DOCUMENTACIÓN SISTEMA PERSONAS FÍSICAS

Correr el Sistema

1. **git clone** **https://github.com/edgardan/TEST_DEV_EDSA_27042022.git**
2. o descargar de **https://github.com/edgardan/TEST_DEV_EDSA_27042022**
3. Cargar el script de base de datos adjunto en el repositorio script.sql
4. Modificar las cadenas de conexión de los archivos
   a. API_Rest_DB_Persona_Manager->appsettings.json

```
"ConnectionStrings": {
  "Database": "Server=DESKTOP-TDE2O60\\SQLEXPRESS;Database=TEST_DEV_EDSA_22042022;
},
```

   b. User_Login_MVC->Web.config

```
connection string=&quot;data source=DESKTOP-TDE2O60\SQLEXPRESS;    initial catalog=TEST_DEV_EDSA_22042022;    persist security info=True;    user id=sa;    password=p4sw0rd;
```

5. Ingresar al sistema con usuario: ***edgar.sanchez*** y password: ***123***

Arquitectura

En el Sistema de WEB de Personas Físicas se utilizó la arquitectura mostrada en la imagen 1, dividiendo el proyecto en dos partes: REST API Personas con la funcionalidad de CRUD (Altas bajas, cambios y eliminación) y Sistema WEB Personas Físicas, el cual consume los servicios del REST API de Personas Físicas y del REST API de los clientes para presentar en un reporte.

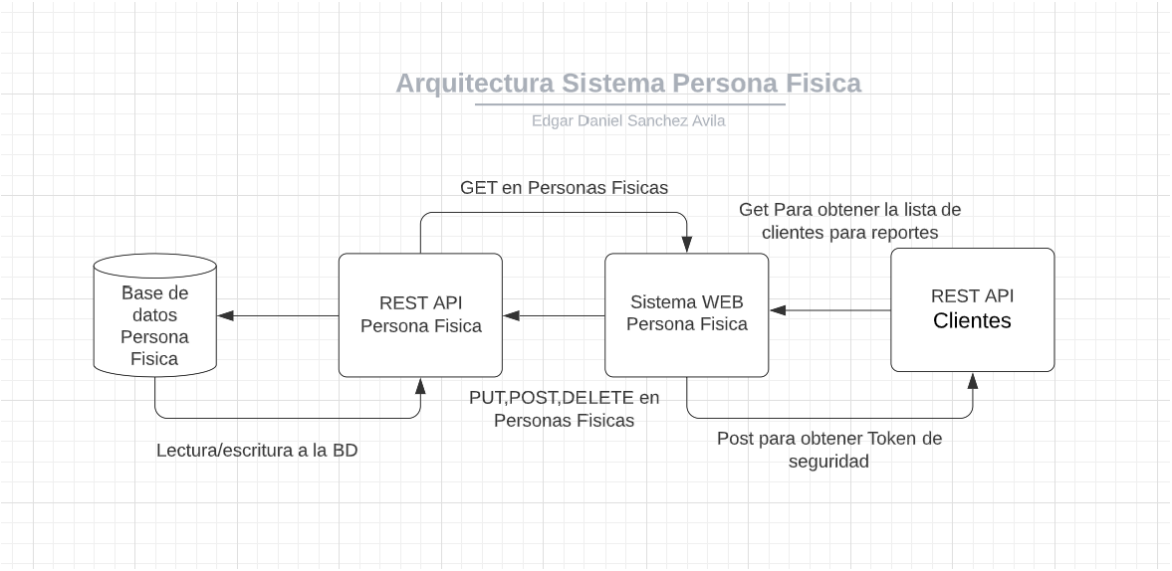


Imagen 1. Arquitectura del Sistema Personas Físicas

Base de Datos

Para crear la base de datos, se corrió el Script proporcionado,

Como parte de las mejoras, se modificaron los store procedures de **sp_EliminarPersonaFisica** y **sp_ActualizarPersonaFisica**, ya que contenían una condición que siempre regresaba que la persona física no existía, cambiando la condición IF **EXISTS por IF NOT EXISTS**

```
BEGIN TRY
    IF NOT EXISTS
    (
      SELECT *
      FROM dbo.Tb_PersonasFisicas
      WHERE IdPersonaFisica = @IdPersonaFisica
          AND Activo = 1
    )
    BEGIN
      SELECT @ERROR = 'La persona física no existe.';
      THROW 50000, @ERROR, 1;
    END;
```

Como parte de las mejoras, también se creó un store produce extra para obtener la lista de personas físicas activas, así como un store procedure para validar los usuarios en el login del sistema WEB.

```
USE [TEST_DEV_EDSA_22042022]
GO
/****** Object:  StoredProcedure [dbo].[Validate_User]    Script Date: 27/04/2022 10:56:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER   PROCEDURE [dbo].[Validate_User]
      @Username NVARCHAR(20),
      @Password NVARCHAR(20)
AS
BEGIN
      SET NOCOUNT ON;
      DECLARE @UserId INT, @LastLoginDate DATETIME

      SELECT @UserId = UserId, @LastLoginDate = LastLoginDate
      FROM Users WHERE Username = @Username AND [Password] = @Password

      IF @UserId IS NOT NULL
      BEGIN
            IF NOT EXISTS(SELECT UserId FROM UserActivation WHERE UserId = @UserId)
            BEGIN
                  UPDATE Users
                  SET LastLoginDate = GETDATE()
                  WHERE UserId = @UserId
                  SELECT @UserId [UserId] -- User Valid
            END
            ELSE
            BEGIN
                  SELECT -2 -- User not activated.
            END
      END
      ELSE
      BEGIN
            SELECT -1 -- User invalid.
      END
END
```

REST API Persona Física

La estructura del REST API de persona física se creó con una arquitectura MVC, se compone de un Controlador que contiene los endspoints, de un paquete de servicios compuesto de una interface (Interface Personas Físicas) para el CRUD, así como la clase Personas Físicas Services SQL, que contiene toda la lógica de negocio, se creó un Contexto para la BD (Core DBContext) que contiene la conexión, así como un DTO (Data Transfer Object) Persona Física el cual es persistido en la BD, estos elementos interactúan entre ellas, como se muestra en la imagen 2.

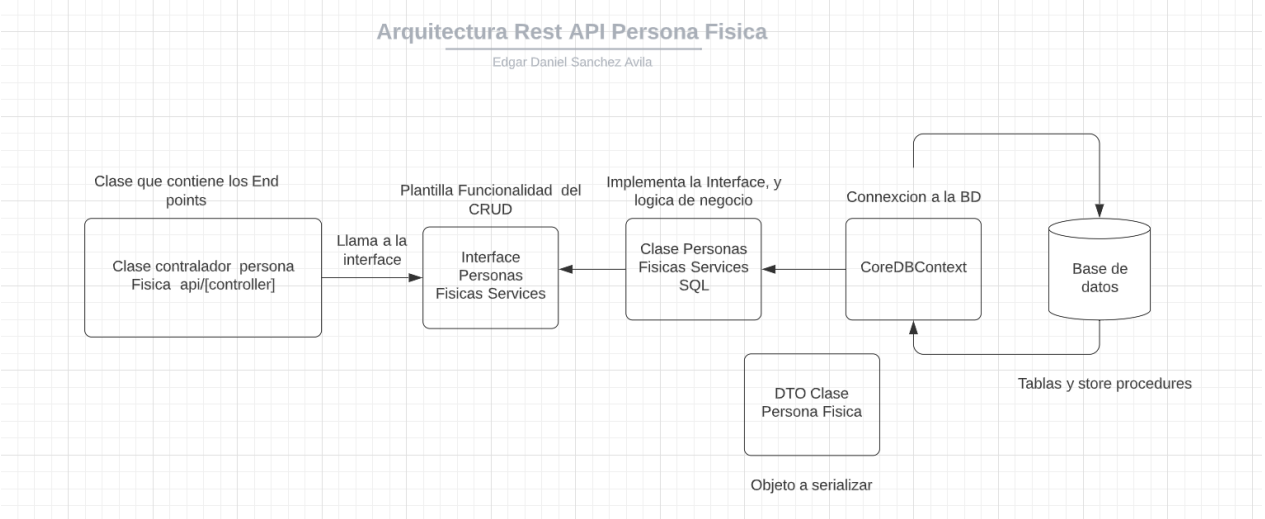
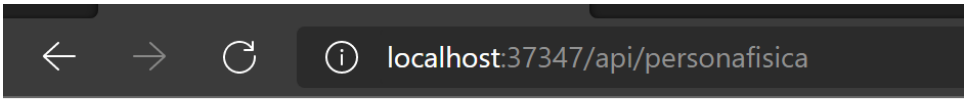


Imagen 2. Arquitectura de la REST API

Los siguientes endpoints fueron desarrollados para hacer el CRUD (CREATE, READ, UPDATE, DELETE) en personas físicas.

Endpoint de lectura

GET localhost:37347/api/personafisica



[ ]

Endpoint de creación

POST localhost:37347/api/personafisica/api/personafisica/agregar

Header content-type application/json
Json Body

{

"Nombre":"Edgar",

"ApellidoPaterno":"Daniel",

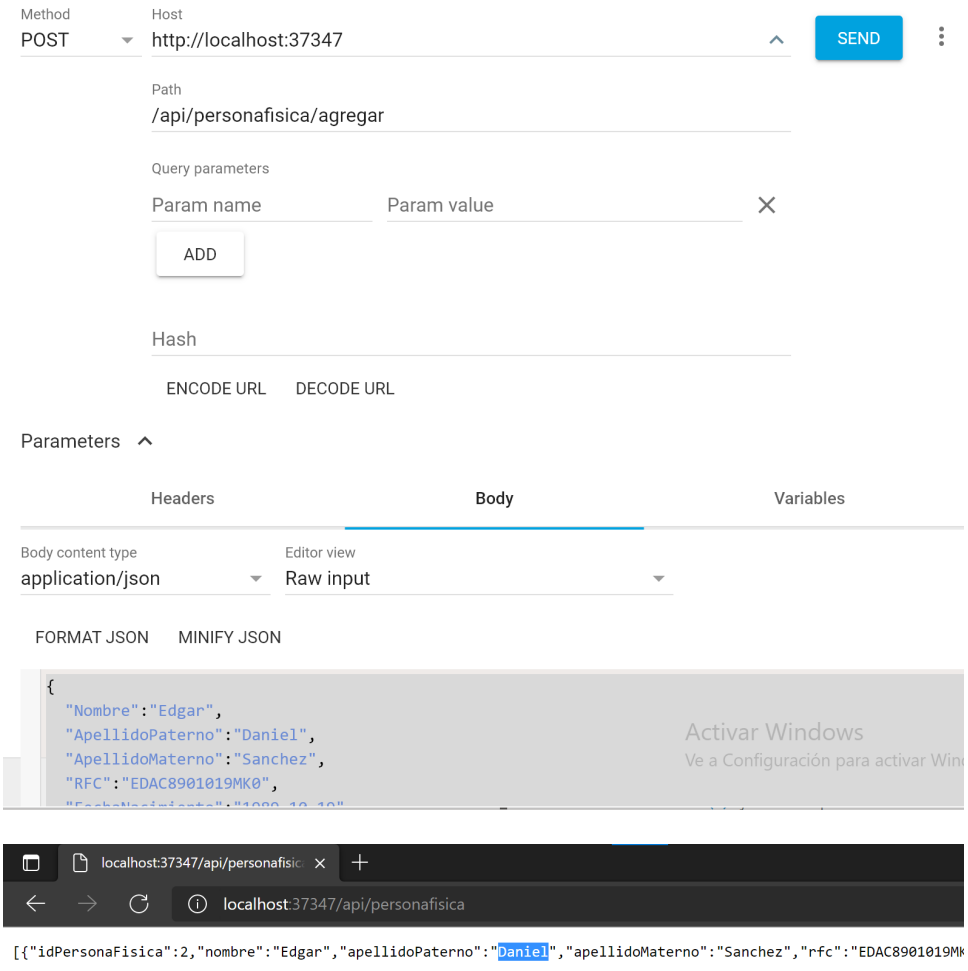
"ApellidoMaterno":"Sanchez",

"RFC":"EDAC8901019MK0",

"FechaNacimiento":"1989-10-19",

"UsuarioAgrega":1

}



Endpoint de actualización

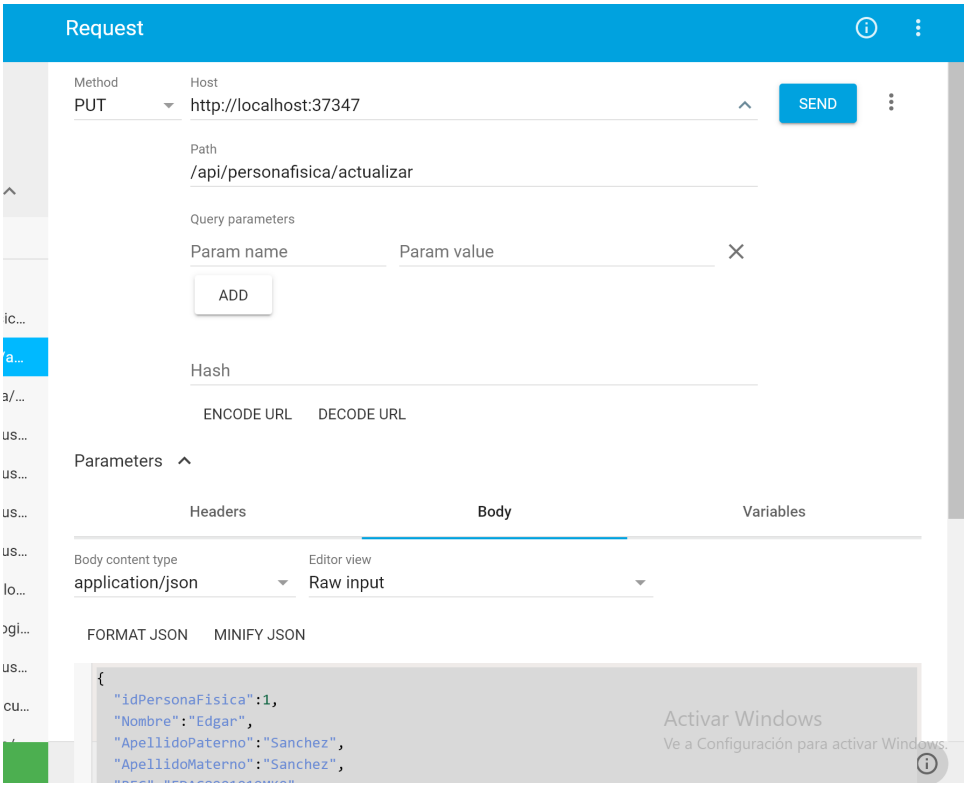
PUT http://localhost:37347/api/personafisica/actualizar

Header content-type application/json
Json Body

{
"idPersonaFisica":1,
"Nombre":"Edgar",
"ApellidoPaterno":"Sanchez",
"ApellidoMaterno":"Sanchez",
"RFC":"EDAC8901019MK0",
"FechaNacimiento":"1989-10-19",
"UsuarioAgrega":1
}

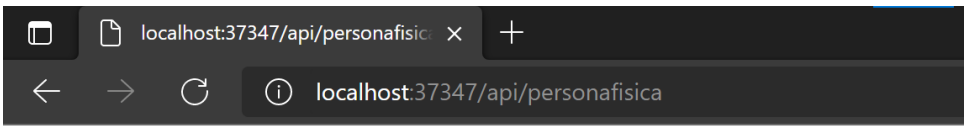
Datos antes de actualizar el apellido Paterno

Actualizando



Datos después de actualizar

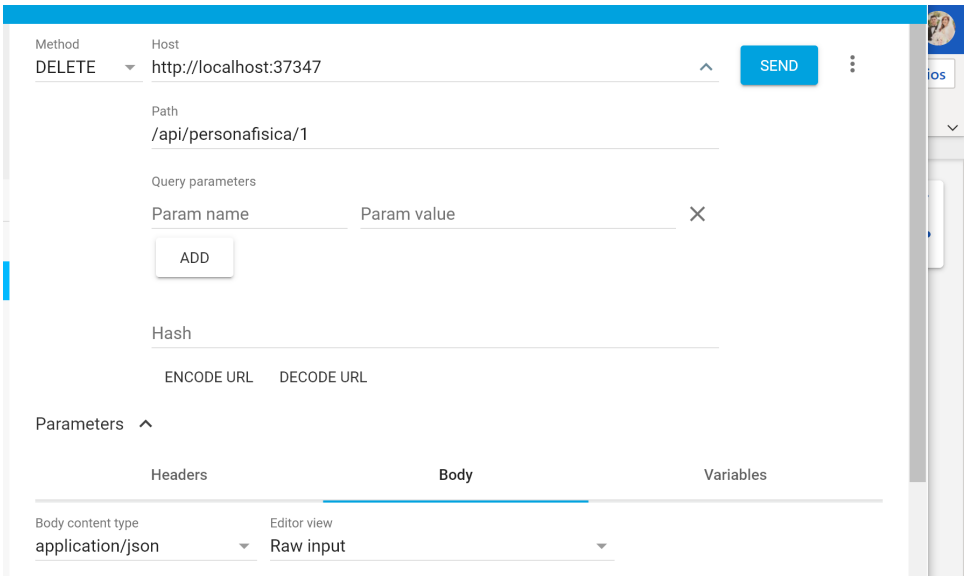
Edgar Daniel Sanchez Avila edgar.dan.s@hotmail.com



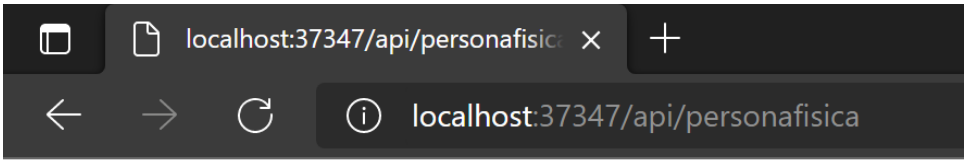
[{"idPersonaFisica":2,"nombre":"Edgar","apellidoPaterno":"Daniel","apellidoMate

Endpoint de borrado

DELETE http://localhost:37347/api/personafisica/2



Datos después de ejecutar el eliminar el id 2



[]

Sistema WEB Persona Fisica

Para el Sistema WEB persona Física se adjunta un video demo funcional en la siguiente liga:

https://youtu.be/31tzPIIQ7vs