Manual Técnico del software, Prueba técnica.

Diego Puentes Satrack

Contenido

Manual Técnico del software, Prueba técnica	. 1
Creación de base de datos y tablas.	. 1
Desarrollo de BACKEND	. 2
FROND END	. 9
Interfaz aráfica	10

Creación de base de datos y tablas.

Se creo la siguiente estructura de base datos.



```
--CREATE DATABASE SATRACK;
USE SATRACK

CREATE TABLE ESTADO (
ID_Estado int IDENTITY(1,1) PRIMARY KEY,
Nombre_Estado NVARCHAR(50)
);

CREATE TABLE CATEGORIA (
ID_Categoria INT IDENTITY(1,1) PRIMARY KEY,
```

```
Nombre_Categoria NVARCHAR(50)
);

CREATE TABLE TAREA (
ID_Tarea INT IDENTITY(1,1) PRIMARY KEY,
Titulo_Tarea NVARCHAR(100) not null,
Descripcion_Tarea text,
Fecha_Creacion DATE,
Fecha_Finalizacion DATE,
Fecha_Modificacion DATE,
Categoria_Tarea INT FOREIGN KEY REFERENCES CATEGORIA(ID_Categoria),
Estado_Tarea INT FOREIGN KEY REFERENCES ESTADO(ID_Estado)
);
```

Inserts de prueba tablas ESTADO y CATEGORIA

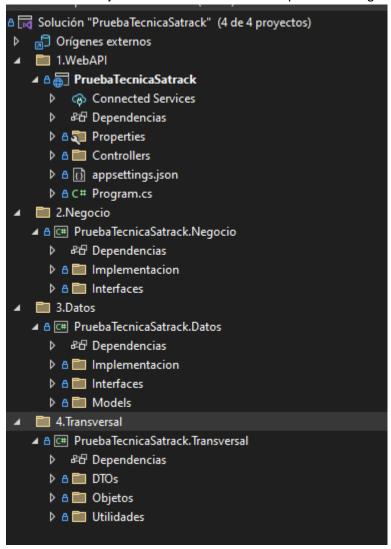
```
--Insersión de datos en la tabla ESTADO
INSERT INTO ESTADO VALUES ('Nueva');
INSERT INTO ESTADO VALUES ('Activa');
INSERT INTO ESTADO VALUES ('Bloqueada');
INSERT INTO ESTADO VALUES ('Cerrada');

--Insersión de datos en la tabla CATEGORIA
INSERT INTO CATEGORIA VALUES ('INCIDENTE');
INSERT INTO CATEGORIA VALUES ('NUEVA FUNCIONALIDAD');
```

Desarrollo de BACKEND

Creación de API REST.

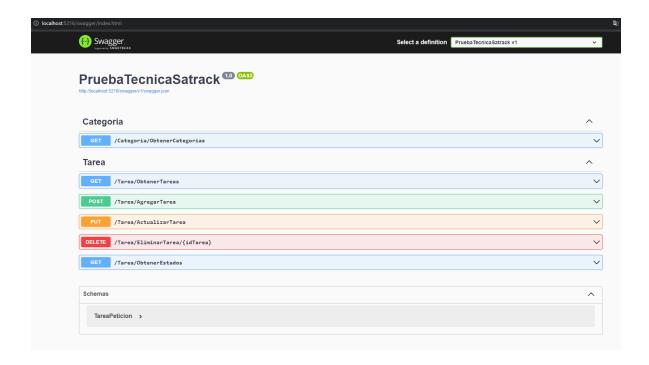
Se desarrollo bajo el framework .NET 6 se implmento la siguiente arquitectura.



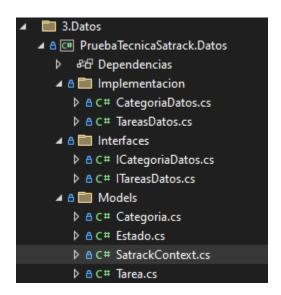
Capa de exposición de WebServices, la cual quedo documentada con Swagger de la siguiente forma.

URL Dominio/swagger/index.html Ejemplo : http://localhost:5216/swagger/index.html

Desarrollo de diferentes métodos HTTP GET POST PUT DELETE



Se implenteto API REST - Database First. Para el acceso a datos con su DBCONTEXT y objetos de entidades de SQL server, se genero el acceso a la implementación de base de datos a través Interfaces.



API REST - Configuración de AutoMapper.

Se implemento esta librería para el mapeo de objetos de negocio DTOs, con el fin de no exponer lo objetos que vienen directamente de base de datos.

```
public class AutoMapperProfile : Profile
    public AutoMapperProfile()
        #region Tarea
        CreateMap<Tarea, TareaDTO>()
             .ForMember(destino => destino.CategoriaTarea, opt => opt.MapFrom(origen => origen.CategoriaTareaNavigation))
             .ForMember(destino => destino.EstadoTarea, opt => opt.MapFrom(origen => origen.EstadoTareaNavigation))
.ForMember(destino => destino.IdCategoria, opt => opt.MapFrom(origen => origen.CategoriaTarea))
              .ForMember(destino => destino.IdEstado, opt => opt.MapFrom(origen => origen.EstadoTarea))
        CreateMap<TareaPeticion, Tarea>()
.ForMember(destino => destino.TituloTarea, opt => opt.MapFrom(origen => origen.TituloTarea))
             .ForMember(destino => destino.EstadoTarea, opt => opt.MapFrom(origen => origen.idEstado))
             .ForMember(destino => destino.CategoriaTarea, opt => opt.MapFrom(origen => origen.idCategoria))
             .ForMember(destino => destino.DescripcionTarea, opt => opt.MapFrom(origen => origen.descripcionTarea))
        #region Categoria
        CreateMap<Categoria, CategoriaDTO>()
             .ForMember(destino => destino.Id, opt => opt.MapFrom(origen => origen.IdCategoria))
             .ForMember(destino => destino.Nombre, opt => opt.MapFrom(origen => origen.NombreCategoria))
        CreateMap<CategoriaDTO, Categoria>()
             .ForMember(destino => destino.IdCategoria, opt => opt.MapFrom(origen => origen.Id))
             .ForMember(destino => destino.NombreCategoria, opt => opt.MapFrom(origen => origen.Nombre))
        #endregion
        #region Estado
        CreateMap<Estado, EstadoDTO>()
.ForMember(destino => destino.Id, opt => opt.MapFrom(origen => origen.IdEstado))
             .ForMember(destino => destino.Nombre, opt => opt.MapFrom(origen => origen.NombreEstado))
        #endregion
```

API REST - Configuración de servicios.

Se realizo la configuración de servicios para uso de inyección de dependencias en el archivo startup.

```
using PruebaTecnicaSatrack.Negocio.Interfaces;
    using PruebaTecnicaSatrack.Transversal.Utilidades;
   using System.Text.Json.Serialization;
    var builder = WebApplication.CreateBuilder(args);
    // Add services to the container.
    builder.Services.AddControllers();
  builder.Services.AddDbContext<SatrackContext>(options =>
        options.UseSqlServer(builder.Configuration.GetConnectionString("ConnectionStringSatrack"));
   3);
    builder.Services.AddScoped<ITareasDatos, TareasDatos>();
    builder.Services.AddScoped<ITareasNegocio, TareasNegocio>();
    builder.Services.AddScoped<ICategoriaNegocio, CategoriaNegocio>();
    builder.Services.AddScoped<ICategoriaDatos, CategoriaDatos>();
    builder.Services.AddAutoMapper(typeof(AutoMapperProfile));
    builder.Services.AddSwaggerGen();
    builder.Services.AddCors(
        options=>options.AddPolicy("NuevaPolitica",app =>
            app.AllowAnyOrigin()
            .AllowAnyHeader()
            .AllowAnyMethod()
        );
    builder.Services.AddControllers().AddJsonOptions(x =>
                    x.JsonSerializerOptions.ReferenceHandler = ReferenceHandler.IgnoreCycles);
    var app = builder.Build();

√ □if (app.Environment.IsDevelopment())
        app.UseSwagger();
        app.UseSwaggerUI();
    app.UseCors("NuevaPolitica");
    app.UseAuthorization();
    app.MapControllers();
    app.Run();
```

API REST - Configuración solicitudes API.

Se implementaron controladores en la capa de servicios (Presentación)

Para la exposición de los métodos, se utilizó programación asíncrona, validaciones de request, control de errores, logs, respuestas http, inyección de dependencias y buenas prácticas de clean code, para que sea un código legible, organizado y funcional.

```
TareaController.cs → X TareaPeticion.cs
                                                               SatrackContext.cs
              🗝 😽 Prueba Tecnica Satrack. Controllers. Tarea Contro 💌 😭 Agregar Tarea (Tarea Peticion tarea)
                                                                                                     ·÷ 🚜 ७·≒ 🗇 🗗 😘 · 🌶 🛎
private readonly ILogger<TareaController> _logger;
private readonly ITareasNegocio _tareasNegocio;
                                                                                                          Buscar en Explorador de soluciones (Ctrl+)
                                                                                                          △ 🖂 Solución "PruebaTecnicaSatrack" (4 de 4 proyectos)
                                                                                                            ▲ 🛅 1.WebAPI
public TareaController(ILogger<TareaController> logger, ITareasNegocio tareasNegocio)
                                                                                                              ▲ △ ➡ PruebaTecnicaSatrack
                                                                                                                 Connected Services
     _tareasNegocio = tareasNegocio;
                                                                                                                 ▶ ₽☐ Dependencias
    _logger = logger;
                                                                                                                ▶ a 🔊 Properties
                                                                                                                 ▲ A Controllers
[HttpGet("ObtenerTareas")]
                                                                                                                  ▶ & C# CategoriaController.cs
public async Task<IActionResult> ObtenerTareas()
                                                                                                                   ▶ 🐧 🔯 appsettings.json

■ 2.Negocio

         var resultado = await _tareasNegocio.ObtenerTodasLasTareas();
if (!resultado.Any())
                                                                                                              ▲ 🐧 💷 PruebaTecnicaSatrack.Negocio
                                                                                                                 ▶ #☐ Dependencias
                                                                                                                 ▶ 🗈 🛅 Implementacion
             return NotFound("No se encontraron tareas.");
                                                                                                                 ▶ A 🛅 Interfaces

■ 3.Datos

         return Ok(resultado);
                                                                                                              ▲ 🗈 💷 PruebaTecnicaSatrack.Datos
                                                                                                                 ▶ ♣☐ Dependencias
    catch (Exception ex)
                                                                                                                 ▲ 🖺 Implementacion
         _logger.LogError(ex.ToString());
return Problem(
                                                                                                                   ▶ & C# CategoriaDatos.cs
                                                                                                                   D ≜ C# TareasDatos.cs
               detail: ex.Message
                                                                                                                 ▲ 🗈 🛅 Interfaces
                                                                                                                   ▶ & C# ICategoriaDatos.cs
                                                                                                                   ▶ A C# |TareasDatos.cs

▲ A I Models

[HttpPost("AgregarTarea")]
                                                                                                                   ▶ & C# Categoria.cs
public async Task<IActionResult> AgregarTarea(TareaPeticion tarea)
                                                                                                                   D A C# Estado.cs
                                                                                                                   ▶ A C# SatrackContext.cs
                                                                                                                   D A C# Tarea.cs
         string validacionObjetoPeticionTarea = _tareasNegocio.ValidarObjeto(tarea);
if (string.IsNullOrEmpty(validacionObjetoPeticionTarea))
                                                                                                           ▲ 4.Transversal
                                                                                                              ▲ A C# PruebaTecnicaSatrack,Transversal
                                                                                                                 ▶ & Dependencias
             await _tareasNegocio.AgregarTarea(tarea);
return Ok();
                                                                                                                ▶ a 🛅 DTOs
                                                                                                                 ▲ 🖺 Utilidades
         else
                                                                                                                    ▶ & C# AutoMapperProfile.cs
              return BadRequest(validacionObjetoPeticionTarea);
```

Se implemento la capa de negocio para construir toda la lógica que requería el servicio.

```
TareasNegocio.cs → X | TareasNegocio.cs

        In Prueba Tecnica Satrack. Negocio. Implementaci
        →
        ✓a_datos

                                                                                                                                      · ÷ 🗿 🖔 · 与 🗇 🖫
              □namespace PruebaTecnicaSatrack.Negocio.Implementacion
                                                                                                                                            ↑ Solución "PruebaTecnicaSatrack" (4 de 4 pr
                     2 referencias public class TareasNegocio : ITareasNegocio
 ĦŤ

■ 1.WebAPI

                          private readonly ITareasDatos _datos;
private readonly IMapper _mapper;
Oreferencias
public TareasNegocio(ITareasDatos datos, IMapper mapper)
{

▲ △ 

■ PruebaTecnicaSatrack

                                                                                                                                                  Connected Services
                                                                                                                                                   ▶ & Dependencias
                                                                                                                                                   ▶ 🐧 🐙 Properties
                                                                                                                                                   ▲ A Controllers
                                _datos = datos;
                                                                                                                                                     ▶ & C# CategoriaController.cs
                                _mapper = mapper;
                                                                                                                                                      ▶ 🐧 🕜 appsettings.json
                                                                                                                                                   D △ C# Program.cs
                          2 referencias
public string ValidarObjeto(TareaPeticion tarea) {
                                                                                                                                             2.Negocio
 ΠŤ
                               string ErrorPeticion = string.Empty;

▲ △ □ PruebaTecnicaSatrack.Negocio

                                                                                                                                                  ▶ ♣© Dependencias
                               if (string.IsNullOrEmpty(tarea.TituloTarea))
    ErrorPeticion = string.Format("{0}-{1}", ErrorPeticion, "Titulo es requerido."
if (tarea.idCategoria==0)
                                                                                                                                                   ▲ 🏝 Implementacion
                                                                                                                                                    ▶ & C# CategoriaNegocio.cs
                               ErrorPeticion = string.Format("{0}-{1}", ErrorPeticion, "Categoria es requerida if (tarea.idEstado==0)

■ 3.Datos
                                    ErrorPeticion = string.Format("{0}-{1}", ErrorPeticion, "Estado es requerido."

▲ A C# PruebaTecnicaSatrack.Datos

                               return ErrorPeticion;
                                                                                                                                                  ▶ ♣© Dependencias
                                                                                                                                                   ▲ â 🛅 Implementacion
                          2 referencias
public async Task<bool> AgregarTarea(TareaPeticion tarea)
                                                                                                                                                    ▶ 🐧 C# CategoriaDatos.cs
 ПΤ
                                                                                                                                                     ▷ 🐧 C# TareasDatos.cs
                                                                                                                                                   ▲ â 🛅 Interfaces
                               var tareaDatos = _mapper.Map<Tarea>(tarea);
tareaDatos.FechaCreacion = DateTime.Now;
                                                                                                                                                     ▶ & C# ICategoriaDatos.cs
                               tareaDatos.EstadoTarea = 1;
return await _datos.AgregarTarea(tareaDatos);
                                                                                                                                                     ▲ 🗂 Models
                                                                                                                                                     ▶ & C# Categoria.cs
                                                                                                                                                      ▶ & C# Estado.cs
                          2 referencias
public async Task<bool> ActualizarTarea(TareaPeticion tarea)
                                                                                                                                                      ▶ A C# SatrackContext.cs
 Πî
                                                                                                                                                     D A C# Tarea.cs
                                                                                                                                             ▲ ■ 4.Transversal
                                var existeTarea = _datos.ObtenerTareaPorId(tarea.IdTarea);
                                                                                                                                                ▲ △ © PruebaTecnicaSatrack.Transversal
                                                                                                                                                  ▶ &⊞ Dependencias
                                                                                                                                                   Þ å 🛅 DTOs
                                                                                                                                                   Þ â 🛅 Objetos

▲ A I Utilidades

                                var tareaDatos = _mapper.Map<Tarea>(tarea);
                                                                                                                                                     ▶ A C# AutoMapperProfile.cs
                               tareaDatos.FechaModificacion = DateTime.Now;
return await _datos.ActualizarTarea(tareaDatos);
                                                                                                  Línea: 1 Carácter: 1 SPC CRLF Explorador de soluciones Cambios de GIT: satr
```

Se implemento una capa transversal para DTOs, Objetos de respuesta y petición y utilidades en paralelo.

```
4.Transversal

4.Emal Prueba Tecnica Satrack. Transversal

Dependencias

A DTOs

A C# Categoria DTO.cs

A C# Estado DTO.cs

A C# Tarea DTO.cs

A Dipletos

A Dipletos
```

FROND END

Implemente **ANGULAR** para el desarrollo del frontend, es un framework muy fuerte para este tipo de aplicaciones, usando su **patron de diseño MVVM modelo vista vista modelo, donde** se puede desacoplar el código y pueda ser mantenible de una forma ordenada.

Se usaron patrones como el observador en su implementación,

Librerias como material las cueales ayudan mucho al desarrollo.

Formularios Reactivos.

```
> .angular
                                                                                                                                                 import { Injectable } from '@angular/core';
import {HttpClient} from '@angular/common/http';
import { environment } from 'src/environments/environment';
import {Observable} from 'rxjs';
import { Tarea } from '../Interfaces/tarea';
 {} tasks.ison
> node modules
                                                                                                                                                  @Injectable({
   providedIn: 'root'

√ app

∨ Dialogs \ dialog-agregar-tarea

    # dialog-agregar-tarea.component.css

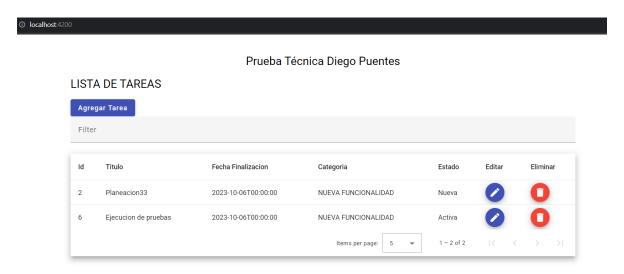
    dialog-agregar-tarea.component.html

                                                                                                                                                    private endpoint:string = environment.endPoint;
private apiUrl:string = this.endpoint + 'Tarea/';
constructor(private http:HttpClient) { }
    TS dialog-agregar-tarea.component.ts
    TS estado.ts
                                                                                                                                                    getTareas():0bservable<Tarea[]>{
    return this.http.get<Tarea[]>(`${this.apiUrl}0btenerTareas`);
    TS tarea.ts
    Services
    TS categoria.service.ts
                                                                                                                                                     addTarea(modelo:Tarea):Observable<Tarea>{
   return this.http.post<Tarea>(`${this.apiUrl}AgregarTarea`,modelo);
    TS app-routing.module.ts
                                                                                                                                                     updateTarea(modelo:Tarea):Observable<Tarea>{
   # app.component.css
                                                                                                                                                       console.log(modelo);
   o app.component.html
                                                                                                                                                     return this.http.put<Tarea>(`${this.apiUrl}ActualizarTarea`,modelo)
   TS app.component.ts
                                                                                                                                                     deleteTarea(idTarea:number):Observable<void>{
   return this.http.delete<void>(`${this.apiUrl}EliminarTarea/${idTarea
  > environments
  * favicon.ico
   index.html
.editorconfig
 gitignore
{} angular.json
{} package-lock.json
{} tsconfig.app.json
🖪 tsconfig.json
```

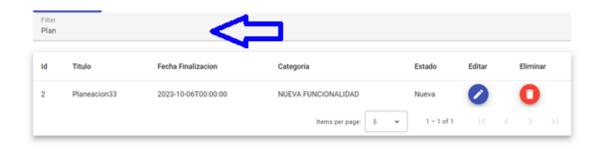
Se implementaron buenas practica como el uso de SOLID en diferentes partes del proyecto tanto en backend como en frond end

Interfaz gráfica.

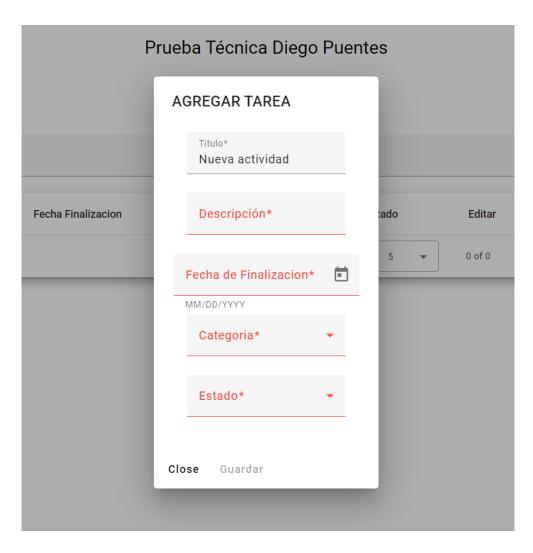
Lista de Tareas



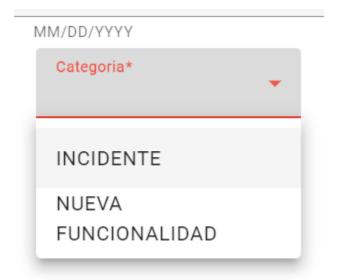
Se realizo un **filtro** el cual puede consultar por cualquier dato incluyendo **categoría** el cual estaba especificado.

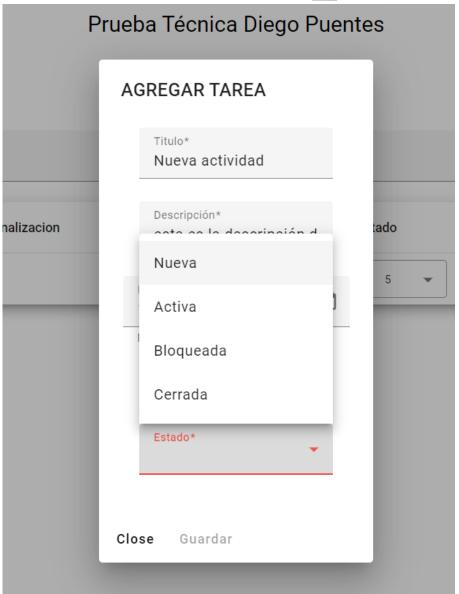


Se agrego un modulo para agregar una nueva tarea el cual es un formulario reactivo y esta validado para que exiga los parámetros requeridos



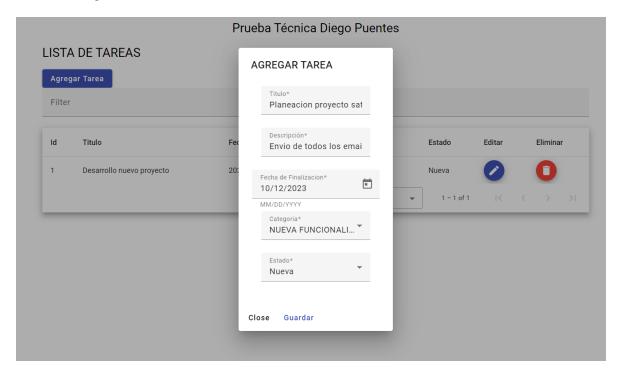
Las categorías tanto como los estados los trae de base de datos y los pinta para seleccionar



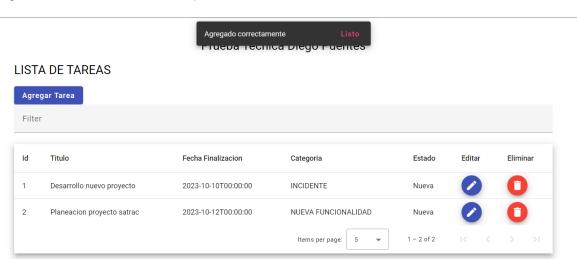


El usuario puede cambiar el estado de la tarea si así lo desea.

Pureba de guardar información



Se guarda satisfactoriamente y se refresca automaticamente



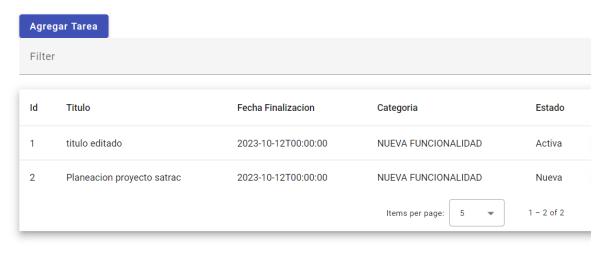
Prueba de edición



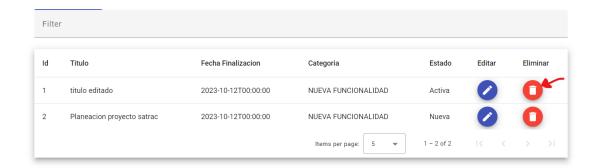
Actualización correcta



LISTA DE TAREAS



<u>Eliminación</u>



Se elimino correctamente



LISTA DE TAREAS

