

 README.md

# BACKEND

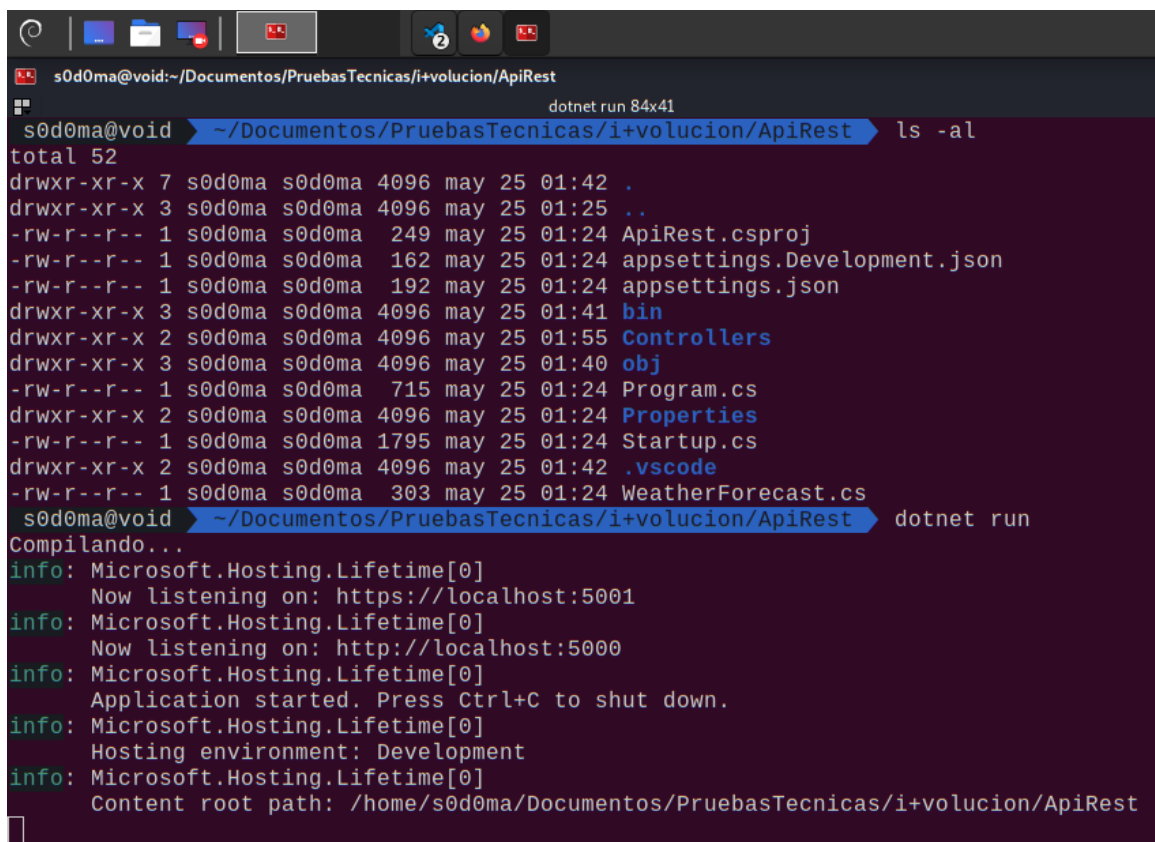
## CREACION DE WEBAPI

comando de creacion de proyecto webapi

```
dotnet new webapi --name ApiRest
```

comando de ejecucion de solucion

```
dotnet run
```



```
s0d0ma@void:~/Documentos/PruebasTecnicas/i+volucion/ApiRest
dotnet run 84x41

s0d0ma@void ~/Documentos/PruebasTecnicas/i+volucion/ApiRest ls -al
total 52
drwxr-xr-x 7 s0d0ma s0d0ma 4096 may 25 01:42 .
drwxr-xr-x 3 s0d0ma s0d0ma 4096 may 25 01:25 ..
-rw-r--r-- 1 s0d0ma s0d0ma 249 may 25 01:24 ApiRest.csproj
-rw-r--r-- 1 s0d0ma s0d0ma 162 may 25 01:24 appsettings.Development.json
-rw-r--r-- 1 s0d0ma s0d0ma 192 may 25 01:24 appsettings.json
drwxr-xr-x 3 s0d0ma s0d0ma 4096 may 25 01:41 bin
drwxr-xr-x 2 s0d0ma s0d0ma 4096 may 25 01:55 Controllers
drwxr-xr-x 3 s0d0ma s0d0ma 4096 may 25 01:40 obj
-rw-r--r-- 1 s0d0ma s0d0ma 715 may 25 01:24 Program.cs
drwxr-xr-x 2 s0d0ma s0d0ma 4096 may 25 01:24 Properties
-rw-r--r-- 1 s0d0ma s0d0ma 1795 may 25 01:24 Startup.cs
drwxr-xr-x 2 s0d0ma s0d0ma 4096 may 25 01:42 .vscode
-rw-r--r-- 1 s0d0ma s0d0ma 303 may 25 01:24 WeatherForecast.cs
s0d0ma@void ~/Documentos/PruebasTecnicas/i+volucion/ApiRest dotnet run
Compilando...
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/s0d0ma/Documentos/PruebasTecnicas/i+volucion/ApiRest
```

## ENDPOINT QUE RETORNA LA HORA COLOMBIANA (GET /HoraCol)

Se obtiene la hora y fecha colombiana en el formato requerido a partir de las zonas horarias del sistema operativo, se valida si la aplicacion se esta ejecutando en windows o linux ya que en ambos sistemas operativos los ids de las zonas horarias se almacenan de manera diferente, en linux en `TimeZoneId /usr/share/zoneinfo/America/Bogota` y en windows se puede ver el listado de timezone con el comando `tzutil -l`. Se realiza esta comprobacion por si la aplicacion se despliega en linux o en windows.

metodo del controlador `Controllers/HoraCol.cs`

```
[HttpGet]
```

```

public ActionResult Get()
{
    string timeZoneId = "";

    // se valida sobre que S0 se esta ejecutando la aplicacion y se
    // asigna el Time Zone Id correspondiente al S0
    // README : https://devblogs.microsoft.com/dotnet/cross-platform-time-zones-with-net-core/
    if(RuntimeInformation.IsOSPlatform(OSPlatform.Linux))
    {
        // PATH TimeZoneId /usr/share/zoneinfo/America/Bogota
        timeZoneId = "America/Bogota";
    }
    else if(RuntimeInformation.IsOSPlatform(OSPlatform.Windows))
    {
        // SA Pacific Standard Time - Bogota, Lima, Quito, Rio Branco
        // https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/default-time-zones
        timeZoneId = "SA Pacific Standard Time";
    }
    else
    {
        return Problem("TimeZoneID no encontrado (win/linux)");
    }

    // Obtener TimeZoneInfo a partir de TimeZoneId
    var coTimeZone = TimeZoneInfo.FindSystemTimeZoneById(timeZoneId);

    // Convertir TimeZoneInfo a DateTime
    DateTime coTime = TimeZoneInfo.ConvertTime(
        DateTime.Now,
        TimeZoneInfo.Local,
        coTimeZone
    );

    // Creacion de objeto anonimo para retonar informacion
    var horaActualColombia = new {
        fecha_colombia = coTime.ToString("dd-mm-yyyy"),
        hora_colombia = coTime.ToString("HH:mm:ss")
    };

    // retornar objeto horaActualColombia
    return Ok(horaActualColombia);
}

```

envio de peticion curl

```

s0d0ma@void ~$ curl -k -X GET "https://localhost:5001/HoraCol" -H "accept: */*" | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0     0     0    0         0         0    0
100    58      0    58      0     0    1288         0 --:--:-- --:--:-- --:--:-- 1288
{
  "fecha_colombia": "25-05-2021",
  "hora_colombia": "03:05:23"
}

```

## ENDPOINT - RECIBE : Num1, Num2 - RETORNA : Num1/Num2 (POST /Division)

Metodo de controlador controllers/Division.cs

```

[HttpPost]
public IActionResult Post(DivisionOperadores operadores)
{

```

```

        if(!ModelState.IsValid)
        {
            return BadRequest();
        }

        return Ok(new { resultado = operadores.Result });
    }

```

Clase custom que recibe el json de los numeros con sus respectivos decoradores para validar la informacion (Decorador NotMapped para excluirlo de la base de datos) Models/CustomModels/DivisionOperadores.cs

```

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ApiRest.Models.CustomModels
{
    [NotMapped]
    public class DivisionOperadores
    {
        [Required(ErrorMessage="Num1 es requerido")]
        public float Num1 { get; set; }

        [Required(ErrorMessage="Num2 es requerido")]
        [RequiredGreaterThanZero(ErrorMessage="Num2 diferente que 0")]
        public float Num2 { get; set; }

        public float Result {
            get { return Num1 / Num2; }
        }
    }

    public class RequiredGreaterThanZero : ValidationAttribute
    {
        public override bool IsValid(object value)
        {
            float i;
            return value != null && float.TryParse(value.ToString(), out i) && i > 0;
        }
    }
}

```

envio de peticion curl

```

s0d0ma@void ~$ curl -X POST "https://localhost:5001/Division" -H "accept: */*" -H "Content-Type: application/json" -d '{"num1":1000,"num2":3}' -k | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0     23  100    22    396    379  --:--:-- --:--:-- --:--:--    775
{
  "resultado": 333.33334
}

```

## CREACION DE BASE DE DATOS VehiculoDb

Se debe crear una base de datos llamada VehiculoDB y con sqlcmd se carga el siguiente script

```
sqlcmd -S localhost -U SA -P 123456 -d VehiculoDB -i Documentos/PruebasTecnicas/i+volucion/database.sql
```

```
use [VehiculoDB];
```

```
-- CREACION DE TABLAS
create table TipoVehiculo(
  TipoVehiculoId int primary key IDENTITY(1,1) NOT NULL,
  NombreTipoVeh varchar(50) NOT NULL,
  Descripcion varchar(50) NOT NULL
);

create table Marca(
  MarcaId int primary key IDENTITY(1,1) NOT NULL,
  Nombre varchar(50) NOT NULL
);

create table Vehiculo(
  VehiculoId int primary key IDENTITY(1,1) NOT NULL,
  Modelo varchar(50) NOT NULL,
  Agno smallint NOT NULL,
  Cilindraje float NOT NULL,
  TipoVehiculoId int NOT NULL,
  MarcaId int NOT NULL
);

-- LLAVES FORANEAS

ALTER TABLE Vehiculo
ADD CONSTRAINT FK_Vehiculo_TipoVehiculo
FOREIGN KEY (TipoVehiculoId) REFERENCES TipoVehiculo(TipoVehiculoId);

ALTER TABLE Vehiculo
ADD CONSTRAINT FK_Vehiculo_Marca
FOREIGN KEY (MarcaId) REFERENCES Marca(MarcaId);

-- REGISTROS INICIALES

INSERT TipoVehiculo(NombreTipoVeh, Descripcion)
VALUES
('Motocicleta','Vehiculo de dos ruedas'),
('Vehiculo Inteligente','Vehiculo controlado por IA'),
('Deportivo','Vehiculo con enfasis en velocidad');

INSERT MARCA(Nombre)
VALUES
('Yamaha'),
('Tesla'),
('Ferrari');

INSERT Vehiculo(Modelo, Agno, Cilindraje, TipoVehiculoId, MarcaId)
VALUES
('ZTX 125', 2010, 125, 1, 1),
('MODEL S', 2008, 670, 2, 2),
('812 Superfast.', 2020, 812, 3, 3);
```

carga base de datos

```
s0d0ma@void ~$ sudo systemctl status mssql-server
● mssql-server.service - Microsoft SQL Server Database Engine
   Loaded: loaded (/lib/systemd/system/mssql-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-05-25 06:46:40 -05; 7min ago
     Docs: https://docs.microsoft.com/en-us/sql/linux
   Main PID: 3578 (sqlservr)
    Tasks: 170
   Memory: 1.5G
      CPU: 25.518s
   CGroup: /system.slice/mssql-server.service
           └─3578 /opt/mssql/bin/sqlservr
             3606 /opt/mssql/bin/sqlservr

may 25 06:52:55 void sqlservr[3606]: [113B blob data]
may 25 06:52:56 void sqlservr[3606]: [139B blob data]
may 25 06:52:56 void sqlservr[3606]: [114B blob data]
may 25 06:53:58 void sqlservr[3606]: [96B blob data]
may 25 06:53:58 void sqlservr[3606]: [76B blob data]
may 25 06:53:58 void sqlservr[3606]: [70B blob data]
may 25 06:53:58 void sqlservr[3606]: [138B blob data]
may 25 06:53:58 void sqlservr[3606]: [113B blob data]
may 25 06:53:58 void sqlservr[3606]: [139B blob data]
may 25 06:53:58 void sqlservr[3606]: [114B blob data]
s0d0ma@void ~$ sqlcmd -S localhost -U SA -P [REDACTED] -d VehiculoDB -i Documentos/PruebasTecnicas/i+volucion/database.sql
Se cambió el contexto de la base de datos a 'VehiculoDB'.

(3 rows affected)

(3 rows affected)

(3 rows affected)
s0d0ma@void ~$
```

instalacion de dependencias:

```
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
```

```
dotnet add package Microsoft.EntityFrameworkCore.Design
```

Creacion de modelos

```
dotnet ef dbcontext scaffold "Server=127.0.0.1;Database=VehiculoDB;ConnectRetryCount=0; User ID=sa;Password=123456"
Microsoft.EntityFrameworkCore.SqlServer -o Models -c "VehiculoDBCTX" -d --force
```

## ENDPOINT RETORNA TODOS LOS VEHICULOS CON SU MARCA Y TIPO DE VEHICULO (GET /VehiculosDB/Listado)

controlador Controllers/Vehiculos.cs

```
// listar todos los vehiculos
[HttpGet("ListaVehiculos")]
public async Task<IActionResult> GetVehiculos()
{
    List<Vehiculo> listaVehiculos = await VehiculoDbCtx.Vehiculos
        .Include("Marca")
        .Include("TipoVehiculo")
        .ToListAsync();

    return Ok(listaVehiculos);
}
```

peticion con curl

```

s0d0ma@void ~/Documentos/PruebasTecnicas/i+volucion/ApiRest curl -k -X GET "https://localhost:5001/Vehiculos/ListaVehiculos" -H "accept: */*" | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left   Speed
100    753    0    753    0     0    4377    0 --:--:-- --:--:-- --:--:--   4377
[
  {
    "vehiculoId": 1,
    "modelo": "ZTX 125",
    "agno": 2010,
    "cilindraje": 125,
    "tipoVehiculoId": 1,
    "marcaId": 1,
    "marca": {
      "marcaId": 1,
      "nombre": "Yamaha"
    },
    "tipoVehiculo": {
      "tipoVehiculoId": 1,
      "nombreTipoVeh": "Motocicleta",
      "descripcion": "Vehiculo de dos ruedas"
    }
  },
  {
    "vehiculoId": 2,
    "modelo": "MODEL S",
    "agno": 2008,
    "cilindraje": 670,
    "tipoVehiculoId": 2,
    "marcaId": 2,
    "marca": {
      "marcaId": 2,
      "nombre": "Tesla"
    },
    "tipoVehiculo": {
      "tipoVehiculoId": 2,
      "nombreTipoVeh": "Vehiculo Inteligente",
      "descripcion": "Vehiculo controlado por IA"
    }
  }
],

```

## ENDPOINT CREACION DE UN NUEVO VEHICULO (POST /VehiculosDB/Creacion)

controlador

```

// crear vehiculo nuevo
[HttpPost("CrearVehiculo")]
public async Task<IActionResult> PostVehiculo(Vehiculo vehiculo)
{
    // verificar que el modelo venga bien
    if(!ModelState.IsValid)
    {
        return BadRequest();
    }

    // obtener marca y tipo de vehiculo para la insercion
    Marca _marca = await VehiculoDbCtx.Marcas.FindAsync(vehiculo.MarcaId);
    TipoVehiculo _tvehiculo = await VehiculoDbCtx.TipoVehiculos.FindAsync(vehiculo.TipoVehiculoId);

    // si no existe marca o tipo de vehiculo muestra mensaje de error
    if( _marca == null || _tvehiculo == null)
    {
        return Ok(new { status = false, msg = "suministre id de marca y tipo vehiculos correctos" });
    }
    else
    {
        // se inserta registro nuevo
        VehiculoDbCtx.Vehiculos.Add(vehiculo);
        int registrosAgregados = await VehiculoDbCtx.SaveChangesAsync();

        // se verifica que se agregara
        if(registrosAgregados > 0)
        {
            // mensaje ok
            return Ok(new {status = true, VehiculoId = vehiculo.VehiculoId});
        }
        else
        {

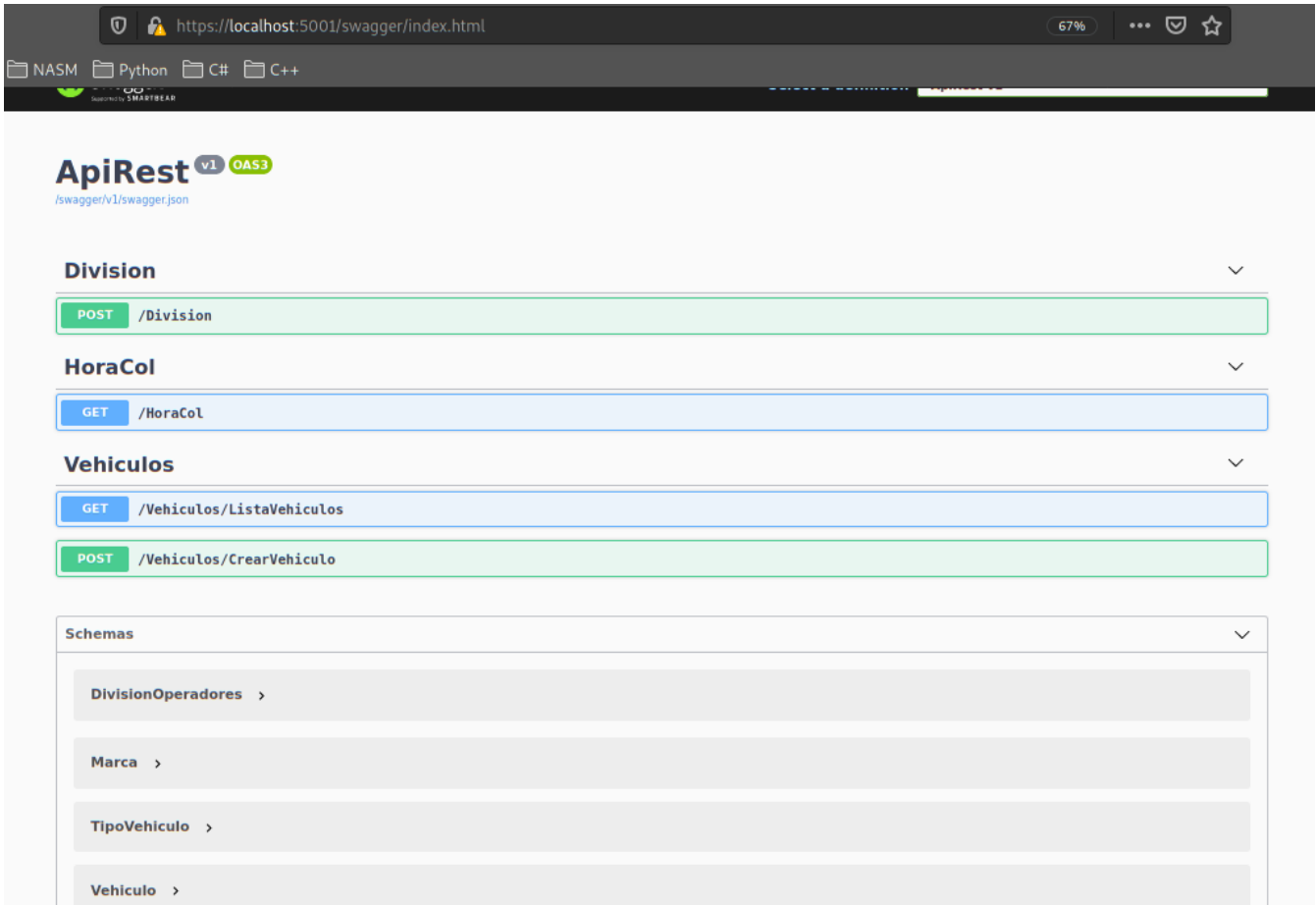
```

```
        // mensaje fail
        return Ok(new {status = false});
    }
}
```

peticion curl

```
s0d0ma@void: ~/Documentos/PruebasTecnicas/i++volucion/ApiRest
curl -k -X POST "https://localhost:5001/Vehiculos/CrearVehiculo" -H "accept: */*" -H "Content-Type: application/json" -d '{"modelo": "Prueba", "agno": 2010, "cilindraje": 9.1, "tipoVehiculoId": 9, "marcaId": 9}' | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total  Spent    Left  Speed
100 153    0 74 100    79      26    28  0:00:02  0:00:02  --:--:--   54
{
  "status": false,
  "msg": "suministre id de marca y tipo vehiculos correctos"
}
s0d0ma@void: ~/Documentos/PruebasTecnicas/i++volucion/ApiRest
curl -k -X POST "https://localhost:5001/Vehiculos/CrearVehiculo" -H "accept: */*" -H "Content-Type: application/json" -d '{"modelo": "Prueba", "agno": 2010, "cilindraje": 9.1, "tipoVehiculoId": 2, "marcaId": 2}' | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total  Spent    Left  Speed
100 109    0 30 100    79      29    77  0:00:01  0:00:01  --:--:--  107
{
  "status": true,
  "vehiculoId": 7
}
```

## Swagger



## FRONTEND

path: FrontEnd exec: ng serve

El frontend fue desarrollado en angular 8 y material. se crearon un componente para visualizar la lista de vehiculos y otro componente para la creacion de vehiculos.

Lista de vehiculos

← → ↺ 🏠

localhost:4200/VehiculosLista

...

🔍 📄 🌙 ☰

EnlacesASP.NETNASMPythonC#C++🔥 Aprende Linux desde ...

Prueba tecnica - Kevin Astroz☰

Lista de vehiculos

Lista de vehiculos

Crear un nuevo vehiculo

ID Vehiculo	Modelo	Cilindraje	Año	Marca	Tipo de vehiculo
1	ZTX 125	125	2010	Yamaha	Motocicleta
2	MODEL S	670	2008	Tesla	Vehiculo Inteligente
3	812 Superfast.	812	2020	Ferrari	Deportivo
4	Prueba	9.1	2010	Yamaha	Motocicleta
5	Prueba	9.1	2010	Tesla	Vehiculo Inteligente
7	Prueba	9.1	2010	Tesla	Vehiculo Inteligente
1002	Prueba	7.1	2222	Tesla	Vehiculo Inteligente
1003	ANGULAR	9.1	2022	Tesla	Vehiculo Inteligente

Creacion de vehiculos

← → ↺ 🏠

localhost:4200/VehiculosCrear

...

🔍 📄 🌙 ☰

EnlacesASP.NETNASMPythonC#C++🔥 Aprende Linux desde ...

Prueba tecnica - Kevin Astroz☰

ModeloCilindrajeAñoTipo de vehiculo

ANGULAR9.120222

Crear vehiculo! 📄

Vehiculo creado

OK

InspectorConsoleDebugger↕ Network{} Style Editor🔄 Performance🔍 Memory📁 Storage👤 Accessibility📦 What's New

Filter Output

ErrorsWarningsLogsInfoDebugCSSXHRRequests⚙

▶ [WDS] Disconnected!client:172

Angular is running in the development mode. Call enableProdMode() to enable the production mode. core.js:38781

ⓘ [WDS] Live Reloading enabled.client:52

creando formulario : ▶ Object { vehiculoId: 0, modelo: "ANGULAR", cilindraje: "9.1", agno: "2022", marcaId: "2", tipoVehiculoId: "2" }vehiculos-crear.component.ts:61:12

added vehiculo : ▼ {\_-}api.service.ts:34:32

status: true

vehiculoId: 1003

▶ <prototype>: Object { \_ }

>>