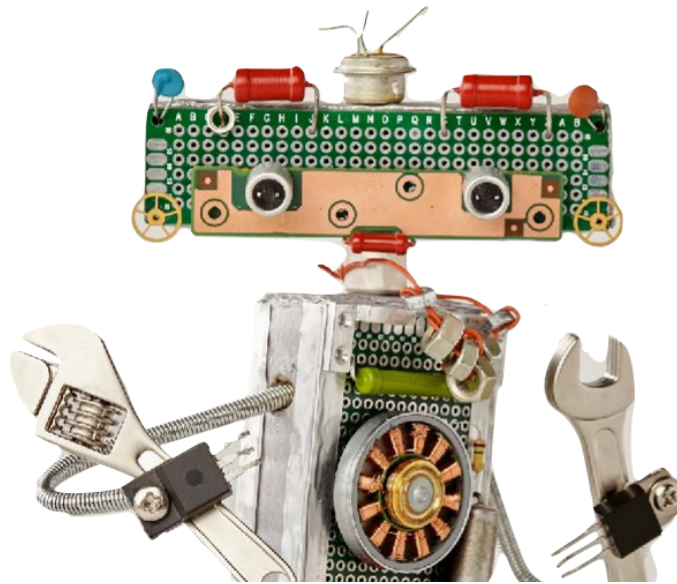


APIS FLUJOS INSTTANTT



Reto técnico

Misión para Backend Developer

Proceso RE013

TABLA DE CONTENIDO

Plataforma para probar las APIs : swagger	2
Características de la solución	9
Plataforma de desarrollo de la aplicación	9
Paquetes o extensiones instalados en la solución	9
Autorización de APIs	9
Arquitectura, organización y estructura de proyecto	9
Prácticas de desarrollo y Patrón de diseño	11
Diseño, modelado e implementación de DB	12
Modelado UML	12
Sistema de consultas de base de datos u ORM	12
Tablas implementadas en la base de datos	13
Notas sobre la implementación	14
URL Repositorio Git Privado	15

Plataforma para probar las APIs : swagger

A continuación se presenta la plataforma integrada swagger, para probar las APIs en .Net Core 6. Se puede utilizar otras plataformas como Postman.

Controlador ConstruccíonFlujo

Es quien construye la configuración de los flujo

Swagger
Supported by SMARTBEAR

Select a definition **EjecutorFlujosInstantt v1**

EjecutorFlujosInstantt 1.0 OAS3
<https://localhost:7124/swagger/v1/swagger.json>

ConstruccionFlujo

- GET /ConstruccionFlujo
- GET /ConstruccionFlujo/consultarflujo
- POST /ConstruccionFlujo/guardarconfigflujo
- POST /ConstruccionFlujo/guardarconfigpaso

ProcesamientoUsuario

- POST /ProcesamientoUsuario/guardarusuario
- POST /ProcesamientoUsuario/guardarflujousuario
- POST /ProcesamientoUsuario/guardarpasousuario
- POST /ProcesamientoUsuario/buscarpasousuario

Método Get	Respuesta / salida
<p>GET /ConstruccionFlujo</p> <p>Este método carga los campos y sus correspondientes validaciones, para que en el front se inicie la construcción de los pasos del flujo</p> <p>URL: https://localhost:7124/ConstruccionFlujo</p>	<p><u>ARCHIVO RESPUESTA</u></p> <pre> Response body { "camposDisponibles": [{ "codigoCampoDispo": "F-0001", "tituloCampo": "Primer nombre", "tipoCampo": "INPUT", "validacionesDisponibles": [{ "codigoValidacionDispo": "V-0001", "msgErrorValidacion": "No ha ingresado información al campo", "tipo": "Error" }, { "codigoValidacionDispo": "V-0002", "msgErrorValidacion": "El valor no debe contener números", "tipo": "Error" }, { "codigoValidacionDispo": "V-0003", "msgErrorValidacion": "El valor no puede contener caracteres especiales", "tipo": "Error" }, { "codigoValidacionDispo": "V-0005", "msgErrorValidacion": "Solo se permiten mayúsculas", "tipo": "Error" }] }] }</pre>

Método Get	Respuesta / salida
------------	--------------------

<div> <div>GET</div> <div>/ConstruccionFlujo/consultarflujo</div> </div> <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>CodigoFlujo</td><td>FLJ-0001</td></tr> <tr> <td>string</td><td></td></tr> <tr> <td>(query)</td><td></td></tr> </table> <p>Este método carga la configuración de un flujo seleccionado(CodigoFlujo) para poder ser diligenciado por el usuario.</p> <p>Ejemplo: FLJ-0001, FLJ-0002, FLJ-0003</p> <p>https://localhost:7124/ConstruccionFlujo/consultarflujo?CodigoFlujo=FLJ-0001</p>	Name	Description	CodigoFlujo	FLJ-0001	string		(query)		<h2>ARCHIVO RESPUESTA</h2> <p>Response body</p> <pre>{ "codigoFlujo": "FLJ-0001", "nombre": "Solicitud Financiera", "categoria": "Finanzas", "fechaCreacion": "2023-04-26T00:00:00", "creadorFlujo": "Wilmar Galvis", "pasosFlujo": [{ "codigoPaso": "STP-0001", "nombrePaso": "Paso1", "pasosPreviosRequeridos": "0", "codigoFlujo": "FLJ-0001", "camposxPasos": [{ "codigoCampoDispo": "F-0001", "codigoPaso": "STP-0001", "validacionesxCampo": [{ "codigoValidacionxCampo": "V-0001", "msgErrorValidacion": "No ha ingresado el nombre", "tipo": "Error", "codigoCampoDispo": "F-0001" }], "codigoValidacionxCampo": "V-0002", "msgErrorValidacion": "El valor no debe contener números", "tipo": "Error", "codigoCampoDispo": "F-0001" }] }] }</pre>
Name	Description								
CodigoFlujo	FLJ-0001								
string									
(query)									

Método Post	Respuesta / salida				
<div> <div>POST</div> <div>/ConstruccionFlujo/guardarconfigflujo</div> </div> <p>Este método guarda la configuración básica de un flujo que se está construyendo.</p> <p>Se piden parámetros descriptivos y de configuración general del flujo. El SP de BD genera el código del flujo automáticamente.</p> <p>Entrada</p> <pre>{</pre>	<p>Devuelve el código de flujo generado</p> <div> <div>Request URL</div> <div>https://localhost:7124/ConstruccionFlujo/guardarconfigflujo</div> </div> <div> <div>Server response</div> <table> <tr> <th>Code</th><th>Details</th></tr> <tr> <td>200</td><td> <div>Response body</div> <div>Código flujo: FLJ-0006</div> </td></tr> </table> </div>	Code	Details	200	<div>Response body</div> <div>Código flujo: FLJ-0006</div>
Code	Details				
200	<div>Response body</div> <div>Código flujo: FLJ-0006</div>				

<pre> "validacionesCreadas": [{ "codigoValidacionxCampo": "V-0002", "msgErrorValidacion": "El valor no debe contener números", "tipo": "Error", "codigoCampoDispo": "F-0002" }] }] } </pre> <p>https://localhost:7124/ConstruccionFlujo/guardarconfi gpaso</p>	
---	--

Controlador ProcesamientoUsuario.

Guarda el diligenciamiento del flujo por parte del usuario

Método Post	Respuesta / salida				
<div> <div>POST /ProcesamientoUsuario/guardarusuario</div> <p>Guarda los datos del usuario que va a diligenciar el flujo</p> <p>Entrada</p> <pre> { "codigoUsuario": "11483972", "nombreUsuario": "María Jimenez", "clave": "456" } </pre> <p>https://localhost:7124/ProcesamientoUsuario/guardar usuario</p> </div>	<div> <p>Responses</p> <pre> Curl curl -X 'POST' \ 'https://localhost:7124/ProcesamientoUsuario/guardarusuario' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "codigoUsuario": "11483972", "nombreUsuario": "María Jimenez", "clave": "456" }' </pre> <p>Request URL</p> <p>https://localhost:7124/ProcesamientoUsuario/guardarusuario</p> <p>Server response</p> <table> <thead> <tr> <th>Code</th><th>Details</th></tr> </thead> <tbody> <tr> <td>200</td><td> <p>Response body</p> <p>Código usuario guardado: 11483972</p> </td></tr> </tbody> </table> </div>	Code	Details	200	<p>Response body</p> <p>Código usuario guardado: 11483972</p>
Code	Details				
200	<p>Response body</p> <p>Código usuario guardado: 11483972</p>				

Método Post	Respuesta / salida				
<div>POST /ProcesamientoUsuario/guardarflujousuario</div> <p>Guarda los datos generales del flujo a completar, antes de diligenciar el 1er paso.</p> <p>El código del flujo debe existir, puesto que es un flujo que el front permite seleccionar crear, porque ya fue construido previamente.</p> <p>Entrada</p> <pre>{ "fechaVencimiento": "28/04/2023", "estadoFlujo": "Iniciado", "codigoUsuario": "11483972", "codigoFlujoUsuario": "FLJ-0003" }</pre> <p>https://localhost:7124/ProcesamientoUsuario/guardarflujousuario</p>	<div>Curl</div> <pre>curl -X 'POST' \ 'https://localhost:7124/ProcesamientoUsuario/guardarflujousuario' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "fechaVencimiento": "28/04/2023", "estadoFlujo": "Iniciado", "codigoUsuario": "11483972", "codigoFlujoUsuario": "FLJ-0003" }'</pre> <div>Request URL</div> <p>https://localhost:7124/ProcesamientoUsuario/guardarflujousuario</p> <div>Server response</div> <table> <thead> <tr> <th>Code</th><th>Details</th></tr> </thead> <tbody> <tr> <td>200</td><td> <div>Response body</div> <p>Código flujo guardado: FLJ-0003</p> </td></tr> </tbody> </table>	Code	Details	200	<div>Response body</div> <p>Código flujo guardado: FLJ-0003</p>
Code	Details				
200	<div>Response body</div> <p>Código flujo guardado: FLJ-0003</p>				

Método Post	Respuesta / salida				
<div>POST /ProcesamientoUsuario/guardarpasousuario</div> <p>Guarda cada paso del diligenciamiento del flujo del usuario, con los campos completados.</p> <p>Los datos de entrada, se dan porque previamente se buscó el flujo construido.</p> <p>Entrada</p> <pre>{ "codigoUsuario": "11483972", "codigoFlujoUsuario": "FLJ-0003", "codigoPasoUsuario": "STP-0001", "estado": "Progreso", "camposPasosUsuarios": [</pre>	<div>Curl</div> <pre>curl -X 'POST' \ 'https://localhost:7124/ProcesamientoUsuario/guardarpasousuario' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "codigoUsuario": "11483972", "codigoFlujoUsuario": "FLJ-0003", "codigoPasoUsuario": "STP-0001", "estado": "Progreso", "camposPasosUsuarios": [{ "tipoCampo": "INPUT", "dataCampo": "https://minube.com/imagen1" }] }'</pre> <div>Request URL</div> <p>https://localhost:7124/ProcesamientoUsuario/guardarpasousuario</p> <div>Server response</div> <table> <thead> <tr> <th>Code</th><th>Details</th></tr> </thead> <tbody> <tr> <td>200</td><td> <div>Response body</div> <p>Código paso guardado: STP-0001</p> </td></tr> </tbody> </table>	Code	Details	200	<div>Response body</div> <p>Código paso guardado: STP-0001</p>
Code	Details				
200	<div>Response body</div> <p>Código paso guardado: STP-0001</p>				

<pre>{ "tipoCampo": "INPUT", "dataCampo": "https://minube.com/imagen1" }]</pre> <p>https://localhost:7124/ProcesamientoUsuario/guardar pasousuario</p>	
---	--

Método Post	Respuesta / salida				
<p>POST /ProcesamientoUsuario/buscarpasousuario</p> <p>Busca la data del paso actual del flujo.</p> <p>Los datos de entrada, se dan porque previamente se buscó el flujo construido.</p> <p>Entrada</p> <pre>{ "codigoUsuario": "98712884", "codigoFlujoUsuario": "FLJ-0003", "codigoPasoUsuario": "STP-0003" }</pre> <p>https://localhost:7124/ProcesamientoUsuario/buscarpasousuario</p> <p>Notas:</p> <p>Por tiempo no fue posible terminar la lógica de este método.</p> <p>Por el momento solo confirma si el paso fue encontrado, pero no devuelve la data.</p> <p>Se debe buscar en la tbl camposxPasoxFlujoxUsuario la data del paso.</p>	<p>Responses</p> <p>Curl</p> <pre>curl -X 'POST' \ 'https://localhost:7124/ProcesamientoUsuario/buscarpasousuario' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "codigoUsuario": "98712884", "codigoFlujoUsuario": "FLJ-0003", "codigoPasoUsuario": "STP-0003" }'</pre> <p>Request URL</p> <p>https://localhost:7124/ProcesamientoUsuario/buscarpasousuario</p> <p>Server response</p> <table> <thead> <tr> <th>Code</th><th>Details</th></tr> </thead> <tbody> <tr> <td>200</td><td> <p>Response body</p> <p>Paso encontrado: STP-0003 por falta de tiempo no devolví los campos hallados</p> </td></tr> </tbody> </table>	Code	Details	200	<p>Response body</p> <p>Paso encontrado: STP-0003 por falta de tiempo no devolví los campos hallados</p>
Code	Details				
200	<p>Response body</p> <p>Paso encontrado: STP-0003 por falta de tiempo no devolví los campos hallados</p>				

para cargarlos a la vista. Al mismo tiempo se valida en la CamposPasosPreviosRequeridos si todos los pasos requeridos previamente ya estaban llenos, para poder continuar.	
--	--

Características de la solución

Plataforma de desarrollo de la aplicación

Las API Rest a mencionar están desarrolladas en .Net Core 6

Paquetes o extensiones instalados en la solución

No fue necesario agregar ningún paquete o librería externa.

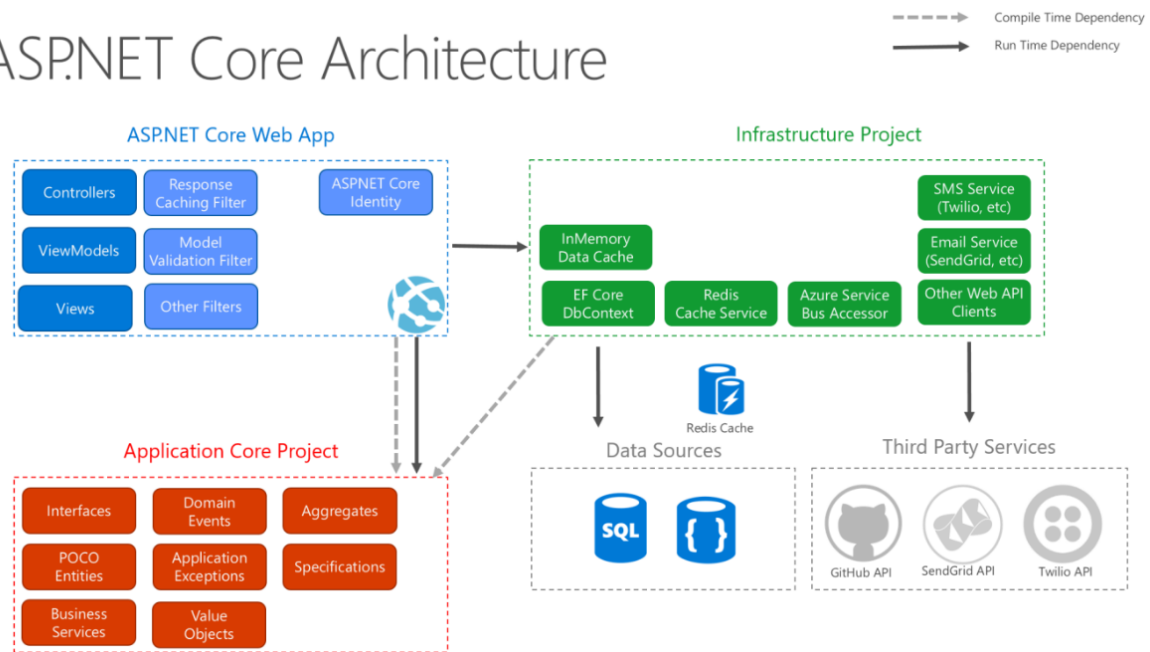
Autorización de APIs

De acuerdo a los requerimientos de la aplicación, no se aplicaron tokens de autorización.

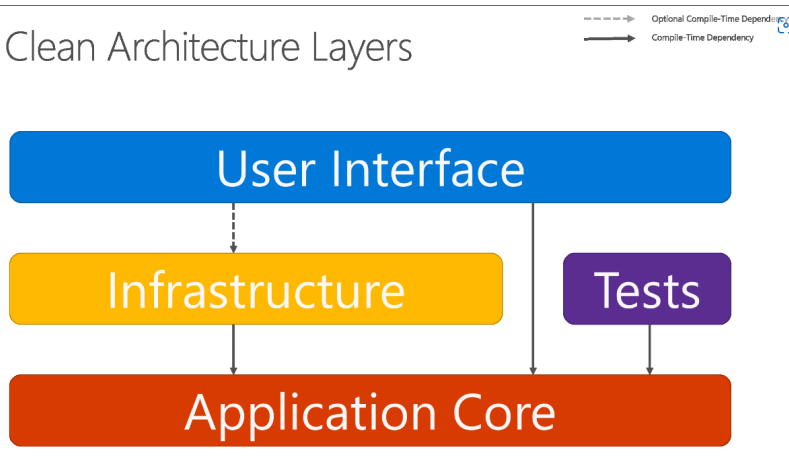
Arquitectura, organización y estructura de proyecto

Se aplicó la siguiente arquitectura recomendada por microsoft: Arquitectura limpia en Capas

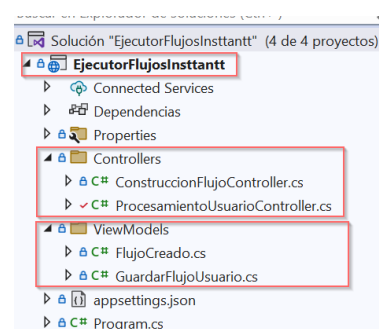
ASP.NET Core Architecture



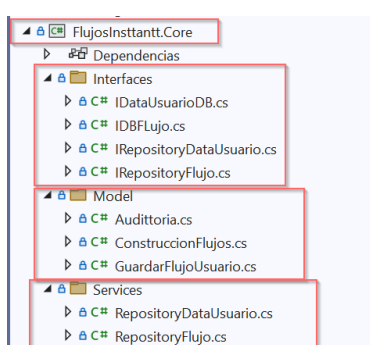
Clean Architecture Layers



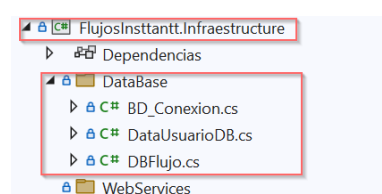
Capa web: controladores y view models necesarias de cara a las entidades del front.



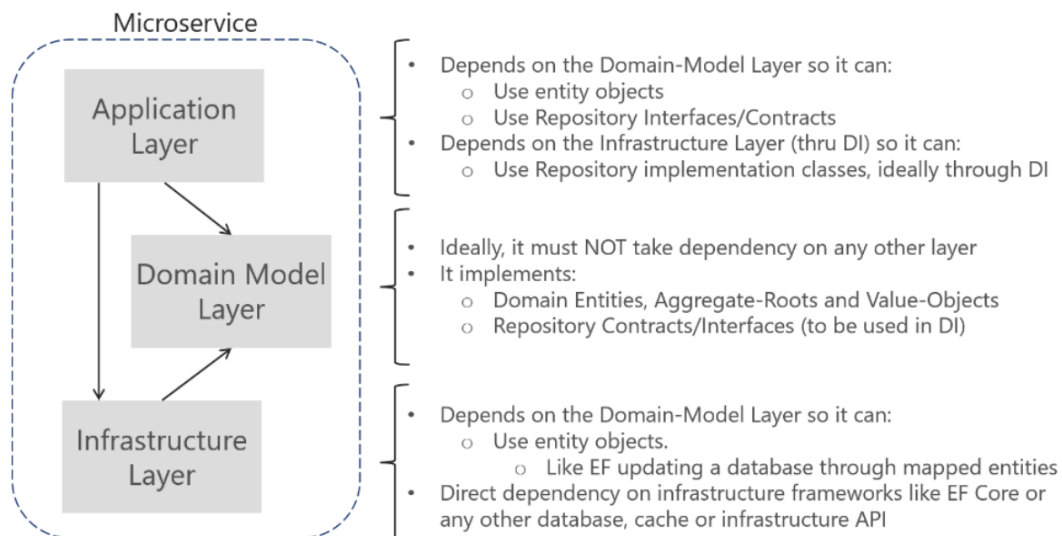
Capa Core: Es la capa de negocio. Capa de dominio si se compara con DDD



Capa Infraestructura: Capa de consulta y acceso a la data.



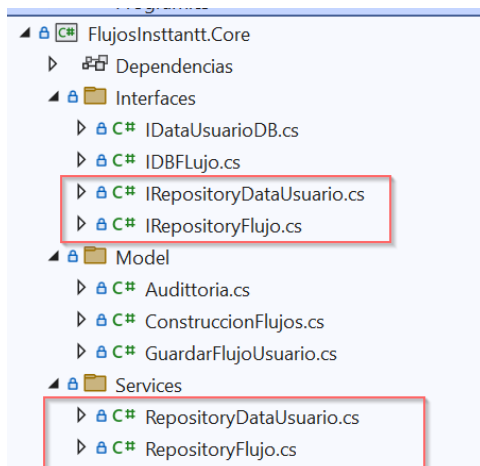
Dependencies between Layers in a Domain-Driven Design service



	<p>No se alcanzó a realizar pruebas unitarias, por falta de tiempo.</p> <p>Normalmente podría utilizar: XUNIT, NUnit</p>
--	--

Prácticas de desarrollo y Patrón de diseño

- Se utilizó Patrón Repositorio (Repository Pattern)



- Se aplicó Inyección de dependencias

```
builder.Services.AddTransient<IDBFlujo, DBFlujo>();
builder.Services.AddTransient<IDataUsuarioDB, DataUsuarioDB>();
```

- POO

Diseño, modelado e implementación de DB

Modelado UML

En el [siguiente enlace](#) está el modelo.

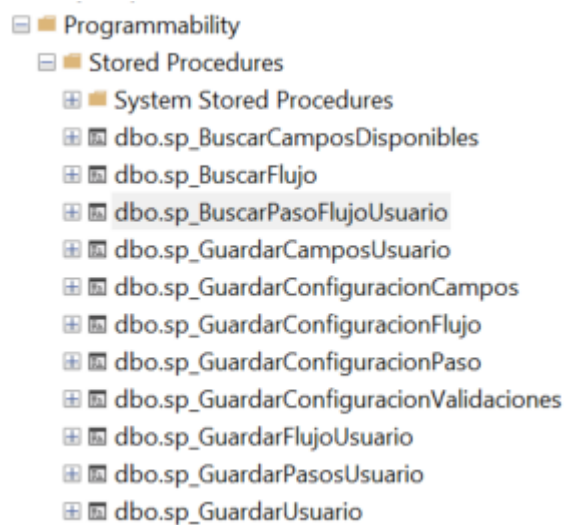
Otra opción de [visualización](#)

Se debe consultar las 2 tab o pestañas

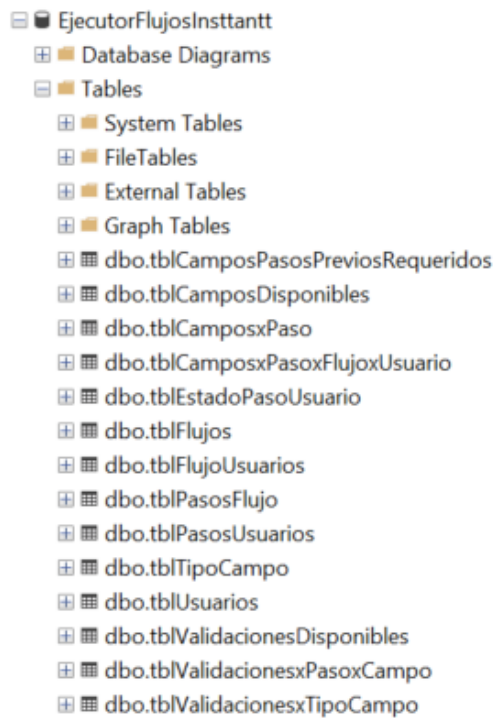


Sistema de consultas de base de datos u ORM

No se utilizó ORM, se crearon Stored procedures, bajo SQL Server



Tablas implementadas en la base de datos



Ejemplos:

sp_BuscarFlujo

```
SQLQuery135.sql - P...wilmar.galvis (53)) * X PMVM207\SQL2019.....tblPasosUsuarios
USE [EjecutorFlujosInstantt]
GO
/***** Object: StoredProcedure [dbo].[sp_BuscarFlujo] Script Date: 28/04/2023 11:17:27 p. m. *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_BuscarFlujo]
    @CodigoFlujo varchar(15)
AS
BEGIN
    SET NOCOUNT ON;

    select F.CodigoFlujo,F.Nombre, F.Categoria, F.FechaCreacion, F.CreadorFlujo, PF.CodigoPaso,
    PF.NombrePaso, PF.PasosPreviosRequeridos,CP.CodigoCampoDispo,VC.CodigoValidacionxCampo,
    VC.MsgErrorValidacion, VC.Tipo
    from tblFlujos AS F INNER JOIN tblPasosFlujo AS PF ON
    PF.CodigoFlujo = F.CodigoFlujo INNER JOIN tblCamposxPaso AS CP ON
    CP.CodigoPaso = PF.CodigoPaso INNER JOIN tblValidacionesxPasoxCampo AS VC ON
    VC.CodigoCampoDispo = CP.CodigoCampoDispo AND VC.CodigoPaso =CP.CodigoPaso
    where F.CodigoFlujo = @CodigoFlujo ORDER BY PF.CodigoPaso ASC
END
```

sp_GuardarConfiguracionPaso

```

DECLARE @CodPaso varchar(15) = '';

--Se consulta el último código de los pasos
DECLARE @query varchar(max);
set @query = (select top 1 CodigoPaso from tblPasosFlujo where CodigoFlujo = @CodigoFlujo order by CodigoPaso desc);

set @CodPaso = @query;
print @CodPaso;

IF (@CodPaso != '')
BEGIN

    --Se extrae el número después del 4to caracter, del código del paso, y se agrega un dígito más
    DECLARE @NumPaso int;
    SET @NumPaso = SUBSTRING(@CodPaso, 5, LEN(@CodPaso));
    set @NumPaso = @NumPaso + 1;
    print @NumPaso;

    --Se agrega al código del flujo los ceros a la izquierda faltantes
    DECLARE @NumPasoCompleto VARCHAR(8);
    SET @NumPasoCompleto = REPLICATE('0', 4 - LEN(RIGHT(@NumPaso, 4))) + CAST(@NumPaso AS VARCHAR(4));

    --Se concatena la sila del flujo con el número correspondiente
    set @CodPaso = 'STP-' + @NumPasoCompleto;
    print @CodPaso;

END
ELSE
BEGIN

    set @CodPaso = 'STP-0001'

END

insert into tblPasosFlujo(CodigoPaso,NombrePaso,PasosPreviosRequeridos,CodigoFlujo)
values(@CodPaso,@NombrePaso,@PasosPreviosRequeridos,@CodigoFlujo)

select @CodPaso
END

```

Notas sobre la implementación

No tuve el tiempo, pero normalmente configuro de manera básica Amazon para subir la aplicación a la nube.

Grupo de usuarios creado: developers

Usuario creado: vs_deploy_agent

Permisos otorgados al usuario:

- AmazonEC2FullAccess
- AdministratorAccess-AWSElasticBeanstalk

Users > **vs_deploy_agent**

Summary

Delete user ⓘ

User ARN arn:aws:iam::517588588598:user/vs_deploy_agent ⓘ

Path /

Creation time 2022-11-27 13:04 EST

Permissions **Groups (1)** **Tags (1)** **Security credentials** **Access Advisor**

Add user to groups

Group name	Attached permissions
developers	AmazonEC2FullAccess and AdministratorAccess-AWSElasticBeanstalk ✕...

URL Repositorio Git

<https://github.com/wilmargalvis/FlujosInstantt>

<> Code Issues Pull requests Actions Projects Security Insights Settings

master **ComunicacionesBaseEspacial** / **ComunicacionesBaseEspacial** /

Wilmar Galvis Implementación inyección de dependencias

..

.config	cambios para publicación y funcionamiento de Swagger
Controllers	Implementación inyección de dependencias
Interfaces	Implementación inyección de dependencias
Models	sincronización inicial solución
Properties	cambios para publicación y funcionamiento de Swagger
Services	Implementación inyección de dependencias
bin	cambios para publicación y funcionamiento de Swagger
obj	Implementación inyección de dependencias
ComunicacionesBaseEspacial.csproj	sincronización inicial solución
Program.cs	Implementación inyección de dependencias