****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto CommentSentiment**

Curso: *INTELIGENCIA ARTIFICIAL*

Docente: Dr. Oscar Juan Jimenez Flores

Integrantes:

***Erick Javier Salinas Condori (2020069046)***

***Aaron Pedro Paco Ramos (2018000654)***

***Justin Zinedine Zevallos Purca (2020066924)***

***Mayner Gonzalo Anahua Coaquira (2020067145)***

**Tacna – Perú**

***2024***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema CommentSentiment

Documento de Especificación de Requerimientos de Software

Versión *{1.0}*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

INDICE GENERAL

Contenido

[1. INTRODUCCIÓN 5](#_Toc69808834)

[1.1. Propósito (Diagrama 4+1) 5](#_Toc69808835)

[1.2. Alcance 5](#_Toc69808836)

[1.3. Definición, siglas y abreviaturas 5](#_Toc69808837)

[1.4. Organización del documento 5](#_Toc69808838)

[2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5](#_Toc69808839)

[2.1.1. Requerimientos Funcionales 5](#_Toc69808840)

[2.1.2. Requerimientos No Funcionales – Atributos de Calidad 5](#_Toc69808841)

[3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6](#_Toc69808842)

[3.1. Vista de Caso de uso 6](#_Toc69808843)

[3.1.1. Diagramas de Casos de uso 6](#_Toc69808844)

[3.2. Vista Lógica 6](#_Toc69808845)

[3.2.1. Diagrama de Subsistemas (paquetes) 7](#_Toc69808846)

[3.2.2. Diagrama de Secuencia (vista de diseño) 7](#_Toc69808847)

[3.2.3. Diagrama de Colaboración (vista de diseño) 7](#_Toc69808848)

[3.2.4. Diagrama de Objetos 7](#_Toc69808849)

[3.2.5. Diagrama de Clases 7](#_Toc69808850)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 7](#_Toc69808851)

[3.3. Vista de Implementación (vista de desarrollo) 7](#_Toc69808852)

[3.3.1. Diagrama de arquitectura software (paquetes) 7](#_Toc69808853)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 7](#_Toc69808854)

[3.4. Vista de procesos 7](#_Toc69808855)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 8](#_Toc69808856)

[3.5. Vista de Despliegue (vista física) 8](#_Toc69808857)

[3.5.1. Diagrama de despliegue 8](#_Toc69808858)

[4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 8](#_Toc69808859)

[Escenario de Funcionalidad 8](#_Toc69808860)

[Escenario de Usabilidad 8](#_Toc69808861)

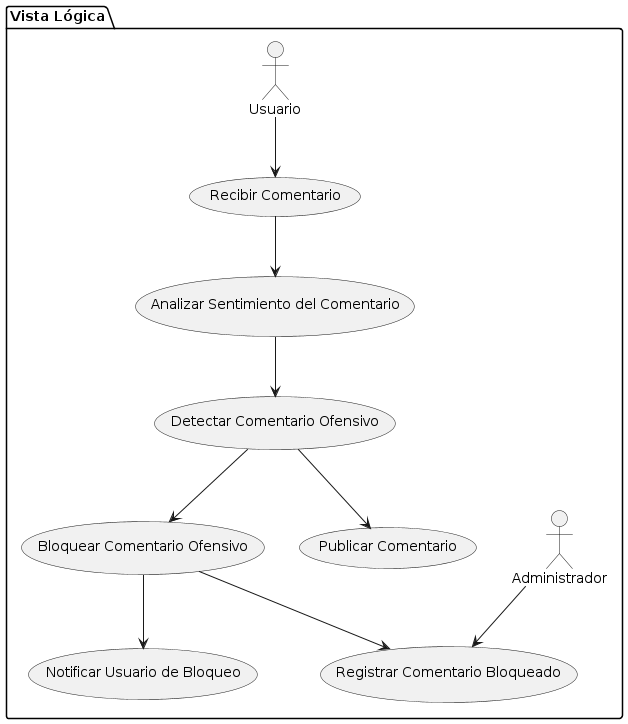
[Escenario de confiabilidad 9](#_Toc69808862)

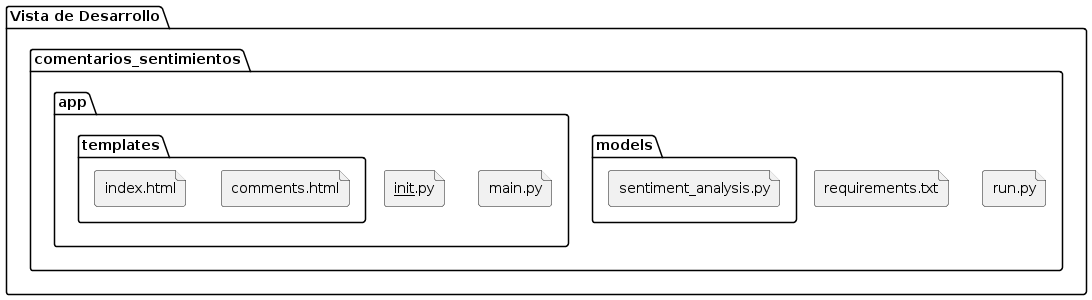
[Escenario de rendimiento 9](#_Toc69808863)

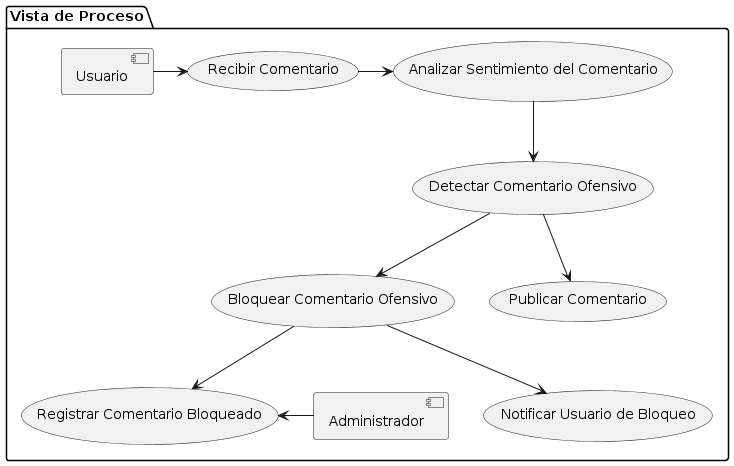
[Escenario de mantenibilidad 9](#_Toc69808864)

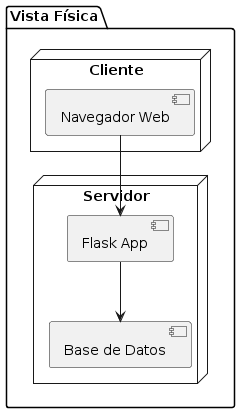
[Otros Escenarios 9](#_Toc69808865)

1. INTRODUCCIÓN
   1. Propósito (Diagrama 4+1)









* 1. Alcance

El proyecto "CommentSentiment" se enfoca en mejorar la eficiencia en el procesamiento de comentarios en plataformas web mediante la implementación de tecnologías avanzadas de inteligencia artificial. El plugin facilitará la comprensión rápida y precisa del contenido de los comentarios, mejorará la toma de decisiones y optimizará los procesos de interacción en línea en diversos sectores y plataformas digitales.

* 1. Definición, siglas y abreviaturas

 **UC (Use Case)**:

* **Definición**: Caso de uso. Es una descripción de cómo los usuarios interactúan con el sistema para lograr un objetivo específico.

 **UML (Unified Modeling Language)**:

* **Definición**: Lenguaje de modelado unificado. Es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos de software.

 **MVC (Model-View-Controller)**:

* **Definición**: Modelo-Vista-Controlador. Es un patrón de diseño que separa la aplicación en tres componentes principales: el modelo (datos), la vista (interfaz de usuario) y el controlador (lógica de negocio).

 **CRUD (Create, Read, Update, Delete)**:

* **Definición**: Crear, Leer, Actualizar, Eliminar. Son las operaciones básicas que se pueden realizar en una base de datos o en una interfaz de usuario.

 **HTML (HyperText Markup Language)**:

* **Definición**: Lenguaje de Marcado de Hipertexto. Es el lenguaje estándar para crear y estructurar páginas web.

 **CSS (Cascading Style Sheets)**:

* **Definición**: Hojas de Estilo en Cascada. Es un lenguaje usado para describir la presentación de un documento escrito en HTML o XML.

 **JS (JavaScript)**:

* **Definición**: Lenguaje de programación que permite crear contenido dinámico en páginas web.

 **SQL (Structured Query Language)**:

* **Definición**: Lenguaje de Consulta Estructurada. Es el lenguaje estándar para gestionar y manipular bases de datos relacionales.

 **API (Application Programming Interface)**:

* **Definición**: Interfaz de Programación de Aplicaciones. Es un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes aplicaciones de software.

 **REST (Representational State Transfer)**:

* **Definición**: Transferencia de Estado Representacional. Es un estilo de arquitectura para diseñar servicios web.

 **JSON (JavaScript Object Notation)**:

* **Definición**: Notación de Objetos de JavaScript. Es un formato ligero de intercambio de datos.

 **CI/CD (Continuous Integration/Continuous Deployment)**:

* **Definición**: Integración Continua/Despliegue Continuo. Es una práctica de desarrollo de software donde los desarrolladores integran código en un repositorio compartido de manera frecuente y el despliegue a producción se automatiza.

 **ML (Machine Learning)**:

* **Definición**: Aprendizaje Automático. Es una rama de la inteligencia artificial que se centra en desarrollar sistemas que aprenden de los datos y mejoran su rendimiento con el tiempo sin ser programados explícitamente.

 **NLP (Natural Language Processing)**:

* **Definición**: Procesamiento del Lenguaje Natural. Es una rama de la inteligencia artificial que se centra en la interacción entre computadoras y el lenguaje humano.

 **AI (Artificial Intelligence)**:

* **Definición**: Inteligencia Artificial. Es la simulación de procesos de inteligencia humana por parte de máquinas, especialmente sistemas informáticos.

 **PaaS (Platform as a Service)**:

* **Definición**: Plataforma como Servicio. Es un modelo de servicio en la nube que proporciona una plataforma y un entorno para permitir a los desarrolladores construir aplicaciones y servicios a través de Internet.

 **IaaS (Infrastructure as a Service)**:

* **Definición**: Infraestructura como Servicio. Es una forma de servicio en la nube que proporciona recursos de computación virtualizados a través de Internet.

 **SaaS (Software as a Service)**:

* **Definición**: Software como Servicio. Es un modelo de distribución de software en el que las aplicaciones están alojadas por un proveedor de servicios y están disponibles para los clientes a través de una red, generalmente Internet.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

[Establezca las prioridades de los requerimientos y las restricciones del proyecto)

* 1. Priorización de requerimientos

### Requerimientos Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **ID del Requerimiento** | **Descripción** | **Prioridad** | **Criterios de Aceptación** |
| RF01 | Recepción de comentarios | Alta | El sistema debe recibir y almacenar todos los comentarios enviados por los usuarios |
| RF02 | Análisis de Sentimientos | Alta | La IA debe analizar cada comentario para determinar su tono emocional (positivo, negativo, neutro). |
| RF03 | Detección de Lenguaje Ofensivo | Alta | La IA debe identificar y clasificar comentarios que contengan lenguaje ofensivo o de odio. |
| RF04 | Bloqueo Automático de Comentarios Ofensivos | Alta | El sistema debe bloquear automáticamente los comentarios clasificados como ofensivos y notificar al usuario sobre el bloqueo. |
| RF05 | Publicación de Comentarios Positivos y Neutros | Alta | Los comentarios clasificados como positivos o neutros deben ser publicados automáticamente en la plataforma |
| RF06 | Publicación de Comentarios de Insatisfacción No Ofensivos | Media | Los comentarios clasificados como insatisfactorios pero no ofensivos deben ser publicados automáticamente. |
| RF07 | Generación de Informes | Alta | El sistema debe generar informes detallados de todos los comentarios analizados, tanto bloqueados como publicados, para revisión del administrador. |
| RF08 | Interfaz de Gestión para Administradores | Alta | Proveer una interfaz intuitiva para que los administradores configuren los parámetros de análisis y moderación, revisen comentarios bloqueados y generen reportes |
| RF09 | Notificaciones al Usuario | Media | El sistema debe notificar a los usuarios cuando sus comentarios sean bloqueados por contener lenguaje ofensivo |
| RF10 | Escalabilidad del Sistema | Media | El sistema debe ser capaz de manejar un gran volumen de comentarios sin comprometer el rendimiento |
| RF11 | Seguridad y Privacidad | Alta | El sistema debe garantizar la seguridad y privacidad de los datos de los usuarios, cumpliendo con las normativas de protección de datos |
| RF12 | Integración con Sistemas Existentes | Alta | El sistema debe ser fácilmente integrable con las infraestructuras tecnológicas existentes en la plataforma web |
| RF13 | Mantenimiento y Actualizaciones | Media | El sistema debe permitir actualizaciones y mantenimiento regular sin interrumpir el servicio. |

### Requerimientos No Funcionales – Atributos de Calidad

|  |  |  |  |
| --- | --- | --- | --- |
| **ID del Requerimiento** | **Descripción** | **Prioridad** | **Criterios de Aceptación** |
| RNF01 | Rendimiento y Escalabilidad | Alta | El sistema debe manejar al menos 10,000 comentarios por hora sin degradar el rendimiento. |
| RNF02 | Disponibilidad | Alta | El sistema debe estar disponible al menos el 99.9% del tiempo, garantizando un tiempo de inactividad no superior a 8.76 horas al año. |
| RNF03 | Seguridad | Alta | El sistema debe implementar autenticación segura y encriptación de datos, cumpliendo con las normativas de protección de datos (por ejemplo, GDPR) |
| RNF04 | Privacidad | Alta | Los datos personales de los usuarios deben ser tratados conforme a las políticas de privacidad y no deben ser accesibles a personal no autorizado |
| RNF05 | Compatibilidad | Media | El sistema debe ser compatible con los navegadores web modernos (Chrome, Firefox, Safari, Edge). |
| RNF06 | Mantenibilidad | Media | El código del sistema debe seguir estándares de desarrollo que faciliten su mantenimiento y actualización, con documentación clara y completa. |
| RNF07 | Usabilidad | Alta | La interfaz de usuario debe ser intuitiva y fácil de usar, con una curva de aprendizaje mínima para los administradores y usuarios finales. |
| RNF08 | Portabilidad | Media | El sistema debe ser portable entre diferentes plataformas de servidores en la nube, como AWS, Google Cloud y Azure. |
| RNF09 | Fiabilidad | Alta | El sistema debe operar sin fallos críticos durante el 99.9% del tiempo operativo. |
| RNF10 | Eficiencia | Alta | El tiempo de respuesta para el análisis de cada comentario no debe exceder los 2 segundos. |
| RNF11 | Documentación | Media | Toda la documentación técnica y de usuario debe estar actualizada y ser accesible, incluyendo manuales de usuario, guías de instalación y mantenimiento. |
| RNF12 | Interoperabilidad | Alta | El sistema debe ser capaz de integrarse con sistemas de terceros mediante APIs estándar |
| RNF13 | Adaptabilidad | Media | El sistema debe permitir la configuración y adaptación a diferentes idiomas y contextos culturales. |

* 1. Restricciones

 **Restricciones de Tiempo**:

* **Fecha de Entrega**: El proyecto debe ser completado y entregado en una fecha específica, lo que limita el tiempo disponible para el desarrollo y pruebas.
* **Plazos Intermedios**: Deben cumplirse fechas límite intermedias para hitos específicos del proyecto, como la finalización de ciertas funcionalidades.

 **Restricciones de Presupuesto**:

* **Costos de Desarrollo**: El presupuesto para el desarrollo del proyecto está limitado, lo que puede afectar la contratación de personal, la adquisición de software y hardware, y otros gastos operativos.
* **Licencias de Software**: Costo de licencias de herramientas y plataformas necesarias para el desarrollo.

 **Restricciones Técnicas**:

* **Compatibilidad de Hardware**: El sistema debe ser compatible con el hardware existente del cliente.
* **Compatibilidad de Software**: El sistema debe ser compatible con el software y sistemas operativos existentes.
* **Tecnologías Utilizadas**: El proyecto debe utilizar tecnologías específicas (e.g., Flask, MySQL, PlantUML).
* **Limitaciones de Rendimiento**: El sistema debe cumplir con ciertos requisitos de rendimiento, como tiempos de respuesta y capacidad de manejo de carga.

 **Restricciones de Seguridad**:

* **Protección de Datos**: El sistema debe cumplir con las normativas de protección de datos personales (e.g., GDPR, CCPA).
* **Autenticación y Autorización**: Debe implementarse un sistema seguro de autenticación y autorización para acceder a los datos y funcionalidades.
* **Encriptación**: Los datos sensibles deben ser encriptados tanto en tránsito como en reposo.

 **Restricciones de Calidad**:

* **Estándares de Calidad**: El sistema debe cumplir con ciertos estándares de calidad, como ISO/IEC 25010, que define características de calidad para sistemas de software.
* **Pruebas y Validaciones**: Deben realizarse pruebas exhaustivas para asegurar que el sistema funciona correctamente y cumple con los requisitos.

 **Restricciones Legales y Regulatorias**:

* **Cumplimiento Normativo**: El sistema debe cumplir con todas las leyes y regulaciones aplicables.
* **Licencias de Uso**: Cumplir con las licencias de uso de las herramientas y tecnologías utilizadas en el proyecto.

 **Restricciones de Recursos Humanos**:

* **Disponibilidad del Equipo**: La disponibilidad de desarrolladores, diseñadores, testers y otros miembros del equipo puede estar limitada.
* **Habilidades del Equipo**: El equipo debe tener las habilidades necesarias para utilizar las tecnologías y herramientas requeridas.

 **Restricciones de Infraestructura**:

* **Capacidad del Servidor**: Limitaciones en la capacidad y escalabilidad del servidor donde se alojará el sistema.
* **Conectividad a Internet**: Dependencia de una conexión a Internet estable y de alta velocidad.

 **Restricciones de Usuario**:

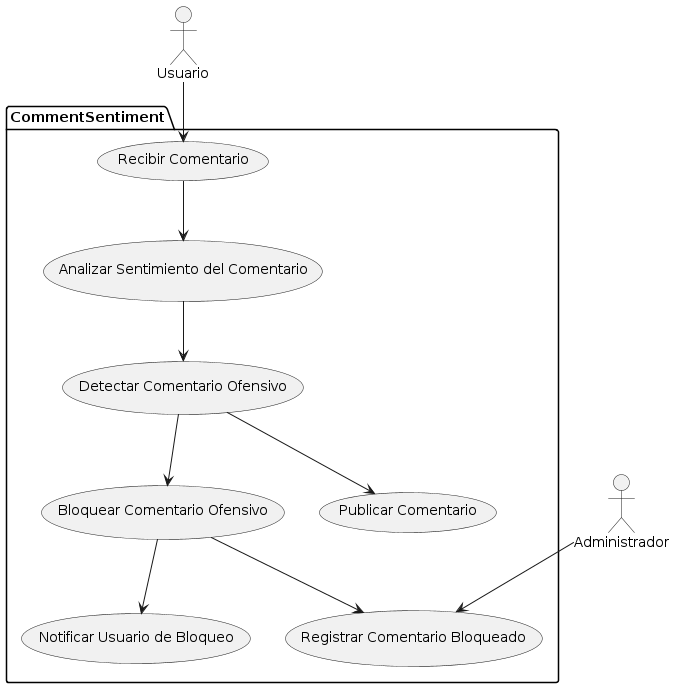
* **Interfaz de Usuario**: La interfaz de usuario debe ser intuitiva y fácil de usar, cumpliendo con ciertos estándares de usabilidad.
* **Soporte Multilingüe**: El sistema debe soportar múltiples idiomas según las necesidades del cliente.

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

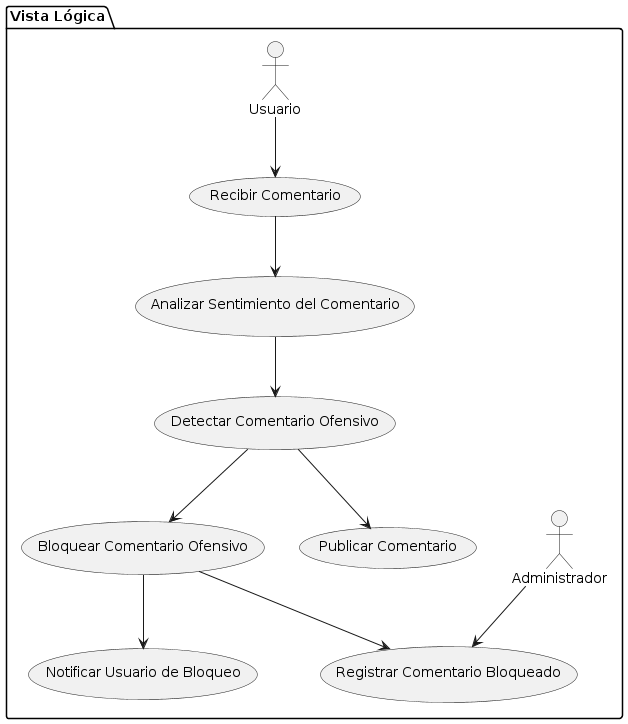
* 1. Vista de Caso de uso

|  |  |
| --- | --- |
| **Caso de Uso** | Modera Comentarios |
| **Actores** | Usuario (iniciador), Administrador |
| **Tipo** | Principal |
| **Descripción** | El sistema de IA analiza los comentarios de los usuarios en tiempo real para detectar lenguaje ofensivo, bloquear comentarios inapropiados, notificar a los usuarios y registrar los comentarios bloqueados para la revisión del administrador. |
| **Propósito** | Mantener un entorno seguro y respetuoso en la plataforma de delivery mediante la moderación automatizada de comentarios. |
| Referencias Cruzadas | Casos de Uso: Publicar Comentarios, Analizar Sentimiento del Comentario, Notificar Usuario de Bloqueo, Registrar Comentario Bloqueado |
| Curso Normal de los Eventos | |
| **Acción de los Actores** | **Respuesta del Sistema** |
| 1. El usuario envía un comentario en la plataforma. | Recibe el comentario y lo envía al módulo de análisis. |
| 2. El sistema analiza el sentimiento del comentario. | Determina el tono emocional (positivo, neutro, negativo). |
| 3. El sistema detecta si el comentario es ofensivo. | Clasifica el comentario como ofensivo o no ofensivo. |
| 4. Si el comentario es positivo o neutro, el sistema lo publica. | Publica el comentario en la plataforma. |
| 5. Si el comentario es ofensivo, el sistema lo bloquea. | Bloquea el comentario y notifica al usuario sobre el bloqueo. |
| 6. El sistema registra el comentario bloqueado. | Guarda un registro del comentario bloqueado para revisión. |
| 7. El administrador revisa los registros de comentarios bloqueados. | Muestra el registro de comentarios bloqueados para revisión. |
| Cursos Alternativos | |
| 1a. El usuario envía un comentario que contiene lenguaje ambiguo. | El sistema marca el comentario para una revisión adicional. |
| 2a. El sistema no puede determinar con certeza si el comentario es ofensivo. | El comentario se envía a un moderador humano para revisión. |
| 3a. El moderador humano revisa el comentario y decide si debe ser publicado o bloqueado. | El sistema registra la decisión del moderador. |
| 4a. Si el comentario es considerado ofensivo por el moderador, el sistema bloquea el comentario. | Bloquea el comentario y notifica al usuario. |
| 5a. Si el comentario es considerado no ofensivo por el moderador, el sistema publica el comentario. | Publica el comentario en la plataforma. |
| 6a. El sistema registra todos los comentarios revisados manualmente. | Guarda un registro detallado de las revisiones manuales. |
| 7a. El administrador puede acceder a un reporte detallado de los comentarios revisados manualmente. | Muestra el reporte para revisión administrativa. |

### Diagramas de Casos de uso

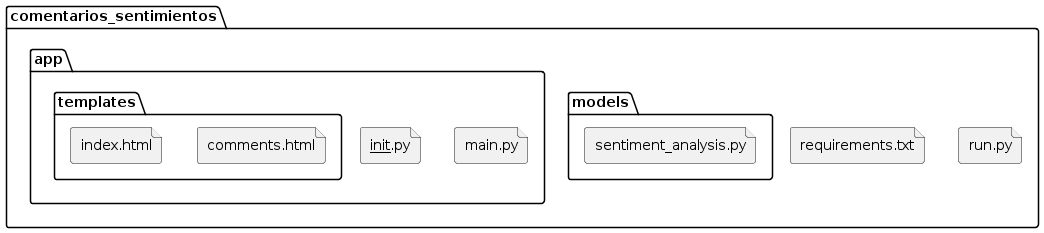


* 1. Vista Lógica

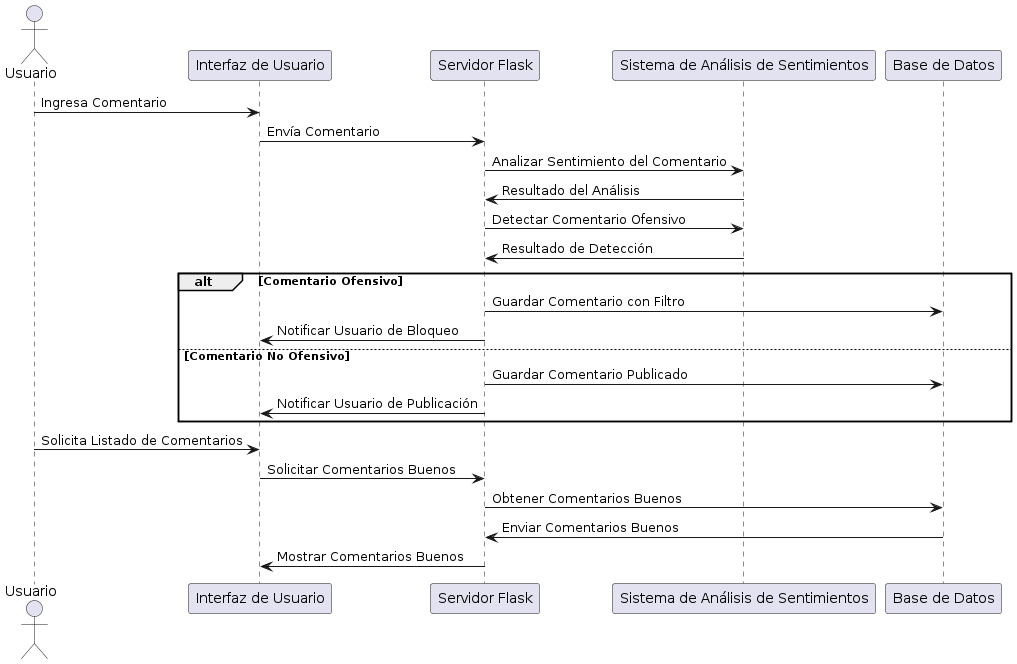


### 

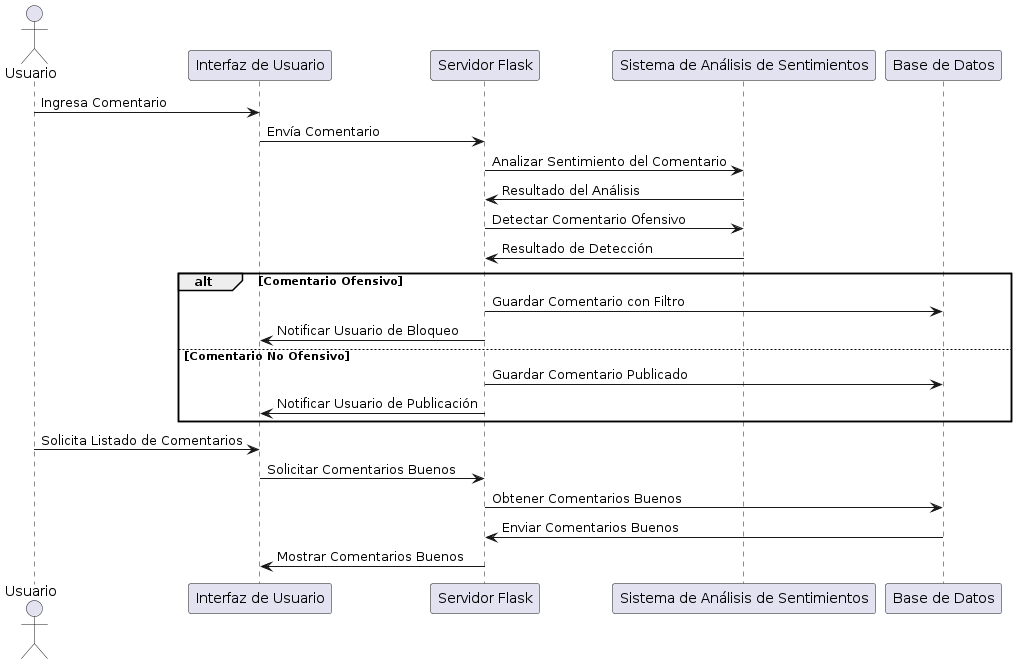
### Diagrama de Subsistemas (paquetes)



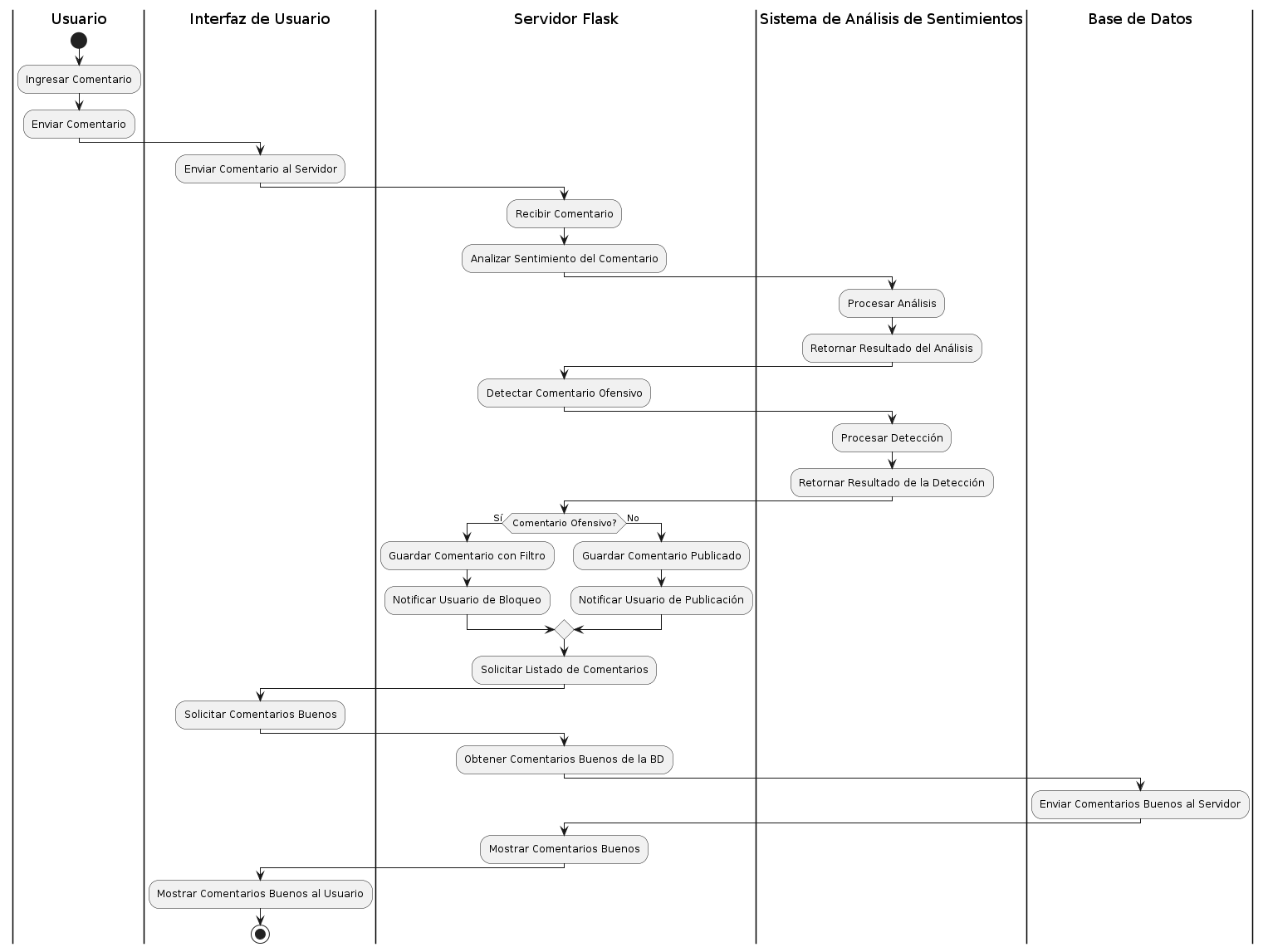
### Diagrama de Secuencia (vista de diseño)



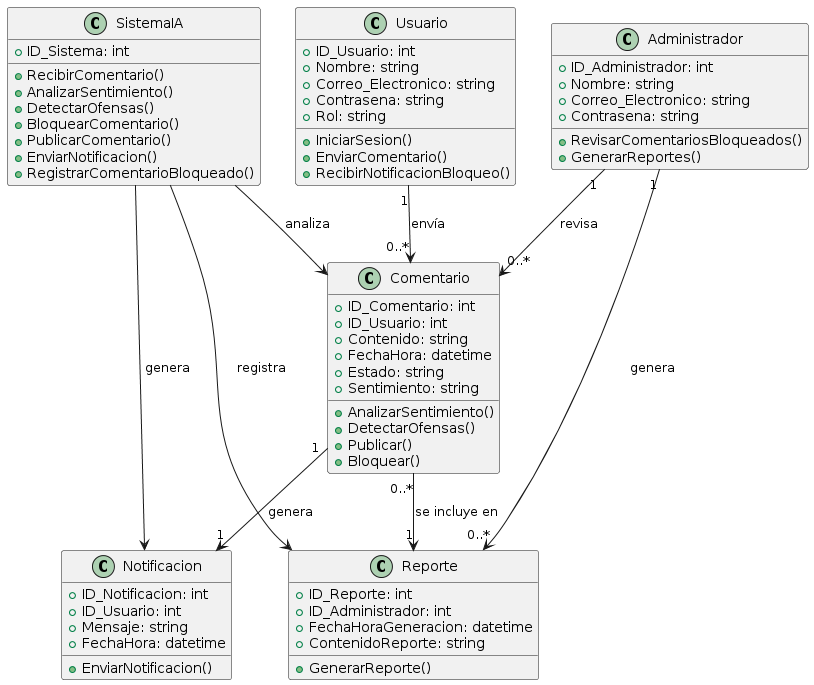
### Diagrama de Colaboración (vista de diseño)



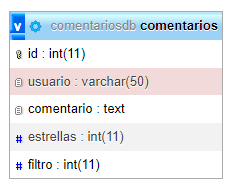
### Diagrama de Objetos



### Diagrama de Clases

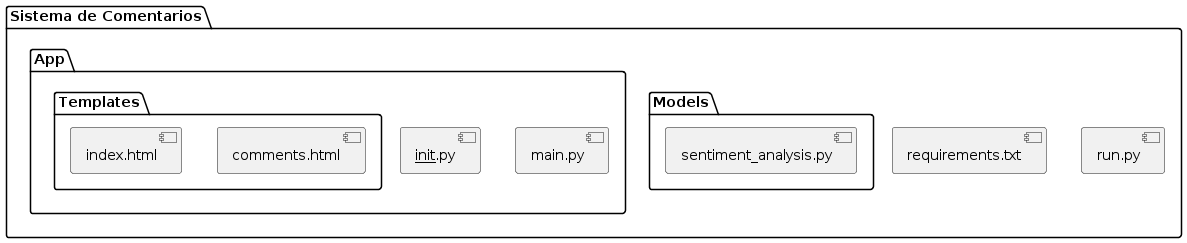
****

### Diagrama de Base de datos (relacional o no relacional)

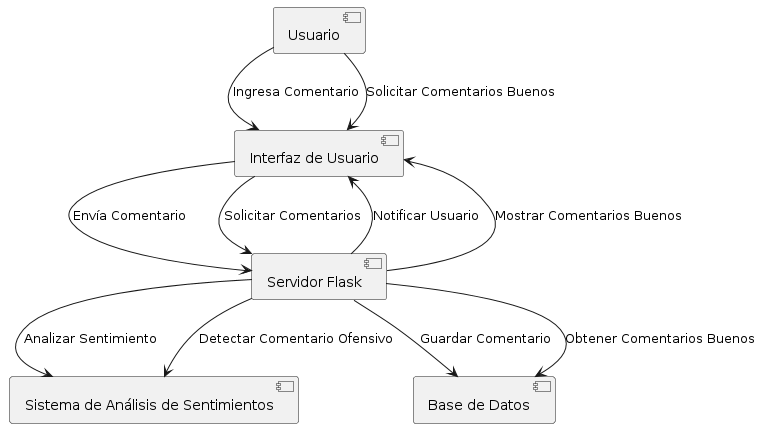


* 1. Vista de Implementación (vista de desarrollo)

### Diagrama de arquitectura software (paquetes)

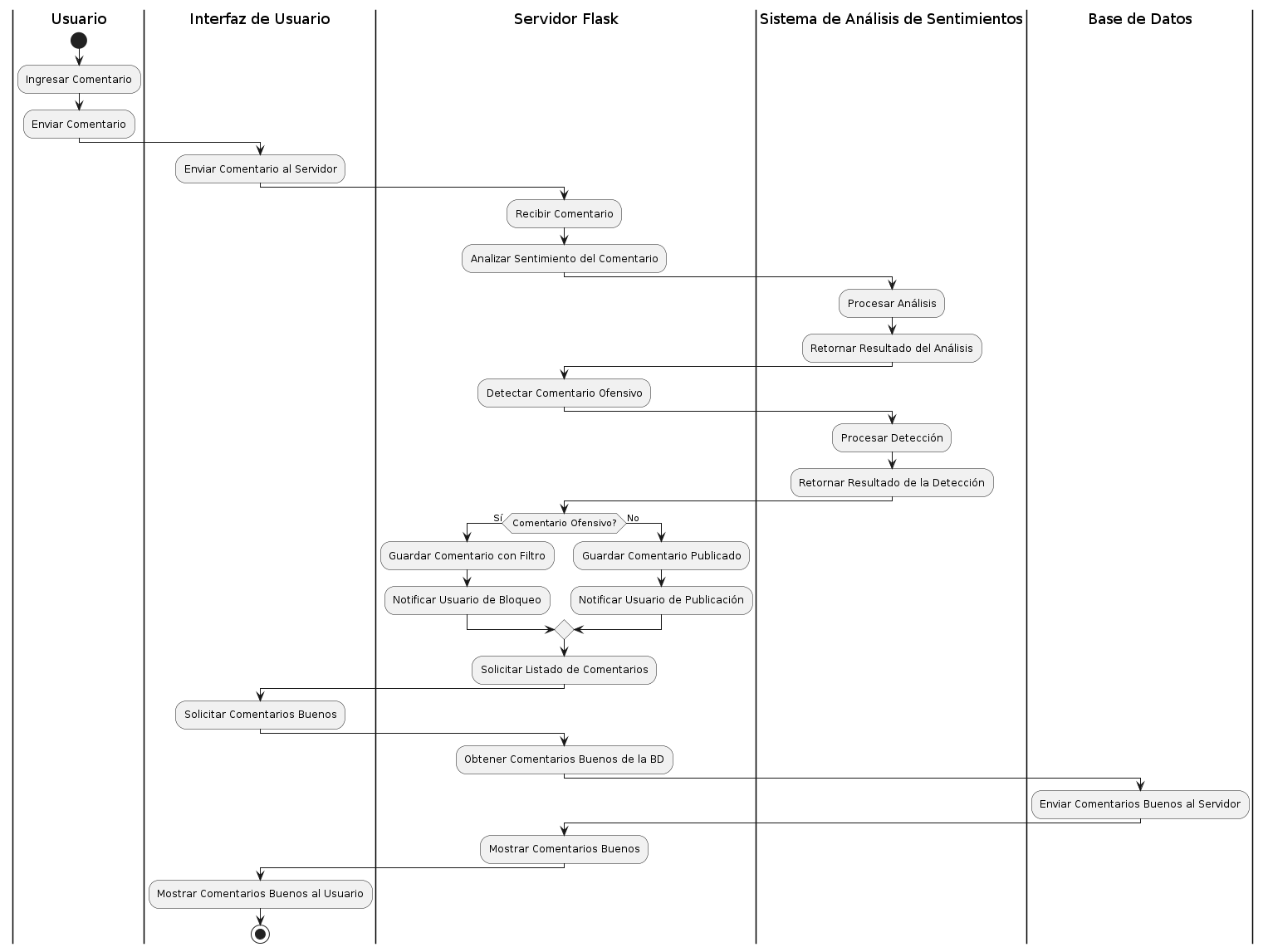


### Diagrama de arquitectura del sistema (Diagrama de componentes)



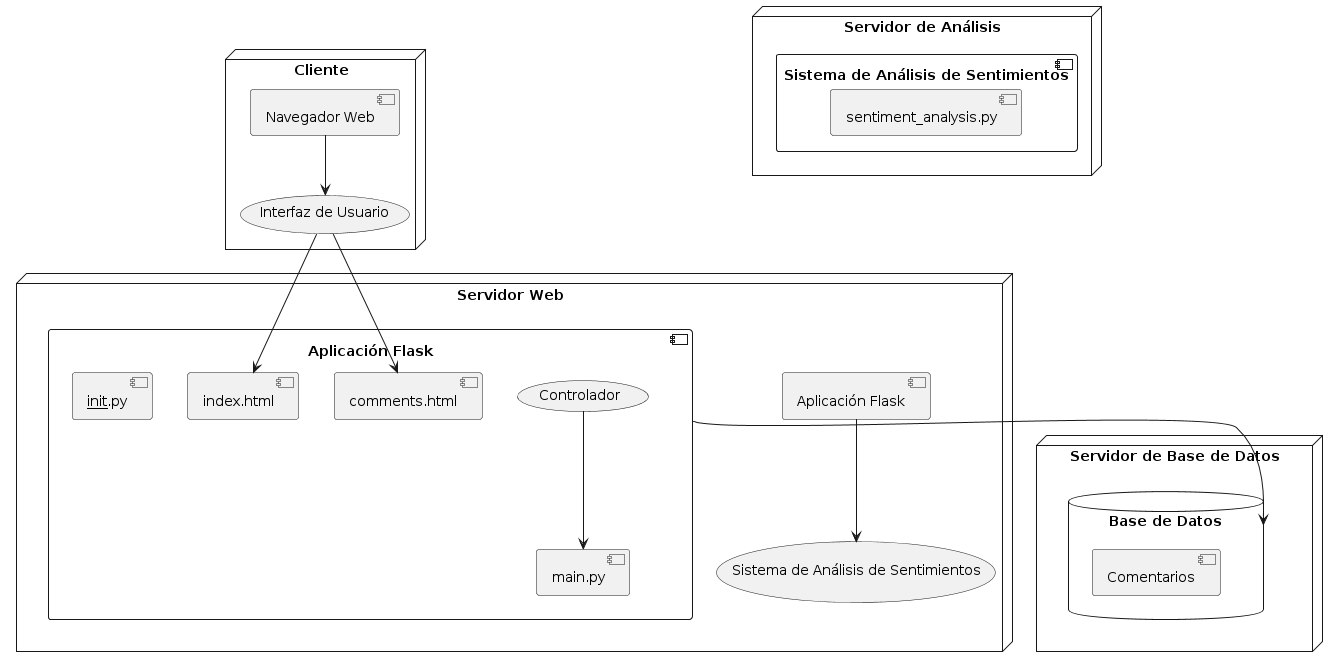
* 1. Vista de procesos

### Diagrama de Procesos del sistema (diagrama de actividad)



* 1. Vista de Despliegue (vista física)

### Diagrama de despliegue



# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

**Escenario de Funcionalidad**

La funcionalidad de nuestro sistema de análisis de comentarios se evalúa según las características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad general del sistema. El sistema está diseñado para:

* **Recibir y procesar comentarios:** Captura comentarios de los usuarios y los envía para su análisis.
* **Analizar el sentimiento:** Utiliza algoritmos de procesamiento de lenguaje natural (NLP) para determinar si un comentario es positivo, negativo o neutro.
* **Detectar lenguaje ofensivo:** Identifica comentarios ofensivos para bloquearlos o marcarlos según sea necesario.
* **Publicar comentarios válidos:** Almacena y muestra comentarios aprobados a otros usuarios.

**Escenario de Usabilidad**

La usabilidad de nuestro sistema se refiere a la facilidad con la que un usuario puede aprender a utilizar el sistema e interpretar sus resultados. La interfaz de usuario está diseñada para ser intuitiva y accesible, considerando los siguientes aspectos:

* **Aprendizaje del sistema:** Los nuevos usuarios pueden aprender rápidamente a ingresar y visualizar comentarios.
* **Utilización eficiente del sistema:** El sistema permite a los usuarios realizar tareas de manera eficiente, con una navegación clara y sencilla.
* **Minimización del impacto de errores:** El sistema maneja errores del usuario de manera efectiva, proporcionando mensajes de error claros y sugerencias para la corrección.
* **Adaptación del sistema a las necesidades del usuario:** La interfaz se adapta a las preferencias del usuario, mejorando su experiencia general.
* **Confianza y satisfacción:** Los usuarios se sienten satisfechos y confiados al utilizar el sistema debido a su facilidad de uso y eficiencia.

**Escenario de Confiabilidad**

La confiabilidad de nuestro sistema es esencial para garantizar la seguridad y disponibilidad de los datos. Se enfoca en:

* **Confidencialidad:** Protección de la información sensible de los usuarios.
* **Integridad:** Aseguramiento de que los datos no sean alterados o destruidos de manera no autorizada.
* **Irrefutabilidad:** Garantía de que las acciones realizadas por los usuarios sean registradas y verificables.
* **Disponibilidad:** Aseguramiento de que el sistema esté disponible y operativo cuando sea necesario.
* **Prevención, precaución y reacción:** Implementación de medidas preventivas, precauciones para minimizar impactos y capacidades de reacción ante incidentes de seguridad.

**Escenario de Rendimiento**

**El rendimiento de nuestro sistema se mide por:**

* **Velocidad de procesamiento:** Capacidad del sistema para analizar y procesar comentarios rápidamente.
* **Tiempo de respuesta:** Rapidez con la que el sistema responde a las solicitudes de los usuarios.
* **Uso de recursos:** Eficiencia en la utilización de recursos del servidor y la base de datos.
* **Eficiencia general:** Capacidad del sistema para manejar cargas de trabajo sin comprometer el rendimiento.

**Escenario de Mantenibilidad**

La mantenibilidad de nuestro sistema combina la capacidad para ser ampliable, adaptable y servicial. Se evalúa en términos de:

* **Extensibilidad:** Facilidad para añadir nuevas funcionalidades y mejorar las existentes.
* **Adaptabilidad:** Capacidad para modificar el sistema para satisfacer nuevas necesidades y requisitos.
* **Servicialidad:** Facilidad para realizar mantenimiento y reparaciones del sistema.

**Otros Escenarios**

**Performance:** La performance del sistema se refiere a su capacidad para responder de manera oportuna a eventos determinados y procesar una cantidad adecuada de eventos en un intervalo de tiempo dado. Este atributo asegura que el sistema pueda manejar picos de carga y mantener un servicio consistente.