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

**Proyecto:** NewsEye  
**Grupo:** 4  
**Fecha:** 4/10/2024

**1. Estructura del sistema**

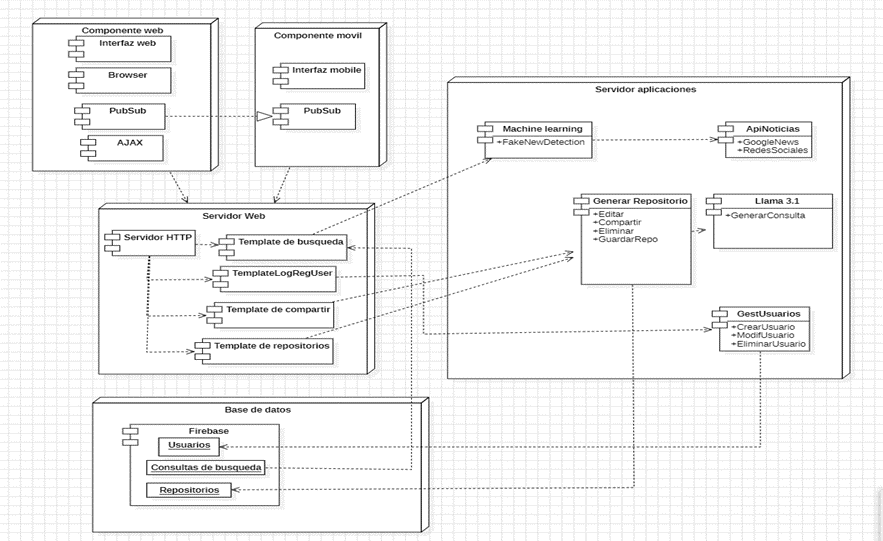
Describe la estructura del sistema de tu proyecto, mencionando los módulos principales y cómo interactúan entre sí. Puedes usar diagramas de arquitectura para complementar tu descripción.

**Descripción de la estructura:**

Nuestro sistema cuenta con una estructura basada en el lenguaje de programación Python, sujeto a cambios de ambiente visual como React para animación y CSS, implementando una integración con una App Móvil, utilizada netamente de React. Contamos con una base de datos local la cual se encuentra directamente en el entorno virtual destinado a la aplicación, en este caso, MySQL actuará como un intermediario de datos de usuario, creación de usuarios y almacenamiento de archivos.

El sistema actual busca procesar la información que obtenga desde la web (ya sea como API o Web Scrapping) de noticias de la actualidad, el usuario podrá navegar en nuestro sitio o aplicación y podrá buscar libremente e integrar procesos adicionales informativos, tanto como “Transformar la noticia buscada en un reporte de investigación”, “Mejorar la búsqueda de información de la API (Home Page) con las preferencias y gustos de los usuarios” y finalmente mejorar la accesibilidad de usuarios tanto jóvenes, profesionales y tercera edad.

**Diagrama de la estructura del sistema:**   
*(Adjunta un diagrama UML o un diagrama de bloques que muestre los módulos principales y sus interacciones.)*



**2. Módulos principales**

Enumera y describe los módulos principales del sistema, explicando sus responsabilidades y cómo interactúan con otros módulos.

**Módulo 1 (Nombre): Módulo de chat IA**  
**Responsabilidad:** Chat de interacción con el usuario para que este pueda buscar información redactada.  
**Interacción con otros módulos:** *Módulo 3, Módulo 2, Módulo 5*

**Módulo 2 (Nombre): Integración de api para asistencia de IA**  
**Responsabilidad:** Integración de api para mejorar la asistencia de IA y dar respuesta en base a los resultados de APIS de información (noticias, documentos informativos).  
**Interacción con otros módulos:** *Módulo 1, Módulo 5.*

**Módulo 3 (Nombre): Integración de IA para generar documentos**  
**Responsabilidad:** Integración de IA para generar documentos a partir de un prompt realizado   
**Interacción con otros módulos:** *Módulo 1, Módulo 2, Módulo 5*

**Módulo 4 (Nombre): Base de datos**  
**Responsabilidad:** Base de datos de usuario para implementar inicios de sesión y almacenamiento de prompts**.**  
**Interacción con otros módulos:** *Módulo 3, Módulo 7*

**Módulo 5 (Nombre): Modelo Web Scrapping**  
**Responsabilidad:** Utilización de web scrapping para recolectar información de documentos, noticias o sitios informativos para generar una respuesta personalizada.  
**Interacción con otros módulos:** *Módulo 1, Módulo 2*

**Módulo 6 (Nombre): Despliegue de información**  
**Responsabilidad:** Utilizar los procesos de información para moldear un documento óptimo que ayude a la divulgación de contenido en distintas redes.  
**Interacción con otros módulos:** *Módulo 1, Módulo 4, Módulo 7*

**Módulo 7 (Nombre): Registro de usuario**  
**Responsabilidad: Permitir la capacidad de registro y logueo de usuario para que este pueda acceder a sus prompt y generación de documentos.**

**Interacción con otros módulos:** *Módulo 3, Módulo 4, Módulo 6*

*(Agrega más módulos si es necesario)*

**3. Dependencias y Librerías**

Especifica las dependencias y librerías/frameworks que estás utilizando en tu proyecto, indicando para qué funcionalidad o módulo se usan.

**Dependencias y Librerías:**

Las dependencias y librerías de nuestros módulos actualmente se ha hecho bastante largo, pero este es más o menos lo que necesitaremos para poder llevar a cabo la implementación de la aplicación en el entorno virtual, lo que se verá ahora se ha traido de un archivo “requirements.txt”.

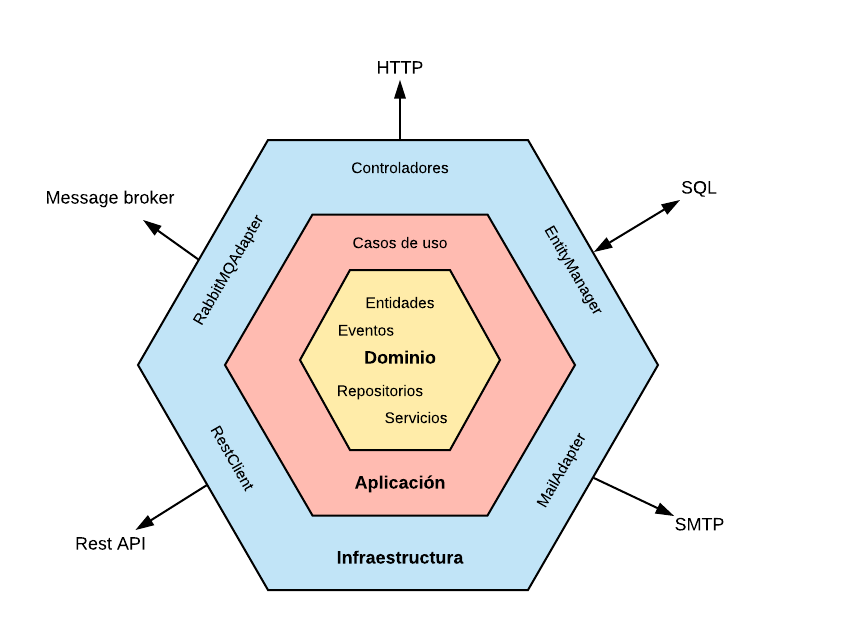
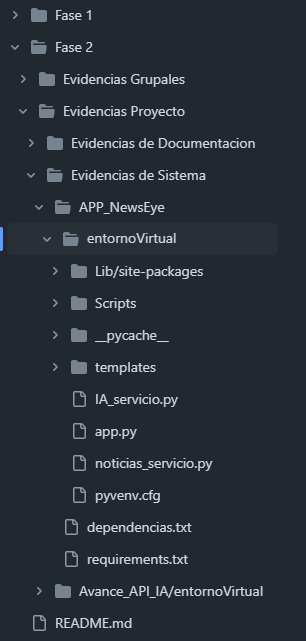
* Flask
* Flask-CORS
* requests
* python-dotenv
* flask-login
* flask-bcrypt
* flask-wtf
* WTForms
* email-validator
* Werkzeug==2.2.2
* mysql-connector-python
* ollama
* python-ollama
* langchain
* langchain\_community
* sentence-transformers
* pypdf
* chromadb

**4. Organización del Código**

Para nuestro código estamos usando una estructura basada en Arquitectura Hexagonal, el cual es esencial en Proyectos medianos/grandes, alineado y que contengan +100 archivos en donde se separan los archivos basados en su funcionalidad

**Organización del código:**

**Arquitectura Hexagonal**



**5. Reglas de Integración**

Especifica las reglas que tu equipo sigue para la integración del código entre los módulos. Esto puede incluir restricciones sobre el acceso a datos, convenciones de desarrollo, etc.

**Reglas de integración:**

* Mantenemos la restricción de lenguajes alternativos al ambiente de desarrollo actual, por ejemplo querer integrar “Raw” a nuestra App “Flask”
* Restricciones de ingreso a las claves de acceso tanto de información de APIs, como la llave de acceso secreta del proyecto
* Quitamos por seguridad las API Keys antes de subir el proyecto a GitHub

**6. Prácticas de desarrollo**

Detalla las prácticas de desarrollo que estás utilizando en el proyecto, como las pruebas unitarias, integración continua, revisión de código, etc.

**Prácticas de desarrollo:**

En nuestras prácticas del desarrollo estamos utilizando la integración continua, ya que siempre al crearse alguna nueva función se integra inmediatamente, de igual manera utilizamos revisión de código utilizando las diversas sugerencias que a nivel grupal surgen así mejorando el código, también la documentación ya que así hacemos que el entendimiento del desarrollo y funcionamiento del software al personal del proyecto y futuras personas que trabajen en él(nadie más solo nosotros), por último pero no menos importante el comentar tu código, son buenas prácticas y ayuda a no desordenar los trozos de código

**7. Herramientas de desarrollo**

Mencione las herramientas que está utilizando para el desarrollo del proyecto (por ejemplo, IDE, sistemas de control de versiones, etc.).

**Herramientas:**

IDE: Visual Studio Code

Sistema control de Versiones: GITHUB

Pruebas de Sistemas: Selenium IDE

Control de código fuente: SonarQube

Investigación y Debugging: Stack Overflow

Documentación: Documentos de Google

**8. Diagrama de Arquitectura del Sistema**

Adjunta un diagrama de arquitectura general del sistema que incluye los módulos, las interacciones entre ellos y las tecnologías involucradas.

**Diagrama de arquitectura general:**

