# **Formulario: Vista Lógica del Proyecto de Software**

**Proyecto:** NewsEye  
**Grupo:** 4  
**Fecha:** 23/10/24

**1. Descripción General del Sistema**

Proporciona una descripción general del sistema y los principales componentes lógicos que lo forman.

**Descripción del sistema:**

El sistema es una plataforma de noticias que integra tecnologías de procesamiento de información y generación de contenido, permitiendo a los usuarios acceder, explorar y personalizar la información actual de diversas temáticas. Se basa en una arquitectura moderna utilizando Next.js y React, con Supabase como base de datos y OpenAI para la generación de contenido y Tavily para hacer web Scraping.

**Principales Componentes Lógicos**

1. **Interfaz de Usuario (UI)**
   * **Diseño responsivo que permite una experiencia de navegación fluida tanto en dispositivos móviles como en escritorio.**
   * **Componentes interactivos como tarjetas para mostrar artículos, menús de navegación y secciones de comunidad.**
2. **Módulo de Registro y Autenticación**
   * **Funcionalidades de login y registro de usuarios, asegurando que solo los usuarios autenticados accedan a contenido personalizado.**
   * **Gestión de perfiles de usuario, donde se almacenan preferencias y contenido generado.**
3. **Panel de Inteligencia Artificial**
   * **Generación de contenido utilizando el modelo de OpenAI.**
   * **Interfaz para previsualizar y almacenar contenido generado por el usuario.**
4. **Sistema de Recomendaciones**
   * **Algoritmos que sugieren usuarios y publicaciones basadas en intereses y actividad previa del usuario.**
   * **Sección de comunidad donde los usuarios pueden interactuar con contenido de otros usuarios recomendados.**
5. **Análisis y Clasificación de Contenidos**
   * **Módulo que clasifica publicaciones en diversas categorías (Política, Economía, Salud, etc.) usando OpenAI para mejorar la organización de la información.**
   * **Análisis de popularidad y comentarios para optimizar el contenido mostrado.**
6. **Base de Datos (Supabase)**
   * **Almacenamiento seguro de información del usuario, historial de consultas y contenido generado.**
   * **Gestión de la autenticación de usuarios y sus preferencias.**
7. **Módulo de Accesibilidad**
   * **Herramientas y funcionalidades diseñadas para asegurar que la plataforma sea accesible para jóvenes, profesionales y personas de la tercera edad.**

**2. Principales Clases y Entidades del Sistema**

Describe las clases o entidades principales del sistema, sus atributos y métodos principales, y su propósito dentro del sistema.

**Clase 1: Usuario**

* **Atributos:**
  + **id: Identificador único del usuario.**
  + **nombre: Nombre del usuario.**
  + **Fecha: Fecha de Nacimiento**
  + **Estudios: Nivel estudio del Usuario**
  + **Profesión: Carrera (Técnica/Superior)**
  + **email: Correo electrónico del usuario.**
  + **contraseña: Contraseña (almacenada de forma segura).**
  + **preferencias: Lista de intereses y temas preferidos.**
  + **historial: Registro de artículos y contenido generado.**
* **Métodos:**
  + **registrar(): Permite al usuario crear una nueva cuenta.**
  + **iniciarSesion(): Autentica al usuario.**
  + **guardarPreferencias(): Almacena las preferencias del usuario.**
  + **obtenerHistorial(): Recupera el historial de actividad del usuario.**
* **Propósito:**
  + **Gestionar la información del usuario, autenticarlo y permitir la personalización de su experiencia en la plataforma.**

**Clase 2: Artículo**

* **Atributos:**
  + **id: Identificador único del artículo.**
  + **titulo: Título del artículo.**
  + **contenido: Texto completo del artículo.**
  + **categoria: Clasificación del artículo (Política, Economía, etc.).**
  + **fechaPublicacion: Fecha de publicación del artículo.**
  + **autor: Usuario que creó el artículo.**
* **Métodos:**
  + **obtenerDetalles(): Recupera información detallada del artículo.**
  + **TraerNoticias: Extraer Datos de News API**
* **Propósito:**
  + **Representar los artículos y gestionar su creación, edición y eliminación, así como su visualización.**

**Clase 3: Comentario**

* **Atributos:**
  + **id: Identificador único del comentario.**
  + **contenido: Texto del comentario.**
  + **usuarioId: Identificador del usuario que hizo el comentario.**
  + **articuloId: Identificador del artículo al que pertenece el comentario.**
  + **fecha: Fecha y hora del comentario.**
* **Métodos:**
  + **crear(): Permite a un usuario agregar un nuevo comentario a un artículo.**
  + **eliminar(): Elimina un comentario específico.**
  + **obtenerComentariosPorArticulo(): Recupera todos los comentarios de un artículo.**
* **Propósito:**
  + **Facilitar la interacción entre los usuarios a través de comentarios en los artículos, permitiendo la discusión y retroalimentación.**

**Clase 4: Recomendacion**

* **Atributos:**
  + **id: Identificador único de la recomendación.**
  + **usuarioId: Identificador del usuario que recibe la recomendación.**
  + **articuloId: Identificador del artículo recomendado.**
  + **motivo: Razón de la recomendación (basada en intereses, popularidad, etc.).**
* **Métodos:**
  + **generarRecomendaciones(): Crea una lista de recomendaciones para un usuario basado en sus preferencias.**
  + **obtenerRecomendaciones(): Devuelve las recomendaciones actuales de un usuario.**
* **Propósito:**
  + **Sugerir contenido relevante a los usuarios según sus intereses y actividad previa, mejorando la personalización de la experiencia.**

**Clase 5: IA\_Generador**

* **Atributos:**
  + **modelo: Modelo de IA utilizado (e.g., OpenAI).**
  + **historialConsultas: Registro de consultas realizadas por el usuario.**
* **Métodos:**
  + **generarContenido(): Crea contenido basado en las especificaciones del usuario.**
  + **previsualizarContenido(): Muestra un avance del contenido generado.**
  + **almacenarContenido(): Guarda el contenido generado en el perfil del usuario.**
* **Propósito:**
  + **Facilitar la generación de contenido mediante inteligencia artificial, permitiendo a los usuarios crear y almacenar información personalizada.**

**3. Relaciones entre clases**

Especifica cómo están relacionadas las clases entre sí, indicando relaciones de herencia, composición, agregación o dependencias.

**Relaciones entre Clases**

1. **Usuario ↔ Artículo**
   * **Relación: Dependencia**
   * **Descripción: Un Usuario puede ver múltiples Artículos. Cada Artículo tiene su propio título, cuerpo, imagen y link/url. Esto implica que, aunque los artículos dependen de los usuarios para su creación, pueden existir sin que el usuario esté activo en la plataforma.**
2. **Artículo ↔ Comentario**
   * **Relación: Composición**
   * **Descripción: Un Artículo puede tener múltiples Comentarios asociados. Si un Artículo se elimina, también se eliminarán todos sus Comentarios relacionados, ya que estos no tendrían sentido sin el artículo correspondiente.**
3. **Usuario ↔ Comentario**
   * **Relación: Agregación**
   * **Descripción: Un Usuario puede realizar múltiples Comentarios. Cada Comentario tiene un atributo usuarioId que referencia al Usuario que lo creó. Los comentarios pueden existir independientemente de que el usuario esté activo, pero dependen de la existencia del usuario para su identificación.**
4. **Usuario ↔ Recomendacion**
   * **Relación: Agregación**
   * **Descripción: Un Usuario puede recibir múltiples Recomendaciones. Las recomendaciones se generan en base a las preferencias del usuario y su historial de actividad. Aunque las recomendaciones son específicas para cada usuario, pueden ser creadas y eliminadas sin afectar la existencia del usuario.**
5. **IA\_Generador ↔ Artículo**
   * **Relación: Dependencia**
   * **Descripción: La clase IA\_Generador tiene una dependencia con la clase Artículo para generar contenido relacionado. El generador utiliza los artículos existentes como referencia para crear nuevos contenidos. Sin embargo, IA\_Generador puede funcionar sin necesidad de que existan artículos previamente.**
6. **Artículo ↔ Categoría**
   * **Relación: Dependencia**
   * **Descripción: Un Artículo tiene un atributo categoria, que clasifica el artículo en diferentes temáticas (Política, Economía, etc.). Esto significa que la existencia de un artículo depende de que se le asigne una categoría.**

**Resumen de Relaciones**

* **Agregación: Usuario ↔ Artículo, Usuario ↔ Comentario, Usuario ↔ Recomendacion**
* **Composición: Artículo ↔ Comentario**
* **Dependencia: IA\_Generador ↔ Artículo, Artículo ↔ Categoría**

**4. Interfaces y dependencias**

Indica si alguna de las clases o entidades tiene interfaces y cómo interactúan con otros módulos del sistema a través de ellas.

**Descripción de Interfaces y Dependencias**

1. **Usuario**
   * **Interfaz**: IUsuario
   * **Descripción**: Define métodos para registrar, iniciar sesión, y gestionar preferencias del usuario.
   * **Interacción**: Esta interfaz permite que otras partes del sistema (como el módulo de autenticación) interactúen con la clase Usuario de manera estandarizada, facilitando la implementación de diferentes métodos de autenticación (por ejemplo, autenticación con redes sociales).
2. **Artículo**
   * **Interfaz**: IArticulo
   * **Descripción**: Especifica métodos para crear, editar, eliminar y obtener detalles de artículos.
   * **Interacción**: Permite que el módulo de gestión de contenido y el panel de usuario se comuniquen con la clase Artículo sin necesidad de conocer su implementación interna. Esto es útil para los módulos que gestionan la visualización de artículos o la generación de informes.
3. **Comentario**
   * **Interfaz**: IComentario
   * **Descripción**: Define métodos para agregar, eliminar y recuperar comentarios.
   * **Interacción**: Facilita la integración entre la clase Comentario y otros módulos, como el sistema de notificaciones, que puede notificar a los usuarios sobre nuevos comentarios en sus artículos.
4. **IA\_Generador**
   * **Interfaz**: IIAGenerador
   * **Descripción**: Especifica métodos para generar contenido, previsualizarlo y almacenarlo.
   * **Interacción**: Esta interfaz permite a otros módulos (como el panel de IA) solicitar generación de contenido sin preocuparse por los detalles de implementación del generador de inteligencia artificial.
5. **Recomendacion**
   * **Interfaz**: IRecomendacion
   * **Descripción**: Define métodos para generar y obtener recomendaciones.
   * **Interacción**: Permite que el módulo de recomendaciones interactúe con la clase Recomendacion para presentar sugerencias personalizadas a los usuarios.

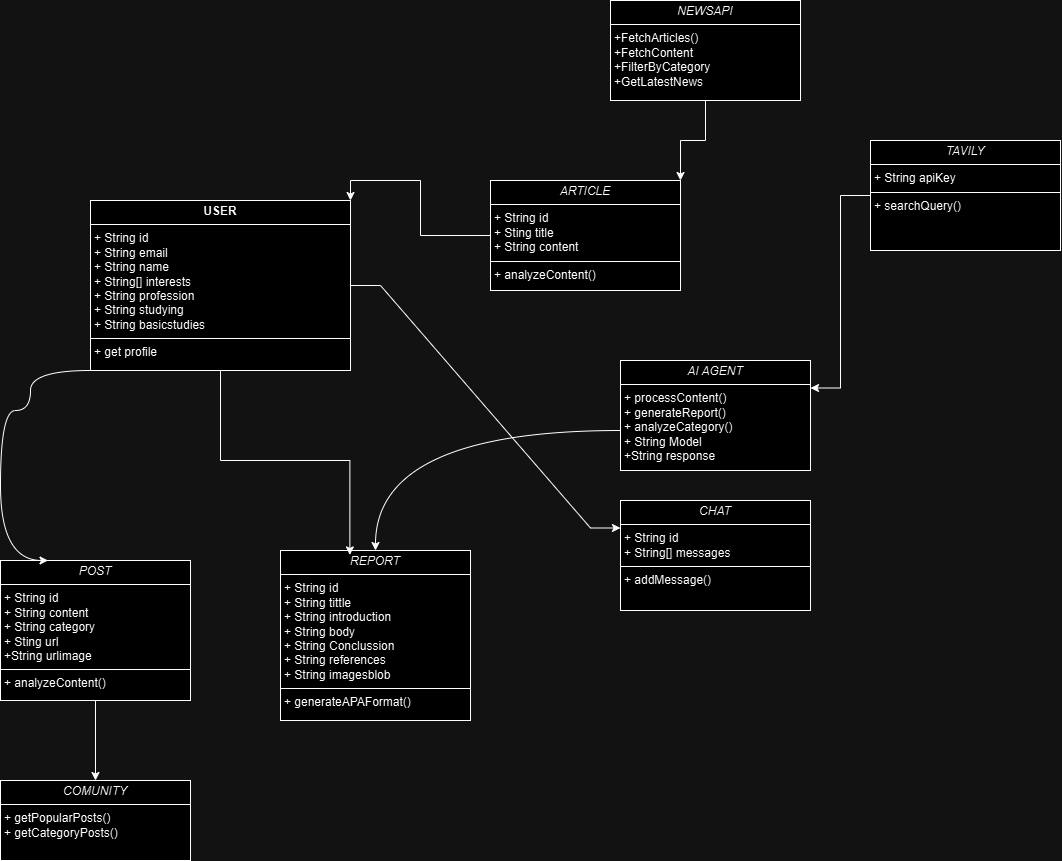
**Resumen de Interacciones**

* **Módulos del sistema**: Las interfaces permiten una comunicación clara entre los módulos, como la gestión de contenido, autenticación, y sistema de recomendaciones.
* **Flexibilidad y escalabilidad**: Al utilizar interfaces, el sistema puede adaptarse más fácilmente a cambios futuros, como la implementación de nuevas fuentes de datos o métodos de autenticación.

**5. Diagrama de Clases del Sistema**

Adjunta un diagrama de clases que representa la estructura lógica del sistema, las clases principales y las relaciones entre ellas.

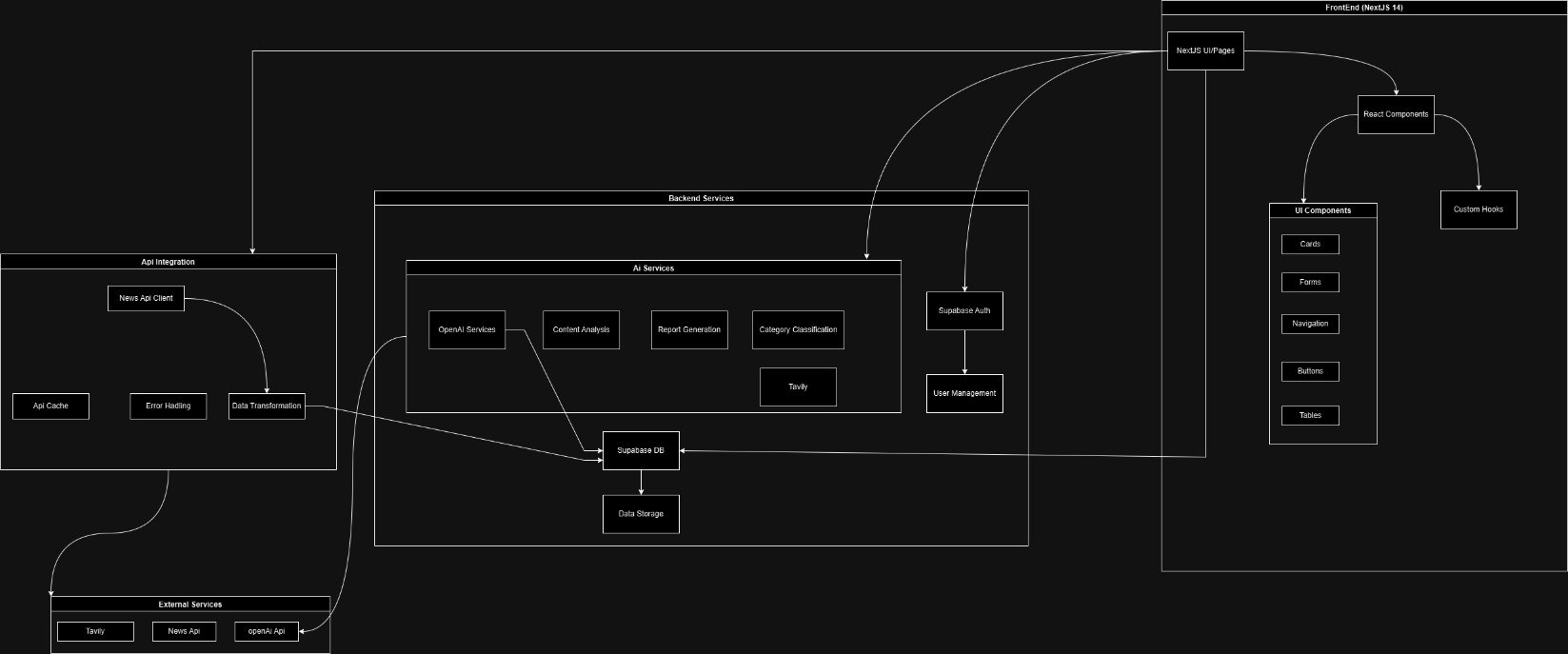
**Diagrama de clases:***(Adjunta el diagrama en esta sección)*

**

**6. Diagrama de componentes (opcional)**

Si su sistema lo requiere, adjunte un diagrama de componentes que represente cómo los componentes principales del sistema están organizados y cómo interactúan.

**Diagrama de componentes:**

****

**7. Comentarios adicionales**

Incluye cualquier comentario adicional o aclaración relevante sobre la vista lógica de tu sistema.

**Comentarios:**

** Escalabilidad: El diseño del sistema permite la adición de nuevos módulos o componentes sin necesidad de reestructurar completamente la arquitectura existente. Esto facilita la evolución del sistema a medida que cambian las necesidades de los usuarios.**

** Seguridad: Se deben implementar medidas de seguridad robustas, especialmente en los módulos de autenticación y manejo de datos sensibles. Esto incluye el uso de técnicas como el hashing de contraseñas y la implementación de OAuth para el acceso a través de redes sociales.**

** Accesibilidad: Dado que uno de los objetivos es mejorar la accesibilidad para diferentes grupos de usuarios, se deben considerar prácticas de diseño inclusivas en la interfaz de usuario. Esto incluye un diseño responsivo, opciones de contraste alto, y compatibilidad con lectores de pantalla.**

** Uso de IA: La integración de IA en el sistema no solo permite la generación de contenido, sino que también puede ser utilizada para mejorar el sistema de recomendaciones. La IA puede aprender de las interacciones de los usuarios para ofrecer sugerencias más personalizadas y relevantes.**

** Optimización de Rendimiento: Dada la naturaleza del sistema, que involucra la recuperación de datos de API externas y la generación de contenido, es crucial implementar estrategias de caché y optimización de consultas a la base de datos para mejorar la experiencia del usuario y reducir tiempos de carga.**

** Pruebas y Mantenimiento: Es fundamental establecer un plan de pruebas exhaustivo que incluya pruebas unitarias, de integración y de usuario. Esto garantizará que el sistema funcione como se espera y facilitará el mantenimiento en el futuro.**

** Documentación: La creación de documentación clara y detallada sobre la arquitectura del sistema, el uso de interfaces y las interacciones entre componentes ayudará a futuros desarrolladores a comprender y contribuir al proyecto.**