

WorkFlowX

Enlace Github = https://github.com/DTomass/Proyecto_DAM

Resumen:

WorkFlowX es una solución diseñada para facilitar la gestión eficiente de proyectos y tareas dentro de una organización. Con esta aplicación, los usuarios pueden crear proyectos, asignar tareas a miembros del equipo, establecer roles y permisos, y mantener un seguimiento detallado del progreso.

La creación de proyectos es el primer paso para estructurar el trabajo. Los usuarios pueden definir los detalles del proyecto, establecer plazos y asignar recursos necesarios. Además, la aplicación permite asignar tareas específicas a los miembros del equipo, junto con fechas límite y prioridades.

Una característica clave de la aplicación es la capacidad de crear roles y asignar permisos. Los usuarios pueden definir diferentes roles, como administrador, gerente de proyecto o miembro del equipo... Esto garantiza que cada usuario tenga acceso solo a las funcionalidades y datos relevantes para su rol, lo que mejora la seguridad y la eficiencia en la colaboración.

La aplicación también ofrece un panel de control intuitivo para visualizar el progreso del proyecto y las tareas asignadas.

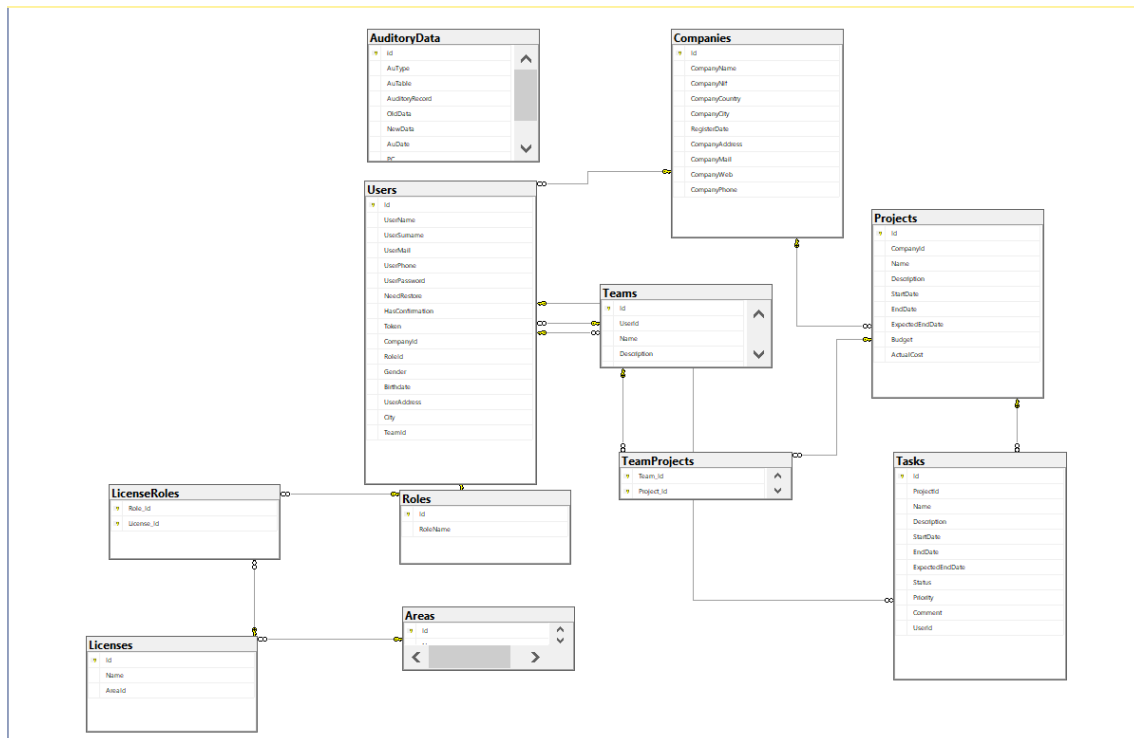
1. Objetivos del Proyecto

El proyecto de WorkFlowX tiene como objetivo principal proporcionar una solución práctica y eficiente para gestionar proyectos y tareas. Aquí te presento los objetivos específicos que se buscan alcanzar con esta aplicación:

1. Optimizar la gestión de proyectos
2. Facilitar la asignación y seguimiento de tareas
3. Establecer roles y permisos personalizados
4. Proporcionar un seguimiento y control efectivo del progreso del proyecto
5. Aumentar la eficiencia y productividad

3. Arquitectura del Proyecto

Diagrama de la base de datos



Estructura del proyecto

El proyecto se construye como una aplicación ASP.NET Que usa Net Framework 4.7.2, el código se estructura de la siguiente manera:

1. **Controllors**: se guarda la lógica de la aplicación, el redireccionamiento de paginas y las llamadas a base de datos desde una clase DbContext.
2. **Models**: guardan los modelos, dtos y enumerables de la aplicación.
3. **Views**: guarda la interfaz con la que interactúa el usuario.
4. **Services**: guarda la lógica de encriptar la contraseña y generar el toquen así como el envío de correos.

```

public static bool SendMail(DtoMail mail)
{
    try
    {
        var message = new MimeMessage();
        message.From.Add(new MailboxAddress(_username, _usermail));
        message.To.Add(MailboxAddress.Parse(mail.MailTo));
        message.Subject = mail.MailSubject;
        message.Body = new TextPart(TextFormat.Html)
        {
            Text = mail.MailBody
        };
        var smtp = new SmtpClient();
        smtp.CheckCertificateRevocation = false;
        smtp.Connect(_host, _port, SecureSocketOptions.Auto);

        smtp.Authenticate(_usermail, _password);
        smtp.Send(message);
        smtp.Disconnect(true);
        return true;
    }
    catch (Exception ex)
    {
        return false;
    }
}

```

3 referencias

```

public static string ConvertToSHA256(string texto)
{
    string hash = string.Empty;

    using(SHA256 sha256 = SHA256.Create())
    {
        byte[] hasValue = sha256.ComputeHash(Encoding.UTF8.GetBytes(texto));

        foreach(byte b in hasValue)
        {
            hash += $"{b:X2}";
        }
        return hash;
    }
}

```

1 referencia

```

public static string GenerateToken()
{
    string token = Guid.NewGuid().ToString("N");
    return token;
}

```

5. Filters: guardan la lógica detrás de la verificación de permisos y de la cuenta con la que te logueas.

```

0 referencias
public override void OnAuthorization(AuthorizationContext filterContext)
{
    var neededLicense = db.Licenses.Include(l => l.Area).FirstOrDefault(l => l.Id == li
    try
    {
        user = (User)HttpContext.Current.Session["User"];
        user = db.Users.Include(u => u.Role.Licenses).FirstOrDefault(u => u.Id == user.

        if (!user.Role.Licenses.Contains(neededLicense))
        {
            filterContext.Result = new RedirectResult("~/Home/NoAuthorized?licenseName=
        }
    } catch (Exception ex)
    {
        filterContext.Result = new RedirectResult("~/Home/NoAuthorized?licenseName=" +
    }
}

```

```

0 referencias
public override void OnActionExecuting(ActionExecutingContext filterContext)
{
    try
    {
        base.OnActionExecuting(filterContext);

        user = (User)HttpContext.Current.Session["User"];
        if (user == null)
        {
            if(filterContext.Controller is StartController == false)
            {
                filterContext.HttpContext.Response.Redirect("~/Start/Login");
            }
        }

        base.OnActionExecuting(filterContext);
    }
    catch (Exception)
    {
        filterContext.Result = new RedirectResult("~/Start/Login");
    }
}

```

6. Data: dentro se haya la clase Context que hereda de la clase DbContext, es la encargada de las llamadas a base de datos.

```

9 referencias
public Context():base("WorkFlowXConnection")
{
}
0 referencias
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
}
16 referencias
public DbSet<User> Users { get; set; }
14 referencias
public DbSet<Role> Roles { get; set; }
10 referencias
public DbSet<Company> Companies { get; set; }
0 referencias
public DbSet<Area> Areas { get; set; }
6 referencias
public DbSet<License> Licenses { get; set; }

public DbSet<Team> Teams { get; set; }

public DbSet<Project> Projects { get; set; }

```

7. Mappings: dentro se guarda el automapperConfig que relaciona los DTO con sus respectivos modelos.

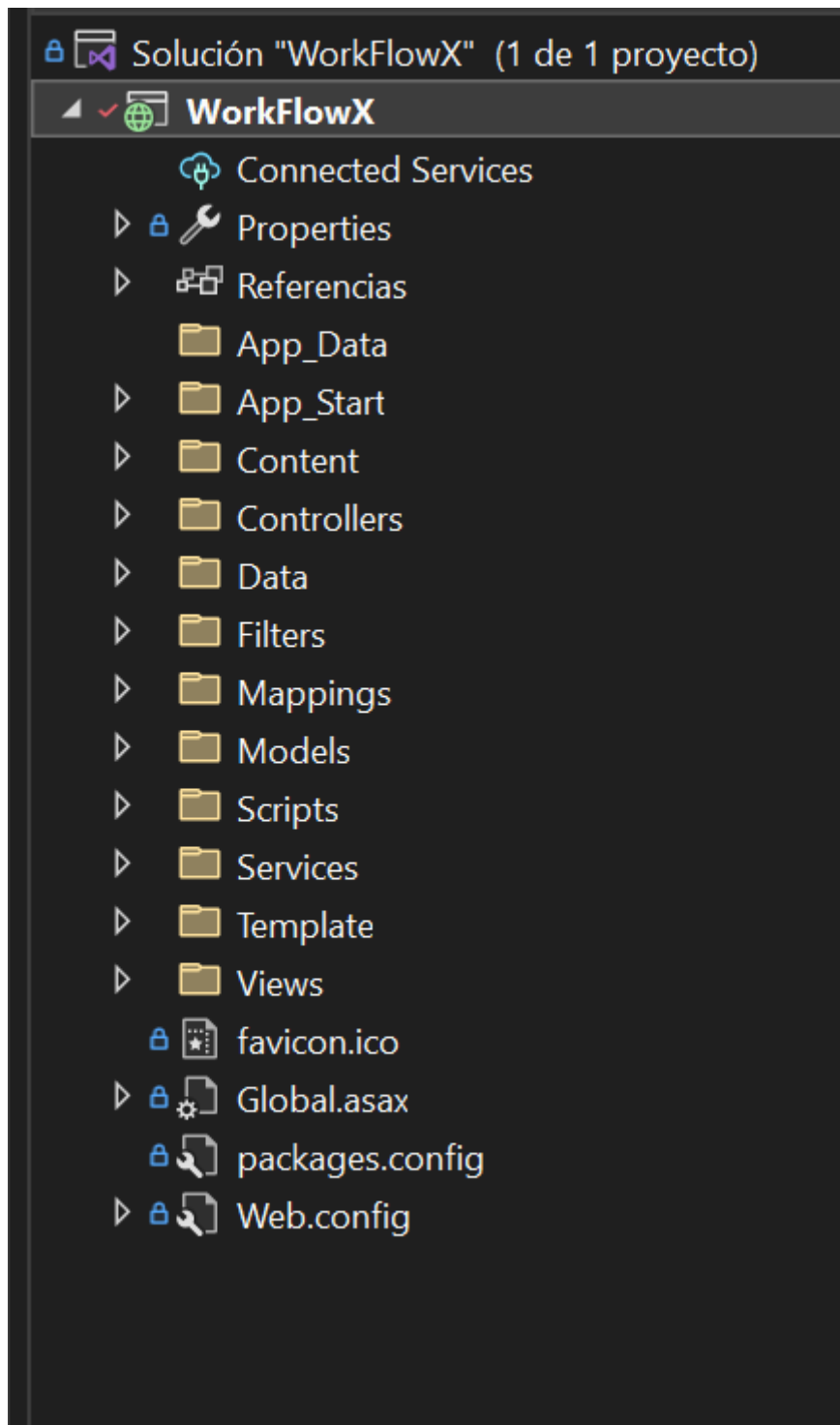
```

1 referencia
public static void Initialize()
{
    Mapper.Initialize(cfg =>
    {
        cfg.CreateMap<License, LicenseDTO>()
            .ForMember(dest => dest.AreaName, opt => opt.MapFrom(src => src.Area.Name));
        cfg.CreateMap<LicenseDTO, License>();

        cfg.CreateMap<Role, RoleDTO>();
        cfg.CreateMap<RoleDTO, Role>();
    });
}

```

Así se ve la estructura de los ficheros:



4. Diseño de la Interfaz de Usuario

Lo primero que ve el usuario es un formulario de login en el que deberá ingresar sus credenciales, por si acaso se ha olvidado de ellas hay una opción de recuperación de la contraseña donde se mandara un correo a la cuenta con la que se registro pidiendo ingresar una nueva contraseña.

Login

Correo

Contraseña

Ingresar

[Has olvidado tu contraseña?](#)

Una vez se ingrese se vera la vista principal de WorkFlowX donde en la barra lateral podremos acceder a nuestro Equipo, ver nuestras tareas y ver nuestros proyectos.

WORKFLOWX

Users Roles Companies Log out

Home

My Team


My Projects

My Tasks

WorkFlowX

Mié, 21. Jun 2023

08:38:30

Zaragoza  20°C

Mi

Ju

Vi

Sá

Do

Lu

Ma

25°C
19°C

28°C
18°C

29°C
16°C

33°C
19°C

35°C
21°C

34°C
20°C

37°C
20°C

Last Tasks

Name	Description	Start Date	End Date	Expected End Date	Status	Priority	Comment	Manager
Hacer cosicas	Pos eso, hacer cosicas	22/12/2023 0:00:00			Planning	Low		Daniel
Terminar	Pos lo de encima	21/06/2023			Pending	Medium		Daniel

En la zona superior se encuentran los apartados de gestión de los administradores, Gestión de usuarios, gestión de Roles y Gestión de compañías, todos siguiendo la misma estructura.

Users

Users

10

Apply

Name	Surname	Mail	Phone	NeedRestore	HasConfirm	Company	Role	Actions
Daniel	Tomas	danieltomasd2003@gmail.com	666666666	✗	✓	SALESIANOS	SUPERADMIN	<div><div></div><div></div><div></div></div>
SuperAdmin	prueba	superadmin@admin.com	111111111	✗	✓	SALESIANOS	SUPERADMIN	<div><div></div><div></div><div></div></div>
Admin	prueba	admin@admin.com	222222222	✗	✓	SALESIANOS	USERADMIN	<div><div></div><div></div><div></div></div>
Pablo	Villarte	pablovillarted2003@gmail.com	666666666	✗	✓	SALESIANOS	PRMANAGER	<div><div></div><div></div><div></div></div>
Marcos	Barcelona	marcosbarcelonad2003@gmail.com	666666666	✗	✓	SAN VALERO	PRMANAGER	<div><div></div><div></div><div></div></div>
Raul	Osuna	raulosunad2003@gmail.com	666666666	✗	✓	SAN VALERO	EDITOR	<div><div></div><div></div><div></div></div>

Roles

Roles

5

Apply

Name	Actions
SUPERADMIN	<div><div></div><div></div><div></div></div>
USERADMIN	<div><div></div><div></div><div></div></div>
PRMANAGER	<div><div></div><div></div><div></div></div>
EDITOR	<div><div></div><div></div><div></div></div>
USER	<div><div></div><div></div><div></div></div>

Licenses

Name
CREATE
EDIT
DETAILS

Companies

WORKFLOWX

[Users](#)[Roles](#)[Companies](#)[Log out](#)

Companies

+

5

Apply

Name	NIF	Address	Phone	Mail	Web	Country	City	Register Date	Actions
SALESIANOS	44556677	DON BOSCO	696969696	salesianos@salesianos.edu	salesianos.com	SPAIN	ZGZ	11/06/1995 0:00:00	<div><div></div><div></div><div></div></div>
SAN VALERO	N889654	C. San Valero	955643112	sanvalero@san.edu	sanvalero.com	SPAIN	MAD	13/09/1997 0:00:00	<div><div></div><div></div><div></div></div>
SAGRADO CORAZON	F864564567	C. Jisus	7567876556	sanvalero@fsbarat.org	jesusito.com	SPAIN	LEON	13/09/1997 0:00:00	<div><div></div><div></div><div></div></div>

1

Además podremos cerrar sesión desde el botón Log Out lo que nos devolverá a la vista de login.

Desde el apartado de My Team, podremos ver los datos de nuestro equipo así como sus integrantes.

WORKFLOWX

[Users](#)[Roles](#)[Companies](#)[Log out](#)

Home

My Team

My Projects

My Tasks

My Team

Your Team

Manager

Daniel

Name

.NET

Description

Equipo encargado de Aplicaciones con C#

Status

Active

Other Team Members

Name	Surname	Mail	Phone	Company	Role
Daniel	Tomas	danieltomasd2003@gmail.com	666666666	SALESIANOS	SUPERADMIN
SuperAdmin	prueba	superadmin@admin.com	111111111	SALESIANOS	SUPERADMIN
Admin	prueba	admin@admin.com	222222222	SALESIANOS	USERADMIN
Pablo	Villarte	pablovillarted2003@gmail.com	666666666	SALESIANOS	PRMANAGER

Desde el apartado de My projects podre ver todos los proyectos en los que participo así como las tareas adjuntas a los proyectos.

WORKFLOWX

[Users](#)[Roles](#)[Companies](#)[Log out](#)

Home

My Team

My Projects

My Tasks

My Projects

+

10

Apply

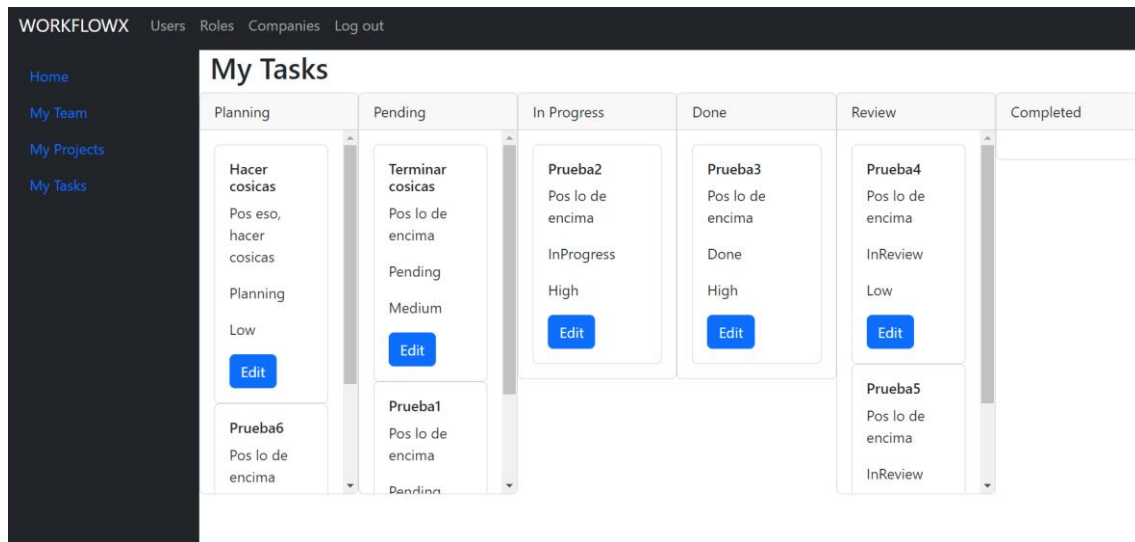
Name	Description	Start Date	End Date	Expected End Date	Budget	Actual Cost	Company	Actions
Cosicas	Cosicas de Salesians	21/06/2023 7:24:55		11/02/2011 0:00:00	150	1	SALESIANOS	<div><div></div><div></div><div></div></div>

1

Page 1 of 1

Total Projects: 1

Por ultimo desde el apartado de My Tasks vere en un panel Kanban todas las tareas según su estado y urgencia.



5 Tecnologías usadas.

Las tecnologías utilizadas en el desarrollo de WorkFlowX incluyen:

1. **Autmapper:** Es una biblioteca de mapeo objeto-relacional que simplifica la transferencia de datos entre diferentes modelos de objetos. Permite mapear automáticamente las propiedades de un objeto a otro, ahorrando tiempo y reduciendo la necesidad de escribir código manualmente.
2. **Auditoría de datos:** Esta funcionalidad permite realizar un seguimiento de los cambios realizados en los datos de la aplicación. Con la auditoría de datos, se registran y almacenan los cambios históricos, incluyendo quién realizó la modificación y cuándo. Esto es útil para realizar un seguimiento de la actividad y mantener un registro de las modificaciones realizadas.
3. **Acciones del CRUD:** CRUD es un acrónimo que significa Crear, Leer, Actualizar y Eliminar. WorkFlowX utiliza acciones del CRUD para proporcionar las operaciones básicas de gestión de datos, como la creación de proyectos y tareas, la lectura de información, la actualización de datos existentes y la eliminación de elementos no deseados.
4. **Roles y permisos:** WorkFlowX implementa un sistema de roles y permisos para gestionar el acceso y las funcionalidades disponibles para cada usuario. Los roles, como administrador, gerente de proyecto o miembro del equipo, determinan qué acciones y datos pueden ser accedidos por cada usuario. Los permisos se asignan a los roles para establecer las acciones específicas que pueden llevar a cabo.

5. Paginación y filtrado: Para mejorar la experiencia del usuario y la eficiencia en la visualización de datos, WorkFlowX utiliza técnicas de paginación y filtrado. La paginación divide grandes conjuntos de datos en páginas más pequeñas, lo que permite cargar y mostrar solo una cantidad limitada de registros a la vez. El filtrado permite al usuario especificar criterios para mostrar solo los datos relevantes según sus necesidades.
6. Estas tecnologías son utilizadas en WorkFlowX para mejorar la eficiencia, la seguridad y la experiencia del usuario al gestionar proyectos y tareas.