# MANUAL TÉCNICO

# UDAPP

## TABLA DE CONTENIDO

**Pág.**

## LISTA DE FIGURAS

**Pág.**

| REGISTRO DE CAMBIOS | | |
|---|---|---|
| **VERSIÓN** | **FECHA DE CAMBIO** | **MOTIVO DEL CAMBIO** |
| **1** | 29-11-2024 | Creación del documento |
| **2** | 05-11-2024 | Modificación |
| | | |

| CONTROL DEL DOCUMENTO | | | |
|---|---|---|---|
| **VERSIÓN** | **ELABORADO** | **REVISADO** | **APROBADO** |
| **1** | *UdApp* | | Oscar Gómez |
| | | | |
| | | | |
| | | | |

# 1 INTRODUCCIÓN

UDAPP, una plataforma de red social exclusiva para la comunidad universitaria. Esta herramienta digital tiene como objetivo principal fomentar la interacción, la colaboración y el intercambio de conocimientos entre los estudiantes, fortaleciendo así los lazos que los unen como comunidad.

Esta idea surge como una aliada indispensable para el aprendizaje, la investigación y la comunicación. Al centralizar las necesidades de la comunidad estudiantil en una única plataforma, se maximiza el potencial de las herramientas digitales y se promueve una experiencia educativa más integral.

Este manual técnico tiene como objetivo describir el desarrollo e implementación de UdApp, una plataforma de red social exclusiva para la comunidad universitaria de la Universidad de Cundinamarca. La creación de este manual responde a la necesidad de documentar de manera clara y completa el funcionamiento de la plataforma, facilitando así su uso, administración y mantenimiento.

# 2 OBJETIVO

Documentar de manera clara y completa el funcionamiento de la plataforma, facilitando así su uso, administración y mantenimiento por parte de administradores de sistemas, desarrolladores y usuarios finales.

# 3 ALCANCE

Este manual técnico tiene como alcance específico los siguientes aspectos:

- Descripción de los objetivos y funcionalidades de UdApp: Se detallan los objetivos generales y específicos de la plataforma, así como las funcionalidades disponibles para los usuarios.
- Detalles de la arquitectura técnica de la plataforma: Se describe la arquitectura tecnológica de UdApp, incluyendo los componentes hardware y software, la base de datos y las tecnologías utilizadas en su desarrollo.
- Políticas de seguridad y privacidad de datos: Se describen las políticas de seguridad y privacidad de datos de UdApp, así como las medidas implementadas para proteger la información de los usuarios.

# 4 GENERALIDADES

La aplicación UdApp está diseñada para gestionar la lógica del servidor, la conexión con la base de datos y la interfaz de usuario. La estructura de las páginas web se define con HTML, el estilo visual se implementa mediante CSS, y la interactividad se añade con JavaScript. Para asegurar una experiencia de usuario coherente en distintos dispositivos, se utiliza Bootstrap.

El frontend de la aplicación está construido con ASP.NET MVC y .NET, mientras que el backend se implementa en .NET Core, con APIs gestionadas por Swagger en Visual Studio Code 2019 y MySQL Server como base de datos. La arquitectura sigue el patrón Modelo-Vista-Controlador (MVC), y se apoya en Entity Framework y ADO.NET para facilitar la interacción con la base de datos. Para la autenticación, se utiliza JWT (JSON Web Tokens) y cookies, permitiendo una gestión segura y persistente de la sesión del usuario. Las sesiones también mantienen información durante la navegación, y un sistema de autenticación y autorización regula el acceso a las distintas funcionalidades.

Roles de usuario en la aplicación:

- Estudiante: Puede ver, crear y administrar publicaciones, interactuar con otros usuarios, actualizar su perfil.
- Administrativos: Tienen acceso para ver, crear y gestionar publicaciones, así como interactuar con otros usuarios y actualizar su perfil.
- Gerente: Generar reportes y gestionar usuarios.
- Moderador: Puede ver, crear y administrar publicaciones, tiene permisos para moderar contenido, eliminar publicaciones y reportes, suspender usuarios inapropiados.

# 5 PRERREQUISITOS DEL SISTEMA

## 5.1 HARDWARE

- Servidor con las siguientes características:
  - Procesador 12th Generación Intel core i5
  - RAM 8 GB
- Conexión a internet estable.

## 5.2 SOFTWARE

**Programas**: Visual studio 2019
SQL SMS
**Navegador**: Los navegadores compatibles incluyen:
Google Chrome: Última versión disponible.
Microsoft Edge: Última versión disponible.

**JavaScript:**
Es necesario tener habilitado JavaScript en el navegador para poder disfrutar de la interactividad y las funcionalidades completas de la página web.

**Lector de PDF:**

Si bien la aplicación UdApp no se basa principalmente en archivos PDF, es posible que algunos documentos o recursos se presenten en este formato.

# 6 FRAMEWORK Y ESTÁNDARES

## 6.1 FRAMEWORKS:

El framework utilizado en el proyecto UdApp es ASP.NET MVC (Model-View-Controller), un framework de desarrollo web. Se basa en el patrón de arquitectura MVC, que separa la aplicación en tres componentes principales:

Modelo (Model): Representa los datos de la aplicación y las reglas de negocio.
Vista (View): Define la presentación de los datos al usuario, generalmente en forma de páginas web HTML.
Controlador (Controller): Maneja la interacción del usuario con la aplicación, procesa las solicitudes y actualiza el modelo y la vista en consecuencia.

ASP.NET MVC ofrece varias ventajas para el desarrollo de aplicaciones web, incluyendo:

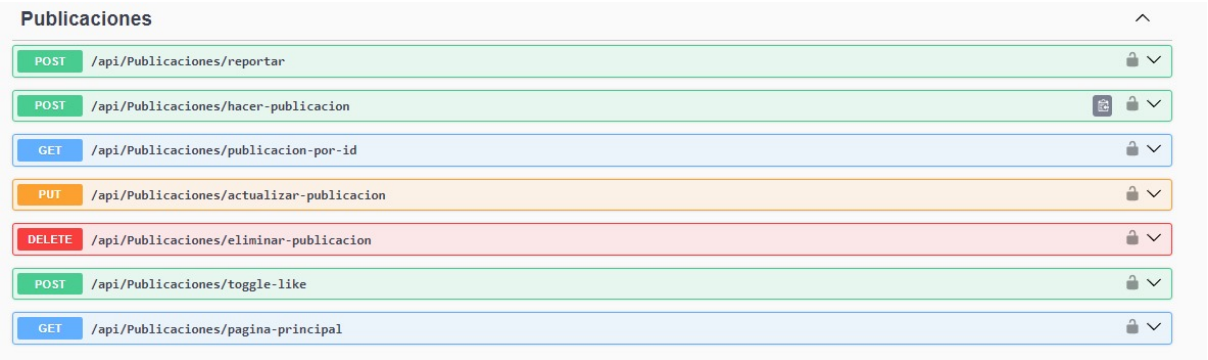
- Separación de preocupaciones: Al separar la aplicación en tres componentes independientes, facilita el desarrollo, mantenimiento y prueba del código.
- Enrutamiento de URL flexible: Permite definir reglas para mapear las URL a las acciones del controlador.
- Soporte para pruebas unitarias: Facilita la escritura y ejecución de pruebas unitarias para el código del controlador y la vista.

Integración con otras tecnologías de ASP.NET: Se integra fácilmente con otras tecnologías de ASP.NET, como Web Forms y Web API.Nombre, versión y una breve descripcion

## 6.2 ESTANDARES:

### API

Al documentar la API con Swagger, el frontend tiene acceso a una interfaz gráfica donde puede ver todos los endpoints disponibles y probarlos de forma directa, lo que reduce los tiempos de prueba y facilita el desarrollo de la integración. Además, Swagger permite probar la API sin necesidad de herramientas externas, ya que la misma interfaz genera y envía las solicitudes, permitiendo a los desarrolladores ver los resultados en tiempo real.




### ENCRIPTACIÓN

```
Tabnine | 3 referencias
public string EncriptarContrasenaArgon2(string contrasena)
{
    var argon2 = new Argon2id(Encoding.UTF8.GetBytes(contrasena));
    argon2.Salt = new byte[16]; //salt aleatoria
    argon2.DegreeOfParallelism = 4; //grado de paralelismo
    argon2.MemorySize = 65536; //tamaño de memoria en kilobytes
    argon2.Iterations = 4; //numero de iteraciones
    byte[] hashBytes = argon2.GetBytes(32); //tamaño del hash en bytes
    string hashString = Convert.ToBase64String(hashBytes);
    return hashString;
}
```

### Descripción del código

- **Definición de Argon2id** : Se crea una instancia de Argon2id, aplicando el método Encoding.UTF8.GetBytesa la contraseña para convertirla en un arreglo de bytes.

- **Salt**: Se define Saltcomo un arreglo de 16 bytes. Aunque el valor de Saltestá configurado, se recomienda generar una sal aleatoria para cada contraseña.
- **Parámetros de configuración del algoritmo Argon2id** :
    - DegreeOfParallelism: Establecido en 4, define el grado de paralelismo, es decir, el número de hilos que se usarán durante el procesamiento.
    - MemorySize: Configurado en 65536 KB, representa la cantidad de memoria que el algoritmo utilizará durante el proceso de hash.
    - Iterations: Configurado en 4, indica el número de iteraciones o rondas de procesamiento que realizará el algoritmo.
- **Hashing** : El método GetBytes(32)genera un hash de 32 bytes.
- **Conversión y Retorno**: Finalmente, el hash en bytes se convierte a una cadena en formato Base64 (ToBase64String) para facilitar su almacenamiento o transporte. La función devuelve esta cadena (hashString), que es el resultado cifrado de la contraseña.

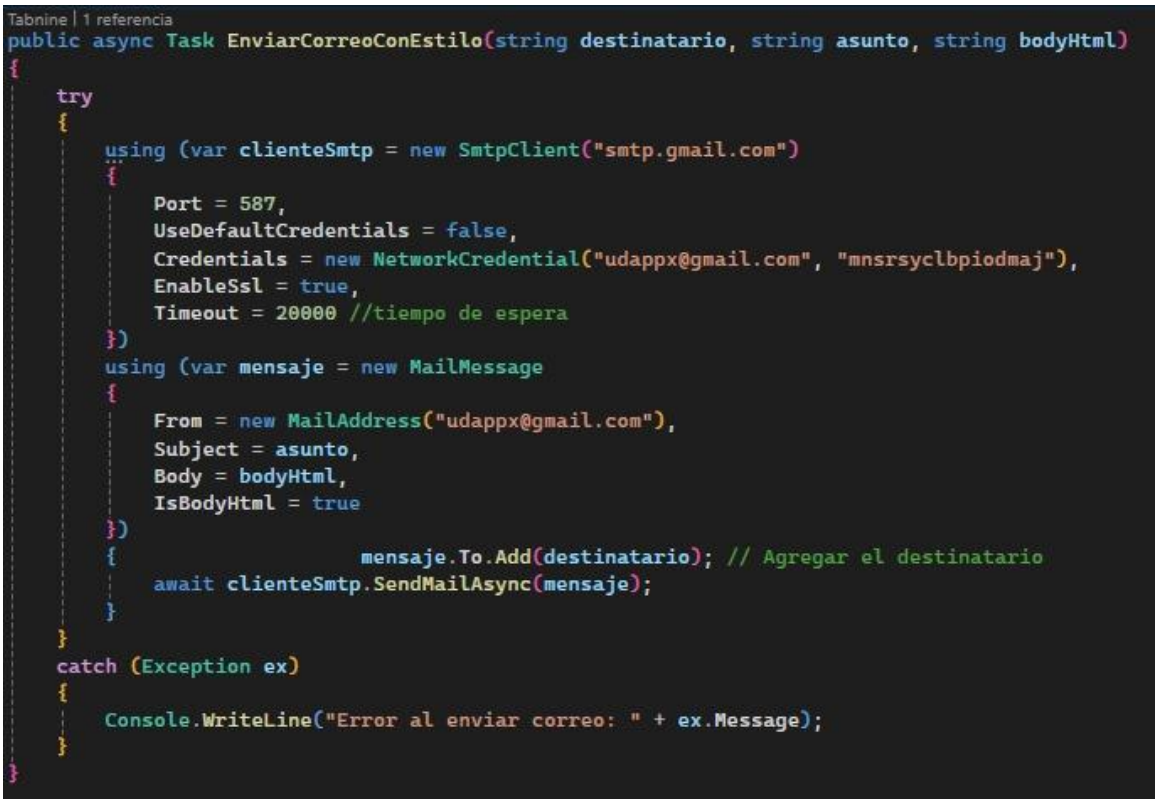
**Parámetros clave y su propósito:**

- **Salt**: Añade aleatoriedad al hash, haciendo que incluso contraseñas idénticas generen hashes diferentes.
- **DegreeOfParallelism** : Aumenta la seguridad al permitir el uso de múltiples hilos.
- **MemorySize** : Cuanta más memoria utilice, más costoso será para un atacante descifrar el hash.
- **Iteraciones**: Aumenta la complejidad del hash al aplicar varias rondas de procesamiento.

Este método de hash es adecuado para almacenar contraseñas de manera segura, dificultando los ataques de fuerza bruta y de diccionario.

**ENVIAR CORREO**

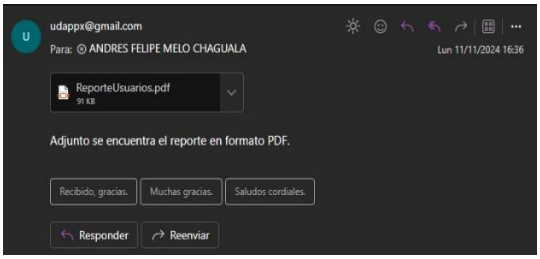
EnviarCorreoConEstilo, envía un correo HTML al destinatario especificado con el asunto y contenido en formato HTML (en bodyHtml). Configura un cliente SMTP con autenticación, puerto seguro y encriptación SSL.



```
Tabnine | 1 referencia
public async Task EnviarCorreoConEstilo(string destinatario, string asunto, string bodyHtml)
{
    try
    {
        using (var clienteSmtp = new SmtpClient("smtp.gmail.com")
        {
            Port = 587,
            UseDefaultCredentials = false,
            Credentials = new NetworkCredential("udappx@gmail.com", "mnsrsyclbpiodmaj"),
            EnableSsl = true,
            Timeout = 20000 //tiempo de espera
        })
        using (var mensaje = new MailMessage
        {
            From = new MailAddress("udappx@gmail.com"),
            Subject = asunto,
            Body = bodyHtml,
            IsBodyHtml = true
        })
        {                   mensaje.To.Add(destinatario); // Agregar el destinatario
            await clienteSmtp.SendMailAsync(mensaje);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al enviar correo: " + ex.Message);
    }
}
```

EnviarCorreoConPDFAdjunto, además del mensaje HTML, permite adjuntar un archivo PDF generado a partir de un arreglo de bytes (pdfBytes) con el nombre especificado (nombrepdf). Ambas funciones manejan posibles errores al enviar el correo e imprimen un mensaje si ocurre una excepción.

```
Tabnine | 3 referencias
public async Task EnviarCorreoConPDFAdjunto(string destinatario, string asunto, string body, byte[] pdfBytes, string nombrepdf)
{
    try
    {
        using (var clieneSmtp = new SmtpClient("smtp.gmail.com")
        {
            Port = 587,
            UseDefaultCredentials = false,
            Credentials = new NetworkCredential("udappx@gmail.com", "mnsrsyclbpiodmaj"),
            EnableSsl = true,
        })
        using (var mailMessage = new MailMessage
        {
            From = new MailAddress("udappx@gmail.com"),
            Subject = asunto,
            Body = body,
            IsBodyHtml = true,
        })
        {
            mailMessage.To.Add(destinatario);

            // Crear el adjunto desde el PDF en bytes
            using (var ms = new MemoryStream(pdfBytes))
            {
                var attachment = new Attachment(ms, $"{nombrepdf}.pdf", MediaTypeNames.Application.Pdf);
                mailMessage.Attachments.Add(attachment);

                await clieneSmtp.SendMailAsync(mailMessage);
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al enviar correo: " + ex.Message);
    }
}
```


**Reporte en pdf**

# 7 DIAGRAMAS DE CASOS DE USO

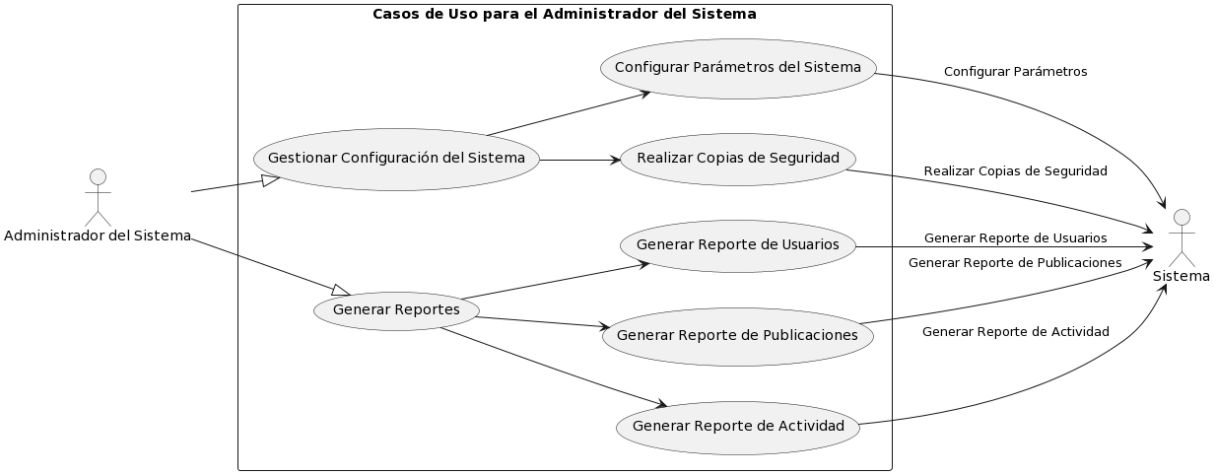



*Figura 1 Caso de uso Administrador del sistema*

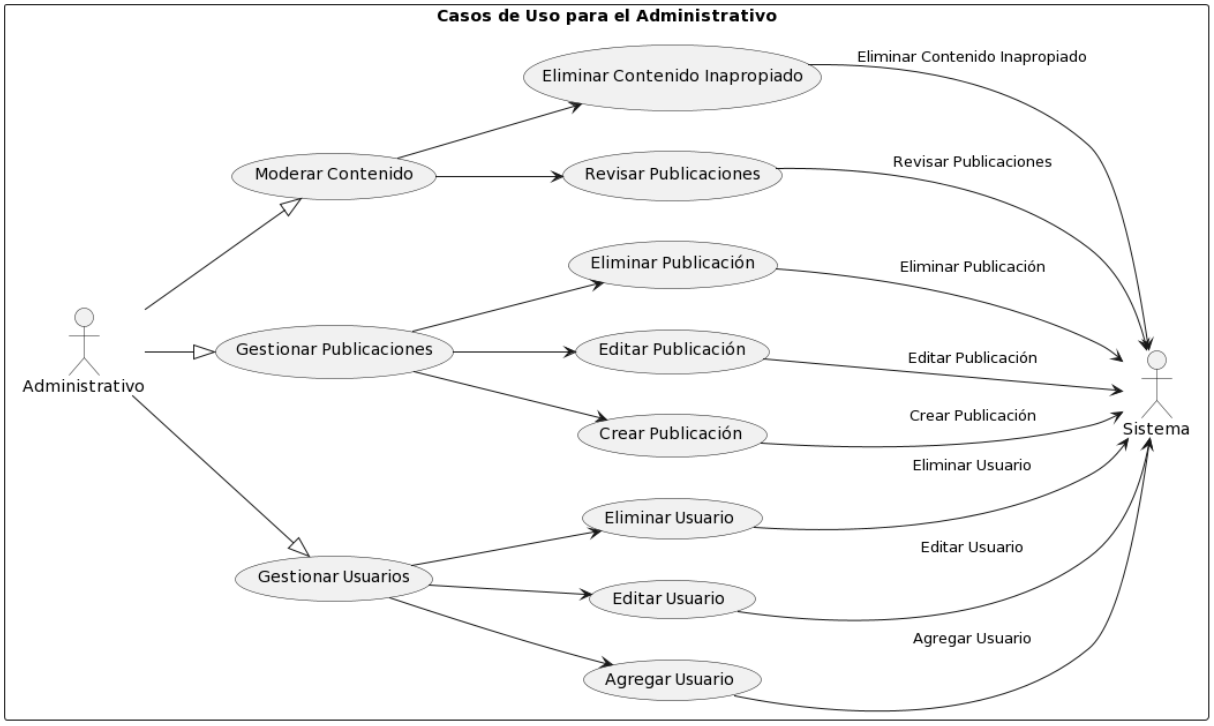



*Figura 2 Caso de uso Administrativo*

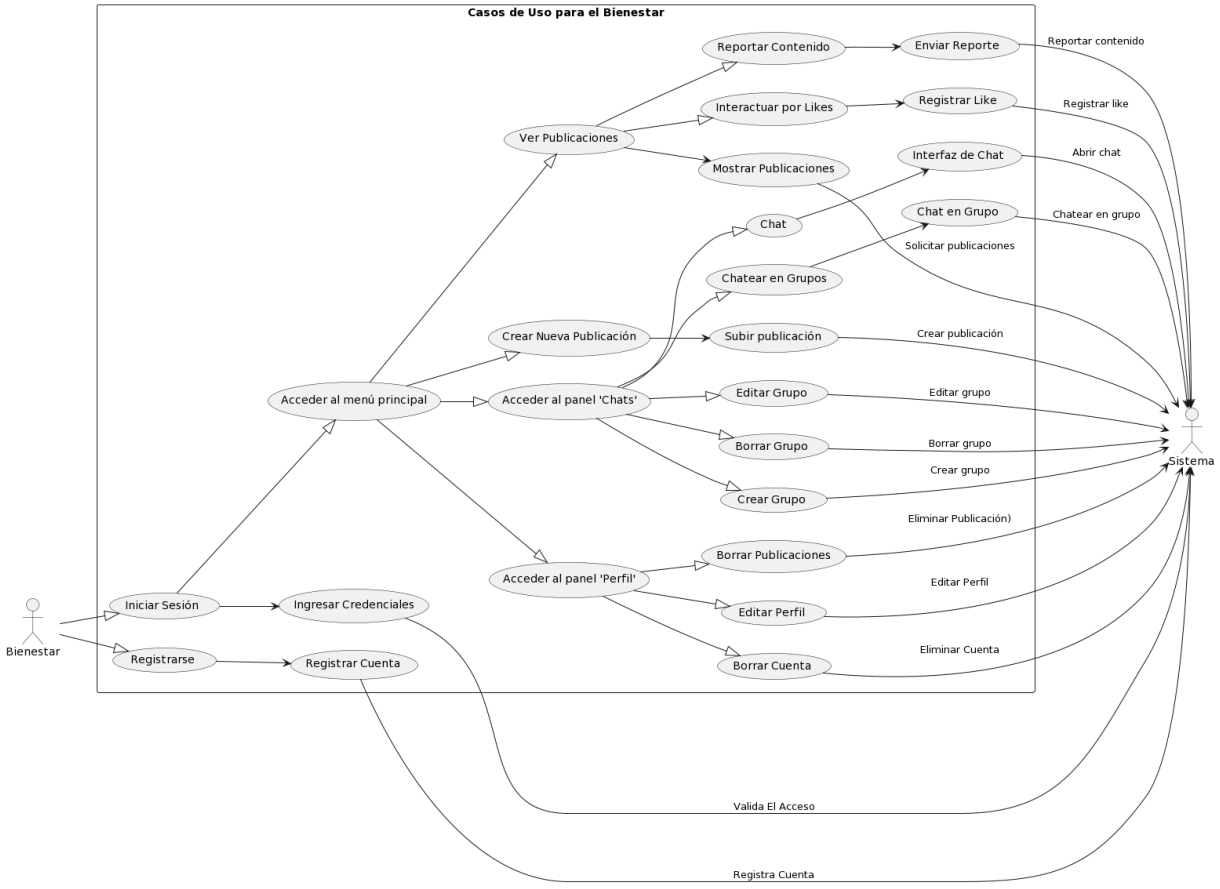



*Figura 3 Caso de uso  Bienestar*

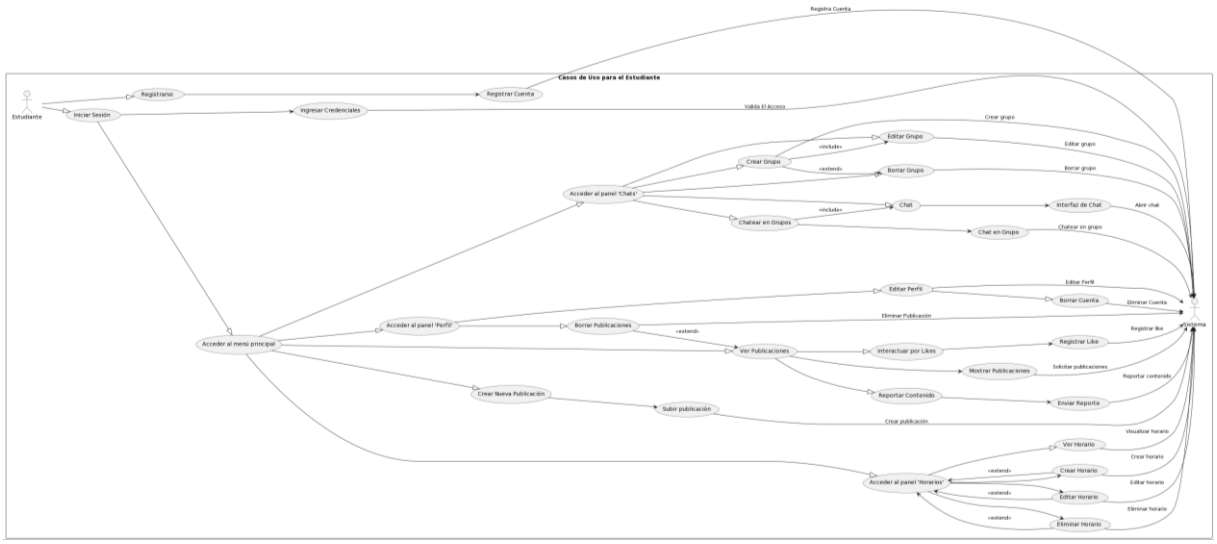



*Figura 4 Caso de uso Estudiante*

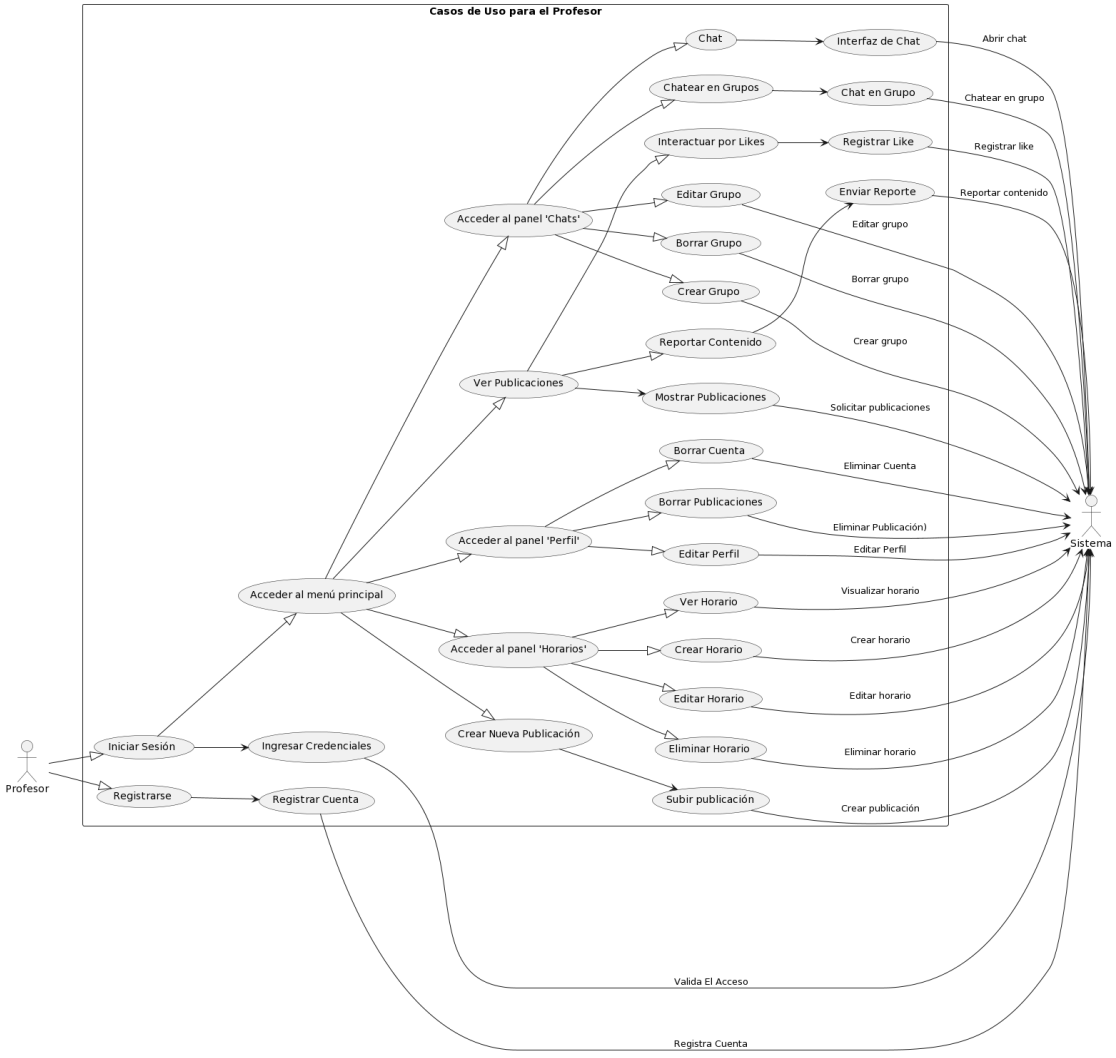



*Figura 5 Caso de uso Profesor*

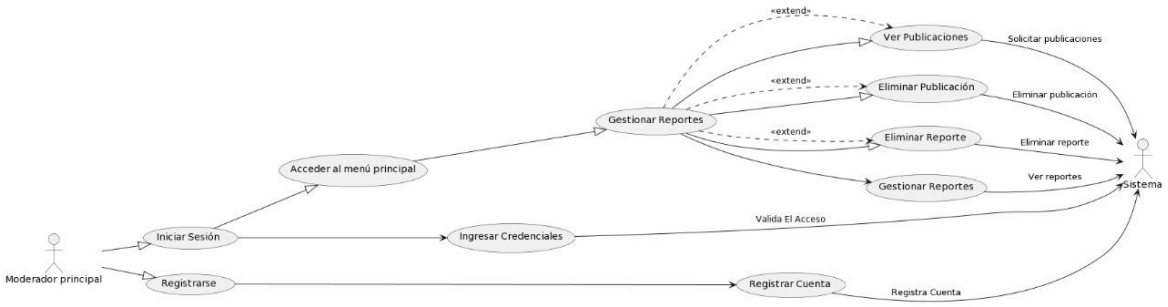



*Figura 6 Caso de uso Moderador*

# 8 MODELO ENTIDAD RELACIÓN


*Figura 7 MER*

# 9 DICCIONARIO DE DATOS DEL SISTEMA

| | REPORT | ServerName | DatabaseName | TableName | SchemaName | ColumnName | DataType | MaxLength | IsNull | IsIdentity | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | idAuditoria | int | 4 | NO | YES | Identificador único de cada registro en la tabla de au... |
| 2 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | nombreTabla | nvarchar | 200 | NO | NO | Nombre de la tabla donde ocurrió la operación |
| 3 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | tipoOperacion | nvarchar | 20 | NO | NO | Tipo de operación realizada (INSERT, UPDATE, DE... |
| 4 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | idRegistro | int | 4 | NO | NO | Identificador del registro afectado en la tabla auditada |
| 5 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | columna | nvarchar | 200 | YES | NO | Nombre de la columna afectada por la operación |
| 6 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | valorAntiguo | nvarchar | -1 | YES | NO | Valor antiguo de la columna antes de la operación |
| 7 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | valorNuevo | nvarchar | -1 | YES | NO | Nuevo valor de la columna después de la operación |
| 8 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | usuario | nvarchar | 200 | NO | NO | Usuario que realizó la operación |
| 9 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | AUDITORIA | dbo | fechaOperacion | datetime | 8 | YES | NO | Fecha y hora en que se realizó la operación |
| 10 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | CHAT | dbo | idChat | int | 4 | NO | NO | Identificador único del chat |
| 11 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | CHAT | dbo | fk_idUsuarioEmisor | int | 4 | YES | NO | Identificador del usuario emisor del chat |
| 12 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | CHAT | dbo | fk_idUsuarioRecept... | int | 4 | YES | NO | Identificador del usuario receptor del chat |
| 13 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | CHAT | dbo | contenido | text | 16 | YES | NO | Contenido del mensaje en el chat |
| 14 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | CHAT | dbo | fechaEnvio | datetime | 8 | YES | NO | Fecha de envío del mensaje en el chat |
| 15 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | COMENT... | dbo | idComentario | int | 4 | NO | YES | Identificador único del comentario |
| 16 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | COMENT... | dbo | contenido | nvarchar | 400 | YES | NO | Contenido del comentario |
| 17 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | COMENT... | dbo | fechaComentario | datetime | 8 | YES | NO | Fecha del comentario |
| 18 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | COMENT... | dbo | fk_idUsuarioComen... | int | 4 | NO | NO | Identificador del usuario que hizo el comentario |
| 19 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | COMENT... | dbo | fk_idPublicacion | int | 4 | NO | NO | Identificador de la publicación comentada |
| 20 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | idGrupo | int | 4 | NO | NO | Identificador único del grupo |
| 21 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | nombreGrupo | varchar | 100 | YES | NO | Nombre del grupo |
| 22 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | descripcion | text | 16 | YES | NO | Descripción del grupo |
| 23 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | fechaCreacion | date | 3 | YES | NO | Fecha de creación del grupo |
| 24 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | fk_idUsuarioCreador | int | 4 | YES | NO | Identificador del usuario creador del grupo |
| 25 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | GRUPO | dbo | Estado | varchar | 20 | YES | NO | Estado del grupo |
| 26 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | HORARIO | dbo | idHorario | int | 4 | NO | NO | Identificador único del horario |
| 27 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | HORARIO | dbo | diaSemana | varchar | 20 | YES | NO | Día de la semana |
| 28 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | HORARIO | dbo | horaInicio | time | 5 | YES | NO | Hora de inicio del horario |
| 29 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | HORARIO | dbo | horaFin | time | 5 | YES | NO | Hora de fin del horario |
| 30 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | HORARIO | dbo | fk_idUsuario | int | 4 | YES | NO | Identificador del usuario asociado al horario |
| 31 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | LIKES | dbo | id | int | 4 | NO | YES | Identificador único del like |
| 32 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | LIKES | dbo | fk_idUsuario | int | 4 | NO | NO | Identificador del usuario que dio el like |
| 33 | DATADICTIONA... | SQL9001 | db_aaf489_udapp... | LIKES | dbo | fk_idPublicacion | int | 4 | NO | NO | Identificador de la publicación a la que se dio like |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | LIKES | dbo | created_at | datetime | 8 | YES | NO | Fecha en la que se dio el like |
| 35 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | idPublicacion | int | 4 | NO | YES | Identificador único de la publicación |
| 36 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | titulo | varchar | 100 | YES | NO | Título de la publicación |
| 37 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | contenido | text | 16 | YES | NO | Contenido de la publicación |
| 38 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | fechaPublicacion | datetime | 8 | YES | NO | Fecha de publicación |
| 39 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | fk_idUsuarioPublic… | int | 4 | YES | NO | Identificador del usuario que publicó |
| 40 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | likes | int | 4 | YES | NO | Número de likes de la publicación |
| 41 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | comentarios | int | 4 | YES | NO | Número de comentarios de la publicación |
| 42 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | PUBLICA… | dbo | reportada | bit | 1 | YES | NO | Indica si la publicación está reportada |
| 43 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | REPORTE | dbo | idReporte | int | 4 | NO | YES | Identificador único del reporte |
| 44 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | REPORTE | dbo | motivoReporte | varchar | 50 | YES | NO | Motivo del reporte |
| 45 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | REPORTE | dbo | fechaReporte | datetime | 8 | YES | NO | Fecha del reporte |
| 46 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | REPORTE | dbo | fk_idUsuarioReport… | int | 4 | YES | NO | Identificador del usuario que reportó |
| 47 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | REPORTE | dbo | fk_idPublicacionRe… | int | 4 | YES | NO | Identificador de la publicación reportada |
| 48 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | ROL | dbo | idRol | int | 4 | NO | YES | Identificador único del rol |
| 49 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | ROL | dbo | nombreRol | nvarchar | 100 | YES | NO | Nombre del rol |
| 50 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | ROL | dbo | permisos | nvarchar | 100 | YES | NO | Permisos asociados al rol |
| 51 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | idUsuario | int | 4 | NO | YES | Identificador único del usuario |
| 52 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | cedulaUsuario | nvarchar | 100 | YES | NO | Cédula del usuario |
| 53 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | nombreUsuario | varchar | 50 | YES | NO | Nombre del usuario |
| 54 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | apellidoUsuario | nvarchar | 100 | YES | NO | Apellido del usuario |
| 55 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | telefono | nvarchar | 100 | YES | NO | Teléfono del usuario |
| 56 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | direccion | nvarchar | 200 | YES | NO | Dirección del usuario |
| 57 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | email | varchar | 100 | YES | NO | Correo electrónico del usuario |
| 58 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | contrasena | varchar | 50 | YES | NO | Contraseña del usuario |
| 59 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | fk_idRol | int | 4 | YES | NO | Identificador del rol del usuario |
| 60 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | estadoSuspension | bit | 1 | YES | NO | Indica si el usuario está suspendido |
| 61 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO | dbo | fechaRegistro | datetime | 8 | YES | NO | Fecha de registro del usuario |
| 62 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO… | dbo | fk_idUsuario | int | 4 | NO | NO | Identificador del usuario en la relación usuario-grupo |
| 63 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO… | dbo | fk_idGrupo | int | 4 | NO | NO | Identificador del grupo en la relación usuario-grupo |
| 64 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO… | dbo | fk_idUsuario | int | 4 | NO | NO | Identificador del usuario en la relación usuario-publi… |
| 65 | DATADICTIONA… | SQL9001 | db_aaf489_udapp… | USUARIO… | dbo | fk_idPublicacion | int | 4 | NO | NO | Identificador de la publicación en la relación usuario… |

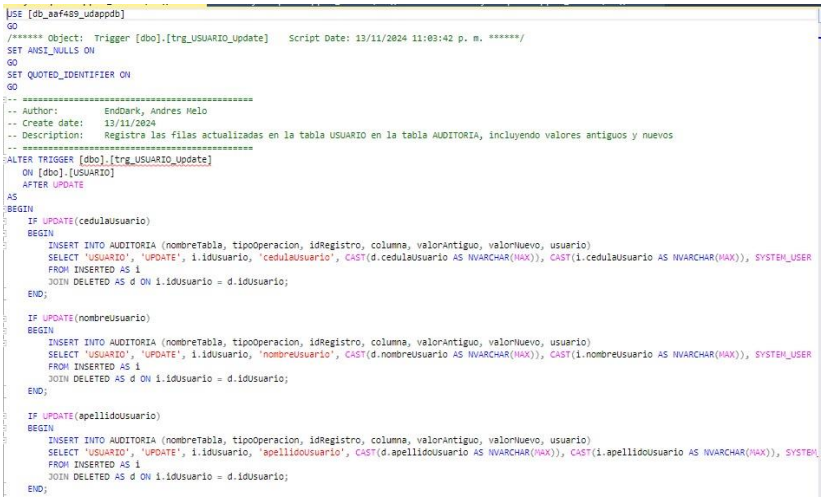
# 10 ANEXOS

## TRIGGERS

Registra las filas eliminadas en la tabla USUARIO en la tabla AUDITORIA.

```
USE [db_aaf489_udappdb]
GO
/****** Object:  Trigger [dbo].[trg_USUARIO_Delete]    Script Date: 13/11/2024 11:02:27 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:         EndDark, Andres Melo
-- Create date:    13/11/2024
-- Description:    Registra las filas eliminadas en la tabla USUARIO en la tabla AUDITORIA
-- =============================================
ALTER TRIGGER [dbo].[trg_USUARIO_Delete]
   ON [dbo].[USUARIO]
   AFTER DELETE
AS
BEGIN
    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'cedulaUsuario', CAST(d.cedulaUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'nombreUsuario', CAST(d.nombreUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'apellidoUsuario', CAST(d.apellidoUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'telefono', CAST(d.telefono AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'direccion', CAST(d.direccion AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, usuario)
    SELECT 'USUARIO', 'DELETE', d.idUsuario, 'email', CAST(d.email AS NVARCHAR(MAX)), SYSTEM_USER
    FROM DELETED AS d;
```

Registra las filas actualizadas en la tabla USUARIO en la tabla AUDITORIA, incluyendo valores antiguos y nuevos



```
USE [db_aaf489_udappdb]
GO
/****** Object:  Trigger [dbo].[trg_USUARIO_Update]    Script Date: 13/11/2024 11:03:42 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          EndDark, Andres Melo
-- Create date:     13/11/2024
-- Description:     Registra las filas actualizadas en la tabla USUARIO en la tabla AUDITORIA, incluyendo valores antiguos y nuevos
-- =============================================
ALTER TRIGGER [dbo].[trg_USUARIO_Update]
   ON [dbo].[USUARIO]
   AFTER UPDATE
AS
BEGIN
    IF UPDATE(cedulaUsuario)
    BEGIN
        INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, valorNuevo, usuario)
        SELECT 'USUARIO', 'UPDATE', i.idUsuario, 'cedulaUsuario', CAST(d.cedulaUsuario AS NVARCHAR(MAX)), CAST(i.cedulaUsuario AS NVARCHAR(MAX)), SYSTEM_USER
        FROM INSERTED AS i
        JOIN DELETED AS d ON i.idUsuario = d.idUsuario;
    END;

    IF UPDATE(nombreUsuario)
    BEGIN
        INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, valorNuevo, usuario)
        SELECT 'USUARIO', 'UPDATE', i.idUsuario, 'nombreUsuario', CAST(d.nombreUsuario AS NVARCHAR(MAX)), CAST(i.nombreUsuario AS NVARCHAR(MAX)), SYSTEM_USER
        FROM INSERTED AS i
        JOIN DELETED AS d ON i.idUsuario = d.idUsuario;
    END;

    IF UPDATE(apellidoUsuario)
    BEGIN
        INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorAntiguo, valorNuevo, usuario)
        SELECT 'USUARIO', 'UPDATE', i.idUsuario, 'apellidoUsuario', CAST(d.apellidoUsuario AS NVARCHAR(MAX)), CAST(i.apellidoUsuario AS NVARCHAR(MAX)), SYSTEM_
        FROM INSERTED AS i
        JOIN DELETED AS d ON i.idUsuario = d.idUsuario;
    END;
```

Registra las filas insertadas en la tabla USUARIO en la tabla AUDITORIA

```
USE [db_aaf489_udappdb]
GO
/****** Object:  Trigger [dbo].[trg_USUARIO_Insert]    Script Date: 13/11/2024 11:00:36 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          EndDark, Andres Melo
-- Create date:     13/11/2024
-- Description:     Registra las filas insertadas en la tabla USUARIO en la tabla AUDITORIA
-- =============================================
ALTER TRIGGER [dbo].[trg_USUARIO_Insert]
   ON [dbo].[USUARIO]
   AFTER INSERT
AS
BEGIN
    -- Para cada columna de la tabla USUARIO, insertamos un registro en la tabla de auditoría
    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorNuevo, usuario)
    SELECT 'USUARIO', 'INSERT', i.idUsuario, 'cedulaUsuario', CAST(i.cedulaUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM INSERTED AS i;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorNuevo, usuario)
    SELECT 'USUARIO', 'INSERT', i.idUsuario, 'nombreUsuario', CAST(i.nombreUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM INSERTED AS i;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorNuevo, usuario)
    SELECT 'USUARIO', 'INSERT', i.idUsuario, 'apellidoUsuario', CAST(i.apellidoUsuario AS NVARCHAR(MAX)), SYSTEM_USER
    FROM INSERTED AS i;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorNuevo, usuario)
    SELECT 'USUARIO', 'INSERT', i.idUsuario, 'telefono', CAST(i.telefono AS NVARCHAR(MAX)), SYSTEM_USER
    FROM INSERTED AS i;

    INSERT INTO AUDITORIA (nombreTabla, tipoOperacion, idRegistro, columna, valorNuevo, usuario)
    SELECT 'USUARIO', 'INSERT', i.idUsuario, 'direccion', CAST(i.direccion AS NVARCHAR(MAX)), SYSTEM_USER
    FROM INSERTED AS i;
```

Se ejecutará después de que se inserte un nuevo registro en la tabla REPORTE, este trigger actualizara la columna reportada en la tabla PUBLICACION a 1 para la publicación especificada en el fk_idPublicacionReportada del nuevo reporte.

```
USE [db_aaf489_udappdb]
GO
/****** Object:  Trigger [dbo].[trg_AfterInsertReporte]    Script Date: 13/11/2024 11:05:58 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:      <EndDark,Andres Melo>
-- Create date: <10,11,2024>
-- Description: <se ejecutará después de que se
-- inserte un nuevo registro en la tabla REPORTE.
-- Este trigger actualizará la columna reportada
-- en la tabla PUBLICACION a 1 para la
-- publicación especificada en el
-- fk_idPublicacionReportada del nuevo reporte.>
-- =============================================
ALTER TRIGGER [dbo].[trg_AfterInsertReporte]
ON [dbo].[REPORTE]
AFTER INSERT
AS
BEGIN
    DECLARE @fk_idPublicacionReportada INT;

    -- Obtenemos el fk_idPublicacionReportada del nuevo reporte insertado
    SELECT @fk_idPublicacionReportada = fk_idPublicacionReportada FROM INSERTED;

    -- Actualizamos la columna 'reportada' en la tabla PUBLICACION a 1 para la publicación correspondiente
    UPDATE PUBLICACION
    SET reportada = 1
    WHERE idPublicacion = @fk_idPublicacionReportada;
END;
```

Verifica si la publicación que esta siendo eliminada tiene otros reportes, si no existen mas reportes asociados, actualiza la columna reportada en la tabla PUBLICACION A 0 PARA ESTA PUBLICACION.

```
USE [db_aaf489_udappdb]
GO
/****** Object:  Trigger [dbo].[trg_AfterDeleteReporte]    Script Date: 13/11/2024 11:05:38 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:      <EndDark,Andres Melo>
-- Create date: <10,11,2024>
-- Description: <verifica si la publicación que
-- está siendo eliminada tiene otros reportes.
-- Si no existen más reportes asociados,
-- actualiza la columna reportada en la tabla
-- PUBLICACION a 0 para esa publicación.>
-- =============================================
ALTER TRIGGER [dbo].[trg_AfterDeleteReporte]
ON [dbo].[REPORTE]
AFTER DELETE
AS
BEGIN
    DECLARE @fk_idPublicacionReportada INT;

    -- Obtenemos el fk_idPublicacionReportada del reporte que se está eliminando
    SELECT @fk_idPublicacionReportada = fk_idPublicacionReportada FROM DELETED;

    -- Verificamos si quedan otros reportes asociados a la misma publicación
    IF NOT EXISTS (
        SELECT 1
        FROM REPORTE
        WHERE fk_idPublicacionReportada = @fk_idPublicacionReportada
    )
    BEGIN
        -- Si no quedan más reportes, actualizamos la columna 'reportada' de la tabla PUBLICACION
        UPDATE PUBLICACION
        SET reportada = 0
        WHERE idPublicacion = @fk_idPublicacionReportada;
    END
END;
```

STORAGE PROCEDURE

Elimina una publicación y sus reportes asociados.

```
USE [db_aaf489_udappdb]
GO
/****** Object:  StoredProcedure [dbo].[sp_EliminarPublicacion]    Script Date: 14/11/2024 9:34:29 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          EndDark, Andres Melo
-- Create date:     10/11/2024
-- Description:     Elimina una publicación y sus reportes asociados.
-- =============================================
ALTER PROCEDURE [dbo].[sp_EliminarPublicacion]
    @IdPublicacion INT,
    @IdUsuario INT,
    @Resultado BIT OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @Resultado = 0;

    BEGIN TRY
        DELETE FROM COMENTARIO WHERE fk_idPublicacion = @IdPublicacion;
        DELETE FROM REPORTE WHERE fk_idPublicacionReportada = @IdPublicacion;
        DELETE FROM PUBLICACION WHERE idPublicacion = @IdPublicacion AND fk_idUsuarioPublicador = @IdUsuario;

        -- Si la publicación se eliminó, se establece el resultado como éxito
        IF @@ROWCOUNT > 0
            SET @Resultado = 1;
    END TRY
    BEGIN CATCH
        SET @Resultado = 0;
    END CATCH;
END
```

Insertar una nueva publicación en la tabla PUBLICACION

```
USE [db_aaf489_udappdb]
GO
/****** Object:  StoredProcedure [dbo].[sp_CrearPublicacion]    Script Date: 14/11/2024 9:34:20 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          EndDark, Andres Melo
-- Create date:     10/11/2024
-- Description:     Inserta una nueva publicación en la tabla PUBLICACION.
-- =============================================
ALTER PROCEDURE [dbo].[sp_CrearPublicacion]
    @Titulo NVARCHAR(255),
    @FechaPublicacion DATETIME,
    @IdUsuarioPublicador INT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO PUBLICACION (titulo, fechaPublicacion, fk_idUsuarioPublicador, comentarios, likes)
        VALUES (@Titulo, @FechaPublicacion, @IdUsuarioPublicador, 0, 0);
    END TRY
    BEGIN CATCH
        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();
        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END
```

Obtiene una publicación por su id

```
USE [db_aaf489_udappdb]
GO
/****** Object:  StoredProcedure [dbo].[sp_ObtenerPublicacionPorId]    Script Date: 14/11/2024 9:34:39 p. m. ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          EndDark, Andres Melo
-- Create date:     10/11/2024
-- Description:     Obtiene una publicación por su Id.
-- =============================================
ALTER PROCEDURE [dbo].[sp_ObtenerPublicacionPorId]
    @Id INT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        SELECT [idPublicacion], [titulo], [contenido], [fechaPublicacion], [fk_idUsuarioPublicador],
               [likes], [comentarios], [reportada]
        FROM PUBLICACION
        WHERE idPublicacion = @Id;
    END TRY
    BEGIN CATCH
        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();
        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END
```