIARTE DE RAMA A LA QUE QUIERAS--
4 git checkout <nombre-de-la-rama-remota>
```

Si ya tienes una rama remota y quieres crear una rama local a partir de ella, puedes usar:

```
1 git checkout -b <nombre-de-la-rama-local> origin/<nombre-de-la-rama-remota>
```

Ver las ramas locales y remotas:

```
1 git branch --all
```

13. Cambiar de rama (para cambiar de rama se pueden utilizar 2 métodos el checkout y el switch)

```
1 git checkout "nombre-de-la-rama"
```

```
1 git switch "nombre-de-la-rama"
```

14. Trae la rama a tu repositorio local:

```
1 git checkout -b nombre-de-la-rama origin/nombre-de-la-rama
```

```
1 git pull
```

15. Guardar tus cambios de local a remota

a. **Asegúrate de estar en la rama local correcta:**

```
1 git checkout nombre-de-la-rama
```

b. **Confirma que todos tus cambios están comprometidos:** Si tienes cambios sin comprometer, primero debes agregarlos y comprometerlos.

```
1 git add .
2 git commit -m "Descripción de los cambios"
```

c. **Sube los cambios a la rama remota:** Ahora, puedes empujar tus cambios a la rama remota. Esto actualizará la rama remota con los cambios realizados en tu rama local.

```
1 git push origin nombre-de-la-rama
```

d. **Verifica la actualización:** Puedes verificar que la rama remota ha sido actualizada correctamente ejecutando:

```
1 git fetch origin
2 git log origin/nombre-de-la-rama
```

e. **Si ya habías configurado un seguimiento** (es decir, si usaste `git push -u origin nombre-de-la-rama` anteriormente), puedes simplemente hacer:

```
1 git push
```

16. Para crear una rama lo hacemos desde la rama main y siempre ejecutamos un pull antes de crear una rama y nos cambiamos con un checkout y con un push creamos la rama en el remoto.

Para crear una nueva rama en tu repositorio local puedes utilizar el comando "git checkout -b" seguido del nombre de la rama que se desea crear.

```
1 git checkout -b nombre_de_la_rama
```

Después de ejecutar este comando, estarás en la nueva rama y podrás comenzar a trabajar en ella. Puedes agregar y confirmar cambios como en cualquier otra rama y si quieres subir esta rama al repositorio remoto después de realizar cambios puedes hacerlo con el comando "git push".

Este código es para enviar la rama al remoto

```
1 git push --set-upstream origin feat/GestionTipoParametros
```

```
1 git push
```

14. Eliminar ramas

a. **Eliminar una rama fusionada:** Si la rama que deseas eliminar ya ha sido fusionada con otra rama (por ejemplo, `main` o `master`), puedes usar:

```
1 git branch -d nombre-de-la-rama
```

Esto eliminará la rama, pero si la rama tiene cambios que no han sido fusionados, Git te advertirá y no la eliminará.

b. **Eliminar una rama no fusionada:** Si estás seguro de que deseas eliminar una rama que no ha sido fusionada (y perder cualquier trabajo que no esté guardado en otra rama), usa:

```
1 git branch -D nombre-de-la-rama
```

Esto forzará la eliminación de la rama, sin importar si tiene cambios no fusionados.

c. **Nota:** Estos comandos eliminan ramas locales. Si necesitas eliminar una rama remota, puedes usar:

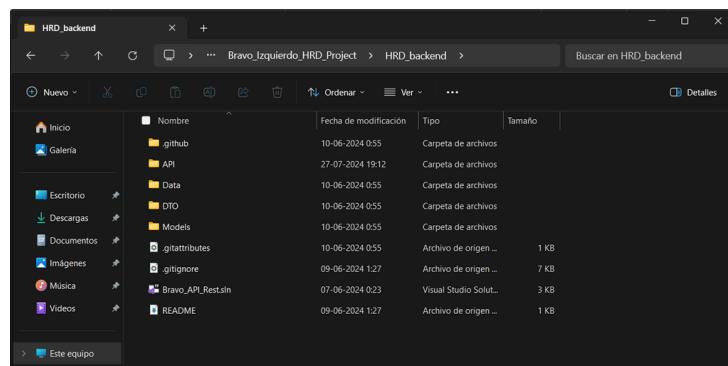
```
1 git push origin --delete nombre-de-la-rama
```

Este comando eliminará la rama del repositorio remoto.

Github en Visual Studio Community 2022

Los comandos de github que vimos anteriormente funcionan de igual manera pero tendremos que ejecutarlos desde un CMD desde la carpeta donde está nuestro proyecto. Por ejemplo:

Desde la carpeta de nuestro proyecto podemos escribir en el navegador de archivos o donde se encuentra la ruta y tipeamos CMD y se abrirá la consola o hacemos click derecho y seleccionamos abrir terminal.



Del modo que se utilizó para poder ubicarse donde está nuestro proyecto podremos empezar a usar los comandos de github. (esta forma de consultar la rama del proyecto en la que estamos trabajando podría resultar más fácil ya que la interfaz de Visual Studio resulta un poco confusa al principio).

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Bravo_Izquierdo_HRD_Project\HRD_backend>
```

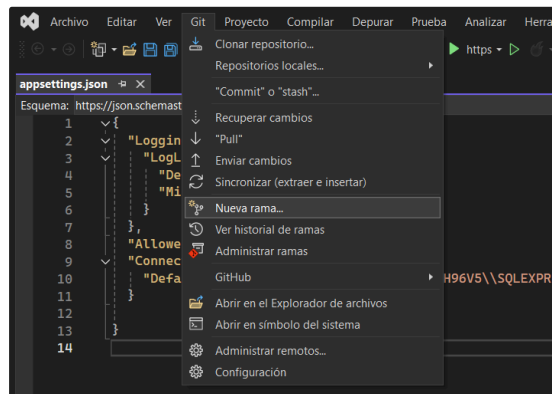
```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Bravo_Izquierdo_HRD_Project\HRD_backend>git branch --show-current
RamaDePueba2

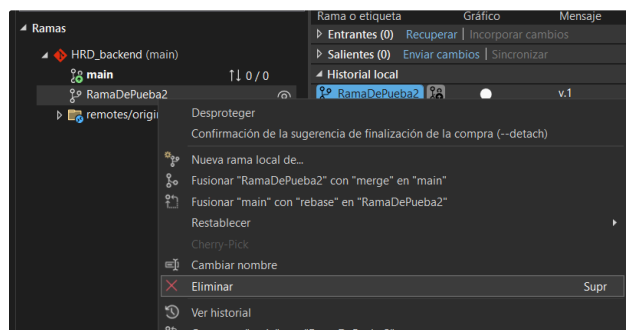
C:\Bravo_Izquierdo_HRD_Project\HRD_backend>git branch
* RamaDePueba2
  Rama_de_Prueba
  main

C:\Bravo_Izquierdo_HRD_Project\HRD_backend>
```

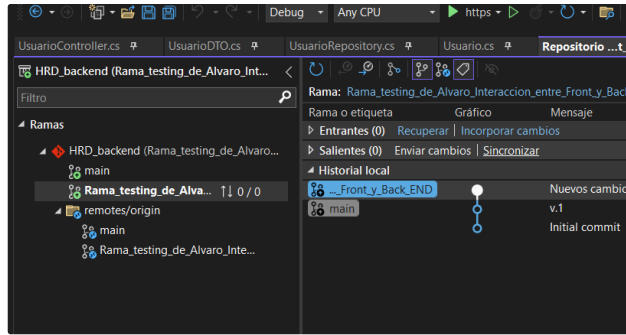
Podemos crear ramas y hacer muchas cosas mas por ahora se recomienda hacer uso tanto de los comandos como la interfaz ya que hay algunas cosas que la interfaz dificulta o imposibilita que los comandos son capaces de hacer.



Nota: para eliminar una rama en la interfaz de VS community se debe cambiar de rama



Para tener los cambios de una rama en tu repositorio es necesario darle a sincronizar en el apartado de salientes



Problemas comunes [↗](#)

Incompatibilidad de versiones de angular [↗](#)

Si acontece un problema de incompatibilidad por versiones de angular realizar lo siguiente:

1. Actualizar Angular CLI:

```
1 npm install --save-dev @angular/cli@18.0.2
```

2. Actualizar Angular Core y Paquetes Relacionados:

```
1 ng update @angular/core@11 @angular/cli@18.0.2
```

3. Revisar y Ajustar Dependencias en package.json:

```
1 "dependencies": {  
2   "@angular/animations": "18.0.2",  
3   "@angular/common": "18.0.2",  
4   "@angular/compiler": "18.0.2",  
5   "@angular/core": "18.0.2",  
6   "@angular/forms": "18.0.2",  
7   "@angular/platform-browser": "18.0.2",  
8   "@angular/platform-browser-dynamic": "18.0.2",  
9   "@angular/router": "18.0.2",  
10  ...  
11 }
```

4. Instalar Dependencias:

```
1 npm install
```

5. Verificar la Versión de Angular:

```
1 ng version
```

Angular sin app.modules [↗](#)

No debería ser un inconveniente tan grande ya que la aplicación se compartió a través de repositorios de Github pero para solucionar el problema de que las aplicaciones de Angular posteriores a la versión 17 ya no incluyen el `app.modules` se debe ejecutar este comando a la hora de crear un proyecto nuevo, así es como se soluciona la carencia de estos módulos necesarios para la creación de nuestra app web.

```
1 ng new Nombre_Del_Proyecto --standalone=false
```

Incompatibilidad de versiones en nodeJS [↗](#)

En NodeJS al meternos en la página nos descargará la versión más nueva de NodeJS a la fecha del desarrollo de esta documentación sería la versión **v20.16.0**. Si llega a ocurrir un conflicto de versiones ya que el desarrollo de la app web en Angular se realizó cuando la versión de NodeJS era **v20.14.0** realizar lo siguiente.

1. Abrimos un CMD (Command Prompt/Símbolo del sistema) como administrador.
2. Verificamos versión de NodeJS en CMD:

```
1 node --version
```

```
C:\Windows\System32>node --version  
v20.14.0
```

3. Instalar versión deseada y establecerla para su uso:

```
1 nvm install 20.14.0  
2 nvm use 20.14.0
```

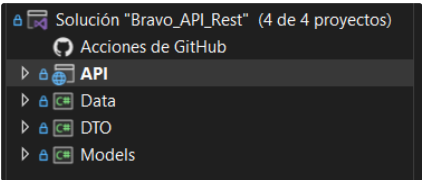
Error de historias no relacionadas en Github

Con el error de historias no relacionadas se resuelve con:

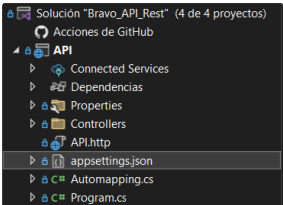
```
1 git merge main --allow-unrelated-histories
2 git commit -m "ivan"
```

Conexion de base de datos (explicar como se conecta la base de datos)

En otro apartado del proyecto se explicara a mejor detalle como se crea la solucion de Vs community pero nos deberia quedar una arquitectura asi.

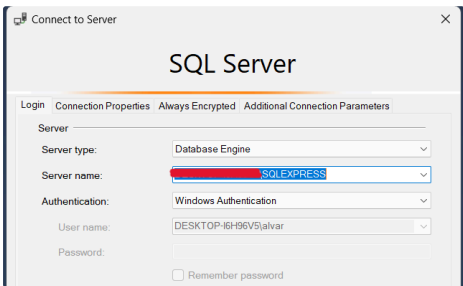


Son 4 proyectos en total, el que tiene el nombre de API corresponde a un proyecto de ASAP.net web api y los demas 3 son bibliotecas de datos y todos estos interactuarian entre si, en el apartado que nos centraremos sera en el el proyecto API el cual contendra lo siguiente.



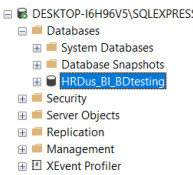
En el archivo **appsettings.json** centraremos nuestraatencion ya que ahi es donde crearemos la cadena de conexion entre nuestra solucion y nuestra base de datos que manejaremos con SQL management studio.

Antes que nada deberemos ir a la conexion que estamos usando en SQL management studio y copiar el server name donde estara la base de datos que utilizaremos.

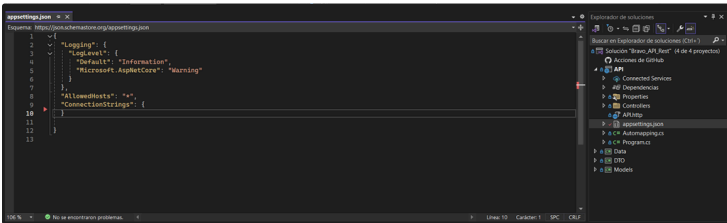


Copiamos el server name y vamos a nuestra solucion de Vs community en el archivo **appsettings.json** y lo editamos. El codigo deberia tener un apartado de ConectionString el cual estara vacio y es ahi donde nosotros añadiremos nuestra cadena de conexion junto al nombre de la base de datos.

A continuacion se muestra la base de datos de prueba que estaremos utilizando:



Ya teniendo estos datos podemos desarrollar nuestra conection string dentro de **appsettings.json** el cual deberia estar asi.



Ahora agregaremos la conexión de la siguiente forma.

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "ConnectionStrings": {
10    "DefaultConnection": "Server=el nombre extraído de servername de Sql manage\\SQLEXPRESS;Database=HRDus_BI_BDtesting;TrustServerCertificate=True;Trusted_Connection=true;
11  }
12 }
13 }
14 }
```

Y con eso se finaliza la conexión de nuestra solución a la base de datos.