**Desarrollo de Plataforma Integral para ConnectForo SPA**

“Diseño e Implementación de un Sistema de Gestión de Comunidades Digitales Personalizadas”

**Integrantes**

Gabriel Vidal

Francisco Argandoña

Mauricio Mateluna

**Asignatura**

DESARROLLO FULLSTACK I

**Docente**

Víctor Isidro Rosendo Lugo

INDICE

[DEFINICIÓN DEL PROBLEMA 4](#_Toc194341432)

[I. Sistema Actual: 4](#_Toc194341433)

[ESTRATEGIA DE MICROSERVICIOS 6](#_Toc194341434)

[**I. Tipo de Estrategia Utilizada** 6](#_Toc194341435)

[ **Estrategia por Dominio:** 6](#_Toc194341436)

[**II. Justificación Técnica y de Diseño:** 6](#_Toc194341437)

[ **La Estrategia Ideal para ConnectForo SPA:** 6](#_Toc194341438)

[ **Función del dominio:** 6](#_Toc194341439)

[ **Características técnicas:** 7](#_Toc194341440)

[1. Refleja la estructura real del negocio 7](#_Toc194341441)

[2. Escalabilidad Independiente 7](#_Toc194341442)

[3. Reducción del Acoplamiento 7](#_Toc194341443)

[4. Independencia Tecnológica y de Desarrollo 7](#_Toc194341444)

[5. Mayor Resiliencia y Tolerancia a Fallos 7](#_Toc194341445)

[6. Facilitación del Desarrollo Paralelo 7](#_Toc194341446)

[ **Diseño de Microservicios por Dominios:** 8](#_Toc194341447)

[1. Organización por Dominios: Se agruparon microservicios según el negocio, no por funcionalidad aislada. 8](#_Toc194341448)

[**2.** Uso de Bases de Datos Distribuidas: 8](#_Toc194341449)

[3. Autonomía y Despliegue Independiente: 8](#_Toc194341450)

[4. Alta Observabilidad y Monitoreo: 8](#_Toc194341451)

[**III. Beneficios de adoptar esta estrategia:** 8](#_Toc194341452)

[**1. Escalabilidad Independiente:** 8](#_Toc194341453)

[**2. Mayor Mantenimiento y Despliegue Ágil:** 9](#_Toc194341454)

[**3. Mejor Resiliencia y Aislamiento de Fallos:** 9](#_Toc194341455)

[**4. Tecnología Optimizada para Cada Dominio:** 9](#_Toc194341456)

[**5. Seguridad y Cumplimiento Mejorados:** 10](#_Toc194341457)

[**6. Flexibilidad para Integraciones Externas:** 10](#_Toc194341458)

[HERRAMIENTAS UTILIZADAS 11](#_Toc194341459)

[**Impacto de** la **Elección** 13](#_Toc194341460)

[HERRAMIENTAS DE TRABAJO COLABORATIVO 14](#_Toc194341461)

[ENFOQUE ÉTICO EN EL DESARROLLO 15](#_Toc194341462)

[**1.** **Privacidad de los datos:** 15](#_Toc194341463)

[**2.** **Seguridad:** 15](#_Toc194341464)

[**3.** **Responsabilidad en el despliegue:** 15](#_Toc194341465)

[**4.** **Impacto en los puestos de trabajo:** 15](#_Toc194341466)

[**5.** **Soluciones y buenas prácticas adoptadas:** 16](#_Toc194341467)

[**6.** **Consideraciones de cumplimiento normativo:** 16](#_Toc194341468)

[**Los aspectos claves de la ley son:** 16](#_Toc194341469)

[ANÁLISIS DE REQUERIMIENTOS 16](#_Toc194341470)

[**Síntesis de necesidades de cada perfil** 20](#_Toc194341471)

[ANÁLISIS DEL SISTEMA ACTUAL 22](#_Toc194341472)

[DISEÑO DE LA NUEVA ARQUITECTURA 23](#_Toc194341473)

[**Descripción general de la arquitectura basada en microservicios:** 23](#_Toc194341474)

[**Diagrama de Actores de Alto Nivel** 27](#_Toc194341475)

[**Diagrama de Casos de Uso** 28](#_Toc194341476)

[**Diagrama de Clases** 29](#_Toc194341477)

[PLANIFICACIÓN DE LA MIGRACIÓN 30](#_Toc194341478)

[**Fases de Migración** 30](#_Toc194341479)

[**Consideraciones de Compatibilidad** 30](#_Toc194341480)

[**Identificación de Riesgos y Plan de Mitigación** 30](#_Toc194341481)

[PLANIFICACIÓN ÁGIL Y CRONOGRAMA 31](#_Toc194341482)

[**Enfoque Ágil** 31](#_Toc194341483)

[**Roles:** 31](#_Toc194341484)

[CONCLUSIONES Y RECOMENDACIONES 32](#_Toc194341485)

[**Lecciones Aprendidas** 32](#_Toc194341486)

[**Potencial Escalabilidad del Sistema** 33](#_Toc194341487)

[**Posibles Mejoras Futuras** 33](#_Toc194341488)

[ANEXOS 34](#_Toc194341489)

# DEFINICIÓN DEL PROBLEMA

## I. Sistema Actual:

ConnectForo SPA, en su estado actual, consta con un sistema el cual opera con una arquitectura monolítica que depende de herramientas de terceros, para gestionar sus servicios de moderación y generación de contenido. Este enfoque presenta varias limitaciones que dificultan su escalabilidad, personalización y control de las comunidades digitales que la empresa busca fomentar.

* Arquitectura Monolítica:

Sistema Monolítico: La aplicación actual está construido como un sistema monolítico, donde todos los componentes y/o módulos como pueden ser: Gestión de usuarios, moderación, publicaciones, etc. Están desarrollados y se integran bajo un único bloque de software compartido. Esto lo hace difícil de escalar, tiene baja modularidad y conlleva más trabajo la manipulación componentes.

Esto trae consigo dificultades para que la empresa pueda adaptarse a las necesidades específicas de cada cliente, esto limita la innovación y la personalización de cada funcionalidad dentro del sistema.

* **Impacto de dicho problema en el negocio/proyecto**

Que la empresa no cuente con una plataforma que le permita actuar de manera eficiente y escalable nos afecta de manera negativa con los siguientes problemas

-Pérdida de control sobre la experiencia del usuario.

-Problemas para escalar con el crecimiento de clientes.

-Limitada capacidad de personalización y análisis.

* **Objetivo general**

Desarrollar una solución tecnológica basada en arquitectura de microservicios que reemplace el sistema actual monolítico, permitiendo control, escalabilidad y personalización y una mejor experiencia al cliente.

* **Objetivos específicos:**

-Dividir funcionalidades clave en microservicios independientes.

-Garantizar la escalabilidad y disponibilidad del sistema.

-Mejorar el monitoreo, moderación y análisis de contenido.

-Tener en cuenta los requisitos funcionales para el sistema, segun las necesidades de cada actor.

# ESTRATEGIA DE MICROSERVICIOS

## **I. Tipo de Estrategia Utilizada**

### **Estrategia por Dominio:**

(Domain Driven Design – DDD - Diseño Basado en Dominios):

La elección de una estrategia para ConnectForo SPA, es clave para garantizar que la migración a microservicios sea escalable, mantenible y alineada con la lógica del negocio. Teniendo esto en cuenta se decidió optar por una estrategia microservicios por dominios, agrupando todas las funcionalidades relacionadas.

¿Pero porque no ocupar una estrategia funcional para la estructura de los microservicios? Primero tenemos que entender que hace que sea la primera opción para este caso en particular, y para eso necesitamos conocerla de manera técnica.

## **II. Justificación Técnica y de Diseño:**

### **La Estrategia Ideal para ConnectForo SPA:**

Un Dominio es una agrupación lógica de microservicios relacionados que trabajan juntos para resolver una parte específica del negocio. Cada dominio tiene un conjunto de microservicios que gestionan funcionalidades que tienen que ver con un aspecto del sistema para ConnectForo SPA, como la autenticación, la gestión de foros, la moderación, etc.

Con este enfoque lo que se busca es dividir el sistema grande en microservicios agrupando estas funcionalidades relacionadas bajo cada dominio. Esta decisión se basa en necesidades técnicas, organizacionales y de escalabilidad.

### **Función del dominio:**

1. Agrupación de servicios relacionados: Los microservicios dentro de un dominio tienen responsabilidades similares o complementarias.

Ejemplo:

* En el dominio "Usuarios y Autenticación", los microservicios pueden ser ‘Servicio Usuario’, ‘Servicio Auth’, ‘Servicio Roles’, todos relacionados con la gestión de usuarios y seguridad.

1. Abstracción del negocio: El dominio refleja una parte específica del negocio, como gestión de usuarios, moderación de contenido o publicaciones. La idea es que un dominio sea una unidad coherente que puede ser gestionada, escalada y mejorada sin interferir con otras partes del sistema.

|  |  |  |
| --- | --- | --- |
| Dominio | Descripción | Microservicios Relacionados |
| Usuarios y Autentificación | - Registro, autenticación  - Perfiles y roles.  - Bloqueo/desbloqueo | Servicio Usuario  Servicio de Auth  Servicio de Roles |
| *Otro Dominio* | *...Funcionalidades* | *...Distintos Servicios* |

### **Características técnicas:**

Los microservicios por dominios tienes ciertas características técnicas que lo hacen una buena elección para un sistema complejo, y que también se alinea con la organización de ConnectForo SPA:

1. Refleja la estructura real del negocio: Equipos de moderación, gestión de foros, anuncios, y los demás.
2. Escalabilidad Independiente: Cada microservicio opera dentro de su propio contexto y puede escalarse individualmente según la demanda.
3. Reducción del Acoplamiento: Los microservicios con contextos acotados evitan compartir modelos de dominio, reduciendo dependencias entre ellos.
4. Independencia Tecnológica y de Desarrollo: Cada microservicio puede implementarse con tecnologías y bases de datos diferentes según sus necesidades.
5. Mayor Resiliencia y Tolerancia a Fallos: Si un microservicio falla, el resto del sistema puede seguir funcionando.
6. Facilitación del Desarrollo Paralelo: Diferentes equipos pueden trabajar en microservicios independientes sin bloquearse mutuamente.

Los microservicios junto con la estrategia de Diseños Basados en Dominios (DDD), ofrece un sistema modular, escalable y mantenible, ideal para entornos dinámicos y de alta disponibilidad. Siendo la Estrategia ideal para el desarrollo del sistema de ConnectForo SPA.

Comparación entre Microservicios: Por Funcional y por Dominio.

|  |  |  |
| --- | --- | --- |
| Criterio | Por funcionalidad | Por dominio |
| Interdependencia | Alta (Ej: Post, Comentarios, Likes, Reportes dependen entre sí). | Baja (Cada dominio gestiona sus propias reglas). |
| Transacciones | Complejas y distribuidas. | Simples dentro de un mismo contexto. |
| Escalabilidad | Difícil, ya que los módulos dependen entre sí. | Fácil, cada dominio escala de forma independiente. |
| Mantenimiento | Cambios pueden afectar varias funcionalidades. | Cambios aislados dentro del dominio. |

### **Diseño de Microservicios por Dominios:**

### Organización por Dominios: Se agruparon microservicios según el negocio, no por funcionalidad aislada.

Ejemplo:

* "Usuarios y Autenticación" puede tener los microservicios de User Service, Auth Service, Role Service.
* "Foros y Categorías" tiene Forum Service, Category Service.
* "Moderación y Reportes" tiene Moderation Service, Report Service.

1. Uso de Bases de Datos Distribuidas: Cada dominio maneja su propia base de datos optimizada para su carga de trabajo.
2. Autonomía y Despliegue Independiente: Cada servicio puede tener su propio ciclo de vida, desplegarse y mantenerse sin afectar otros.
3. Alta Observabilidad y Monitoreo: Se pueden detectar picos de tráfico en el servicio de foros y escalar automáticamente.

## **III. Beneficios de adoptar esta estrategia:**

### **1. Escalabilidad Independiente:**

* **Monolítico (*Antes*):**
* Si una funcionalidad experimenta alta demanda (ej. interacciones sociales), todo el sistema debe escalar, consumiendo recursos innecesarios.
* Un aumento en la carga puede afectar todo el sistema, causando ralentizaciones globales.
* **Microservicios Por Dominios (*Ahora*):**
* Cada dominio puede escalar de forma independiente según su demanda (ej. solo el servicio de foros si hay muchas publicaciones).
* Optimización de recursos: los módulos más usados pueden ejecutarse en servidores más potentes sin impactar los demás.

### **2. Mayor Mantenimiento y Despliegue Ágil:**

* **Monolítico (Antes):**
* Un pequeño cambio en una funcionalidad requiere desplegar todo el sistema, aumentando el riesgo de fallos.
* Desarrollo más lento, ya que múltiples equipos trabajan sobre el mismo código.
* **Microservicios Por Dominios (Ahora):**
* Despliegue modular: cada servicio se actualiza sin afectar al resto del sistema.
* Equipos pueden trabajar de forma independiente, acelerando el desarrollo.

### **3. Mejor Resiliencia y Aislamiento de Fallos:**

* **Monolítico (Antes):**
* Un fallo en un módulo (ej. notificaciones) puede derribar todo el sistema.
* Más difícil de depurar cuando ocurre un error crítico.
* **Microservicios Por Dominios (Ahora):**
* Si un microservicio falla, los demás siguen operativos.

### **4. Tecnología Optimizada para Cada Dominio:**

* **Monolítico (Antes):**
* Todo el sistema usa un solo stack tecnológico, lo que puede ser ineficiente para ciertos módulos (ej. usar una misma BD para todo).
* **Microservicios Por Dominios (Ahora):**
* Se pueden usar tecnologías especializadas por dominio:

Ej:

* Foros y Búsqueda: MySQL+ Elasticsearch AWS.
* Cada servicio usa el stack más eficiente, optimizando desempeño y costos.

### **5. Seguridad y Cumplimiento Mejorados:**

* **Monolítico (Antes):**
* Difícil aplicar reglas de seguridad modulares (ej. control de acceso por roles).
* Un ataque en una parte del sistema puede comprometer todo.
* **Microservicios Por Dominios (Ahora):**
* Autenticación y seguridad centralizada en el microservicio.
* Fácil de cambiar aplicar nuevas normas.
* Backups y restauración más eficiente.

### **6. Flexibilidad para Integraciones Externas:**

* **Monolítico (Antes):**
* Difícil conectar con APIs externas o servicios de terceros.
* Modificaciones en integraciones afectan todo el código.
* **Microservicios Por Dominios (Ahora):**
* Cada microservicio puede tener su propia API, facilitando integraciones con herramientas externas.

La estrategia de microservicios por dominios es la óptima para ConnectForo SPA porque:

1. **Resuelve los cuellos de botella del monolito**: Acoplamiento, escalabilidad y resiliencia.
2. **Se adapta a la estructura organizacional**: Equipos independientes por área (moderación, foros, etc.).
3. **Prepara el sistema para el futuro**: Cloud-native, fácil de integrar con IA (ej. moderación automática) o nuevos canales (ej. móvil)

**Ejemplo de comparación de infraestructuras:**

* **Problema actual en el monolito**:
* Si se añade una nueva regla de moderación (ej. "bloquear palabras ofensivas"), debe probarse todo el sistema.
* **Con Microservicios Por Dominios**:
* El cambio se hace solo en el servicio de moderación. Manejado dentro del Dominio correspondiente, en este caso de Moderación.
* Se despliega sin tocar otros servicios como foros o usuarios, o Dominios distintos.

# HERRAMIENTAS UTILIZADAS

Como se explicó en el capítulo anterior de la arquitectura de microservicios por dominios, una de las ventajas de esta es que cada dominio puede ser construido por tecnologías y herramientas que más le favorezca, a modo de sacar partido de los beneficios que puede entregar cada lenguaje de programación, framework, base de datos, entre otras tecnologías.

Las siguientes tablas representan los dominios y las tecnologías y herramientas con las que se diseñaron, pensado para el mejor desarrollo y eficacia. Los microservicios que existen dentro de cada Dominio pueden tener una o más tecnología implementada.

Las herramientas utilizadas dentro de este proyecto seran:

1. Java – Spring Boot: Para el desarrollo de los microservicios y la lógica de negocio.
2. MySQL: Almacenamiento en la base de datos.
3. AWS: Como Cloud para el despliegue, escalabilidad y gestión de la aplicación.

* **Funcionalidad Especifica y Justificación de la Elección:**

|  |  |  |
| --- | --- | --- |
| Tecnología | Funcionalidad Especifica | Justificación de la Elección |
| Java – Spring Boot | Desarrollo de microservicios escalables y seguros. | Spring Boot permite una estructura modular, con integración robusta para APIs REST, seguridad y gestión de dependencias. |
| MySQL | Almacenamiento relacional de datos estructurados. | MySQL ofrece integridad relacional, consultas eficientes y compatibilidad con Spring Boot. |
| AWS | Despliegue, escalabilidad y gestión de infraestructura. | AWS proporciona alta disponibilidad, balanceo de carga y servicios gestionados para bases de datos y almacenamiento. |

### **Impacto de** la **Elección**

1. **Spring Boot como framework principal**
   1. Facilita la creación de microservicios independientes y bien organizados.
   2. Incluye herramientas para autenticación, seguridad y procesamiento de datos.
   3. Compatible con MySQL y servicios en la nube de AWS.
2. **MySQL como base de datos**
   1. Maneja relaciones entre entidades como usuarios, roles, publicaciones y comentarios.
   2. Permite optimización de consultas mediante índices y caché.
   3. Compatible con Spring Data JPA para una gestión eficiente.
3. **AWS como infraestructura en la nube**
   1. Facilita el despliegue con servicios como EC2, RDS (para MySQL) y S3 (para almacenamiento de archivos adjuntos).
   2. Permite escalar recursos automáticamente con AWS Auto Scaling y Elastic Load Balancing.
   3. Seguridad y alta disponibilidad mediante IAM, CloudWatch y copias de seguridad automát

# HERRAMIENTAS DE TRABAJO COLABORATIVO

Las herramientas para el equipo de trabajo fueron fundamentales para la organización de información y también la coordinación del trabajo colaborativo. Algunas de las herramientas utilizadas son las siguientes:

Trello

Una herramienta clave para nuestro equipo, el propósito de su uso es la asignación de tareas y actividades para cada uno de los integrantes del equipo de trabajo. De este modo podemos seguir un flujo de trabajo estructurando, revisando los avances conociendo sus estados durante el transcurso del proyecto.

Enlace de nuestro tablero de Trello:

<https://trello.com/invite/b/67e071a37cd31bcae4edc257/ATTIfd84fcdb668486630b23bcda897ef09eD9618893/connectforo>

GitHub:

Su funcionalidad dentro del grupo de trabajo es mantener un repositorio común para el almacenaje de archivos importantes y relevantes para el desarrollo del proyecto.

Enlace a nuestro Repositorio de GitHub:

https://github.com/franar22/Grupo7

Discord:

Una aplicación de comunicación que utilizamos para la coordinación en tiempo real del equipo. Nos permite realizar reuniones virtuales, compartir actualizaciones instantáneas y discutir temas relacionados con el proyecto de manera eficiente.

# ENFOQUE ÉTICO EN EL DESARROLLO

### **Privacidad de los datos:**

La privacidad de los datos no solo es algo que todas las empresas deben de tener mirándolo de la manera técnica, sino que es una obligación ética, porque trata sobre el respeto que se le tiene a la información personal ya sea de los usuarios, clientes o trabajadores. Debido a que los usuarios confían en nosotros ellos toman en cuenta que sus datos no serán mal utilizados o compartidos sin su consentimiento.

### **Seguridad:**

La seguridad informática no solo es una responsabilidad técnica, sino también ética. Mas En una arquitectura basada en microservicios, donde cada componente se comunica con otros por medio de APIs, lo que incrementa la posibilidad de ataque cibernético si se llegara a Fallar en asegurar estos puntos de comunicación esto podría llevar a filtraciones o manipulación de datos, por lo cual de manera ética se les debe de asegurar activamente a los usuarios que se hace todo lo necesario para combatir contra accesos no autorizados, ataques o pérdida de datos.

### **Responsabilidad en el despliegue:**

Es ético asegurar que cualquier nuevo sistema o cambio, esté debidamente probado y validado antes de ser desplegado. Desplegar un sistema inestable o inseguro puede generar problemas en el servicio como, pérdida de datos o errores que afecten directamente a usuarios finales. Por lo cual, se debe mantener transparencia con los usuarios respecto a los cambios implementados, afectaciones posibles y el soporte disponible. Migrar sin considerar el impacto real puede interpretarse como una falta de responsabilidad profesional.

### **Impacto en los puestos de trabajo:**

Una migración tecnológica puede generar temor en los trabajadores que sienten que sus funciones pueden automatizarse o quedar obsoletas. Desde el enfoque ético, se debe considerar cómo la transición afecta a las personas involucradas en la operación del sistema actual, por lo cual es importante garantizarles procesos donde aprenderán a ocupar las nuevas herramientas disponibles, permitiéndoles no quedarse atrás, asegurando un avance en conjunto.

### **Soluciones y buenas prácticas adoptadas:**

La solución y buena práctica adoptada es el cambio en el enfoque de la empresa que no solo se piense de manera técnica donde buscando un crecimiento, no se toma en cuenta lo ético, Con este nuevo enfoque se toman en cuenta a todos los posibles afectados en las practicas que la empresa haga. No dejar que, por la ambición de ganar más dinero, nos lleve a cometer actos pocos éticos.

### **Consideraciones de cumplimiento normativo:**

Según la ley 19.628 La cual su objetivo principal es proteger la vida privada de todas las personas de chile, Por lo cual se establecieron normas sobra la recolección, almacenamiento, uso y comunicación de los datos personales.

### **Los aspectos claves de la ley son:**

* Consentimiento: EL trato de datos personales solo puede realizarse con el consentimiento expreso del individuo o cuando este autorizado por otras disposiciones legales.
* Derechos de los titulares: Los individuos tienen derecho a acceder, rectificar, cancelar u oponerse al tratamiento de sus datos personales.
* Definiciones importantes: La ley establece conceptos como almacenamiento, modificación, eliminación, bloqueo y comunicación de datos, así como distingue entre datos sensibles y no sensibles.

# ANÁLISIS DE REQUERIMIENTOS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R-N° | Nombre de Requerimiento | Tipo Requerimiento | Actores Relacionados | Descripción Requerimiento | Estado |
| 1 | Gestión de Usuarios | Funcional | Administrador | Permite crear, modificar y eliminar cuentas de administradores, moderadores y usuarios. | Por hacer |
| 2 | Configuración de Roles | Funcional | Administrador | |  | | --- | | Define y asigna permisos según el tipo de cuenta (admin, moderador, usuario). |      |  | | --- | |  | | Por hacer |
| 3 | Gestión de Categorías | Funcional | Administrador, Moderador | Crea y configura secciones temáticas dentro del foro. | Por hacer |
| 4 | Supervisión de Actividad | Funcional | Administrador | Permite monitorear publicaciones, comentarios y reportes en el foro. | Por hacer |
| 5 | Respaldar y Restaurar Datos | Funcional | Administrador, Soporte Técnico | Realiza copias de seguridad y permite restaurar el sistema en caso de fallos | Por hacer |
| 6 | Moderación de Contenido | Funcional | Moderador | Aprueba, edita o elimina publicaciones que infringen las normas. | Por hacer |
| 7 | Gestión de Reportes | Funcional | Moderador | Revisa denuncias sobre contenido inapropiado o spam. | Por hacer |
| 8 | Control de Conversaciones | Funcional | Moderador | Permite cerrar hilos, fijar temas o destacar publicaciones. | Por hacer |
| 9 | Bloqueo de Usuarios | Funcional | Moderador | Suspende o restringe cuentas por mal comportamiento. | Por hacer |
| 10 | Resolución de Incidencias | Funcional | Soporte Técnico | |  | | --- | |  |      |  | | --- | | Atiende problemas reportados por los usuarios. | | Por hacer |
| 11 | Monitoreo del Funcionamiento del Sistema | Funcional | Soporte Técnico | |  | | --- | |  |      |  | | --- | | Analiza métricas de rendimiento y detecta posibles fallos | | Por hacer |
| 12 | Publicación de Avisos Institucionales | Funcional | Gestor de Anuncios | |  | | --- | |  |      |  | | --- | | Publica comunicados globales o por categoría dentro del foro | | Por hacer |
| 13 | Segmentación de Mensajes | Funcional | Gestor de Anuncios | Envía anuncios dirigidos a perfiles o grupos específicos. | Por hacer |
| 14 | Programación de Avisos Emergentes | Funcional | Gestor de Anuncios | Muestra alertas destacadas en el sitio según su prioridad. | Por hacer |
| 15 | Historial de Comunicaciones | Funcional | Gestor de Anuncios | Permite acceder a anuncios previos del sistema. | Por hacer |
| 16 | Crear Cuenta | Funcional | Cliente/Usuario | Permite registrarse en la plataforma. | Por hacer |
| 17 | Iniciar Sesión | Funcional | Cliente/Usuario | Permite autenticarse en el sistema mediante credenciales. | Por hacer |
| 18 | Gestionar Perfil | Funcional | Cliente/Usuario | Permite actualizar información personal | Por hacer |
| 19 | Creación de Publicaciones | Funcional | Cliente/Usuario | |  | | --- | |  |      |  | | --- | | Permite crear nuevos temas o participar en hilos. | | Por hacer |
| 20 | Interacción Social | Funcional | Cliente/Usuario | Comentar, dar likes/dislikes, compartir y seguir temas. | Por hacer |
| 21 | Historial de Participación | Funcional | Cliente/Usuario | |  | | --- | |  |      |  | | --- | | Consulta de toda la actividad realizada en el foro. | | Por hacer |
| 22 | Generar Reportes (de publicaciones o soporte) | Funcional | Cliente/Usuario | |  | | --- | |  |      |  | | --- | | Permite reportar publicaciones y enviar formularios de contacto. | | Por hacer |
| 23 | Rendimiento y Escalabilidad | No funcional | Todos | |  | | --- | |  |      |  | | --- | | El sistema debe ser capaz de funcionar bajo alta carga y escalar horizontalmente | | Por hacer |
| 24 | Seguridad | No funcional | Todos | El sistema debe garantizar autenticación, autorización y cifrado de datos. | Por hacer |
| 25 | Usabilidad | No funcional | Cliente/Usuario | La interfaz debe ser clara, accesible y fácil de usar para todos los perfiles. | Por hacer |
| 26 | Disponibilidad | No funcional | Todos | El sistema debe estar disponible al menos el 99% del tiempo. | Por hacer |
| 27 | Respaldo automático | No funcional | Administrador, Soporte Técnico | El sistema debe realizar respaldos automáticos periódicos. | Por hacer |
| 28 | Tiempo de Respuesta del Sistema | No funcional | Cliente/Usuario | Las respuestas a interacciones deben demorar menos de 2 segundos. | Por hacer |

### **Síntesis de necesidades de cada perfil**

1. **Administrador:**

Los administradores son responsables de la gestión integral del sistema. Su rol está centrado en la administración general del foro, garantizando que la plataforma funcione de manera ordenada, segura y estructurada.

Necesidades:

* Gestionar cuentas de usuarios, moderadores y otros administradores.
* Asignar y configurar roles con distintos niveles de permisos.
* Supervisar la actividad del sistema para prevenir malas prácticas.
* Crear, modificar y eliminar categorías temáticas del foro.
* Realizar respaldos periódicos de la información y restaurarla en caso de fallos.

1. **Moderador:**

Los moderadores tienen un rol clave en mantener el orden y la buena convivencia dentro del foro. Son responsables de vigilar el contenido publicado por los usuarios y asegurar que se respeten las normas de la comunidad.

Necesidades:

* Aprobar, editar o eliminar publicaciones que infrinjan las normas.
* Atender y gestionar reportes realizados por los usuarios.
* Cerrar hilos que ya no son relevantes o que incumplen reglas.
* Fijar o destacar publicaciones importantes dentro de las categorías.
* Suspender temporalmente o bloquear a usuarios que incumplan reiteradamente las reglas.
* Administrar categorías temáticas específicas bajo su supervisión.

1. **Soporte Técnico:**

El equipo de soporte técnico se encarga de garantizar la disponibilidad, estabilidad y correcto funcionamiento del sistema. Su labor es principalmente técnica, orientada a prevenir y solucionar problemas operativos.

Necesidades:

* Atender incidencias técnicas reportadas por los usuarios.
* Supervisar el rendimiento del sistema y anticipar posibles fallos.
* Implementar medidas de respaldo y recuperación ante caídas o pérdida de información.
* Realizar mantenimiento a la infraestructura y al software para asegurar su funcionamiento continuo.

1. **Gestor de Anuncios:**

El gestor de anuncios es el encargado de comunicar información relevante a los usuarios del foro. Su rol es esencial para mantener informada a la comunidad, ya sea a través de avisos generales o mensajes dirigidos a grupos específicos.

Necesidades:

* Crear y publicar comunicados institucionales dentro del foro.
* Segmentar los anuncios según tipo de usuario o categoría temática.
* Programar alertas emergentes visibles en la interfaz del sitio.
* Consultar el historial de mensajes enviados anteriormente.

# ANÁLISIS DEL SISTEMA ACTUAL

Actualmente, el sistema se basa en una arquitectura monolitica,donde todos los componentes (autenticación, lógica de negocio, acceso a datos, interfaces de usuario, etc.) están agrupados en una única aplicación. Este indica que en el sistema se tiene Una estructura de despliegue unificada donde cualquier cambio requiere recompilar y desplegar toda la aplicacion  
  
Puntos débiles detectados  
  
- de los principales puntos débiles detectados en el sistema es la Falta de escalabilidad independiente, por lo cual, Si un módulo como por ejemplo el de pagos necesita más recursos por alta demanda, no se puede escalar por separado sin escalar toda la aplicación.  
  
-Otro punto débil detectado es que solo basta Una falla en un módulo y esto afecta a todo el sistema, generando posibles vulnerabilidades en el sistema.  
  
-Los equipos de desarrollo deben coordinarse constantemente, ya que todos trabajan sobre el mismo código base.  
  
-Al ser un sistema monolitico Todo el sistema debe usar el mismo lenguaje, framework o versión de librerías, lo que impide adoptar nuevas formas de trabajo.

|  |
| --- |
|  |

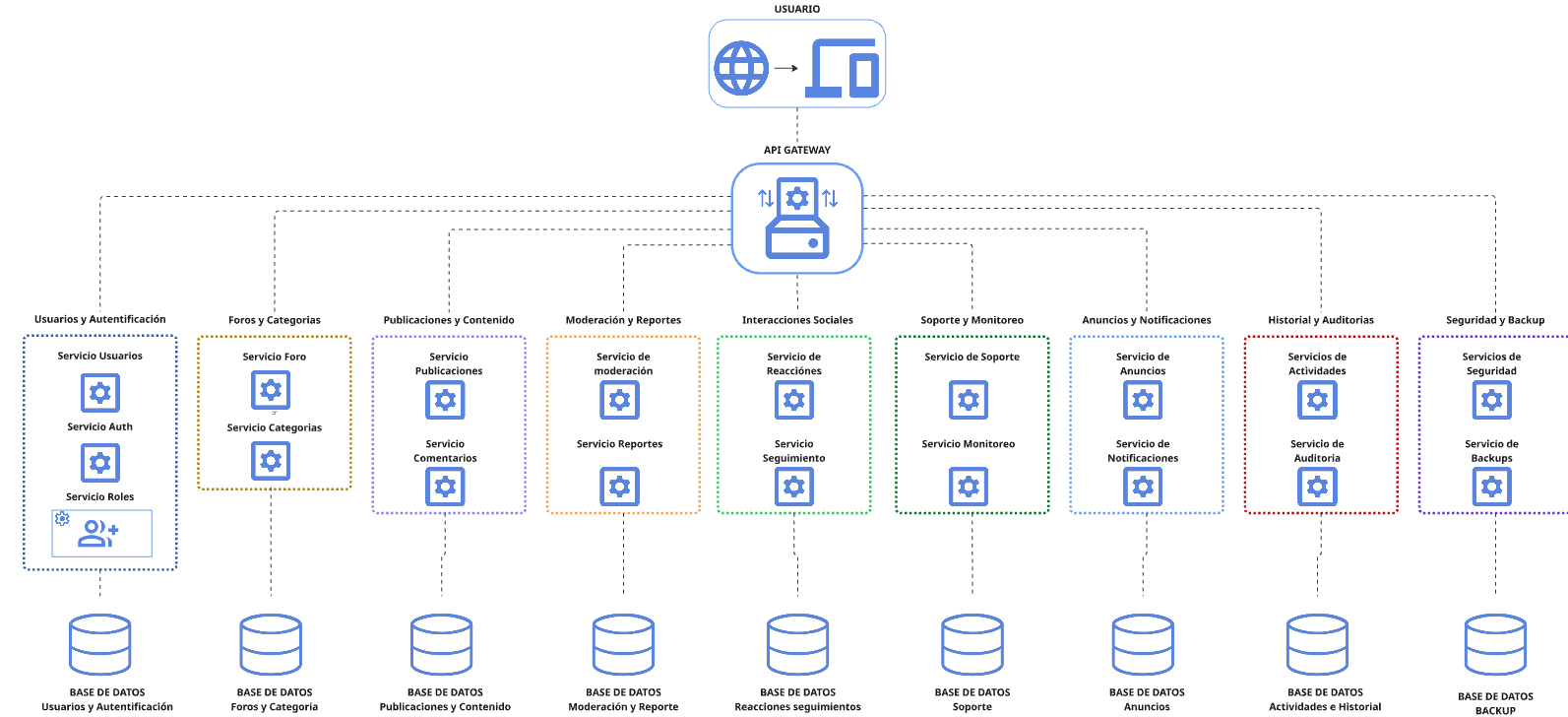
# DISEÑO DE LA NUEVA ARQUITECTURA

### **Descripción general de la arquitectura basada en microservicios:**

ConnectForo SPA basará su nueva arquitectura en una basada en microservicios, lo que permitirá escalabilidad, flexibilidad y un mantenimiento más eficiente. Cada componente principal del sistema ahora será desarrollado como un microservicio independiente, y comunicándose entre sí a través de API. Estos microservicios se encontrarán desacoplados, permitiendo actualizaciones y despliegues individuales sin afectar al resto del sistema.

Los microservicios principales son:

1. Usuarios y Autenticación.
2. Foros y Categorías.
3. Publicaciones y Contenido.
4. Moderación y Reportes.
5. Interacciones Sociales.
6. Soporte y Monitoreo.
7. Anuncios y Notificaciones.
8. Historial y Auditoría.
9. Seguridad y Backups.

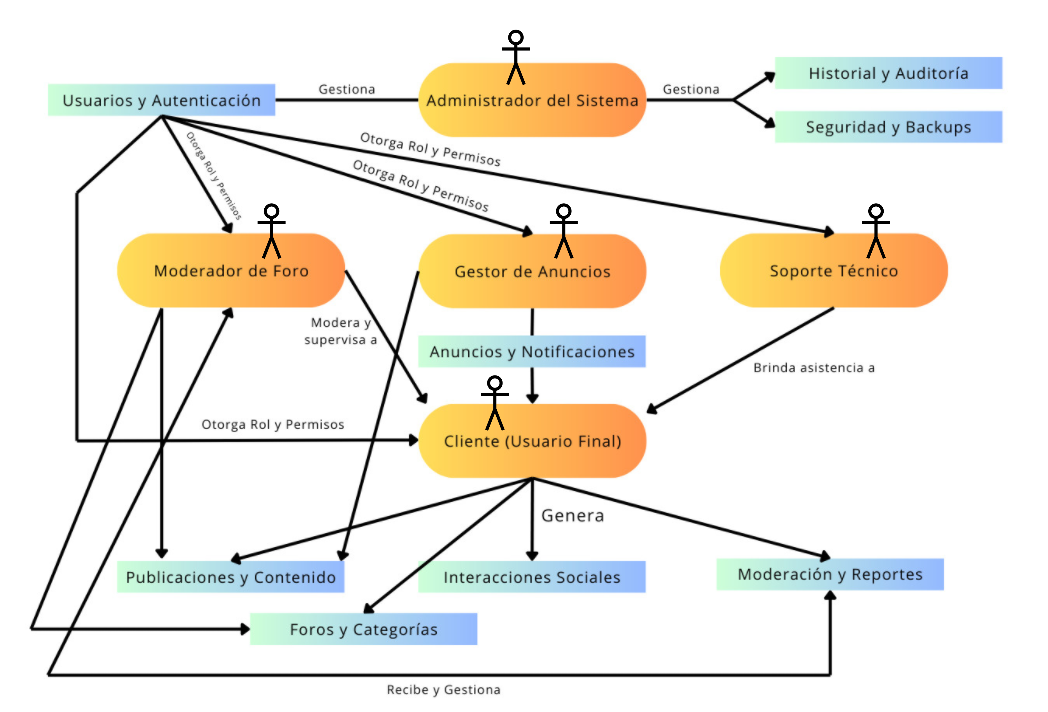


**Link al diagrama: https://tinyurl.com/2d2xfyth**

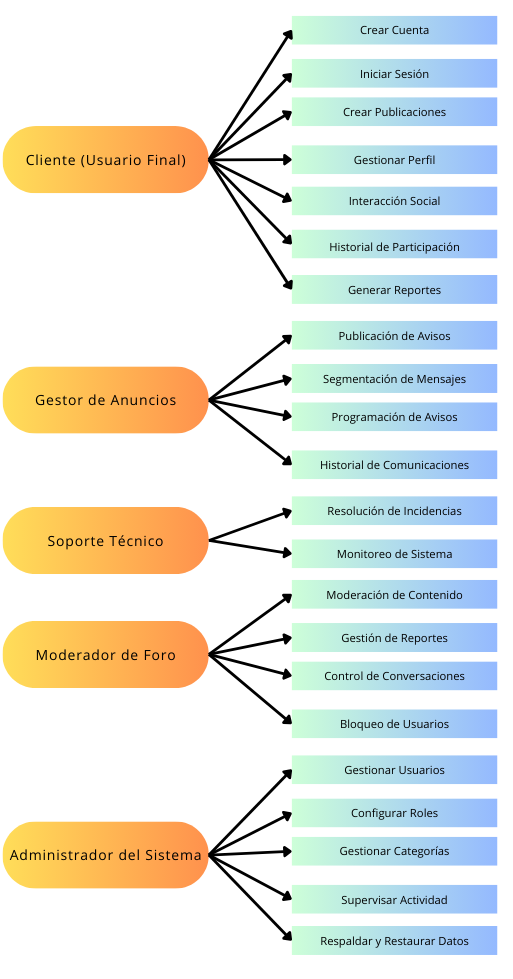
**Explicación del sistema:**

|  |  |  |
| --- | --- | --- |
| Dominio | Microservicio | Funcionalidad |
| Usuarios y Autentificación | User Service | Perfiles y configuración de usuario.  - Bloqueo y suspensión de cuentas (eventos con moderation-service). |
| Auth Service | - Registro e inicio de sesión |
| Role Service | - - Administración de roles y permisos. |
| Foros y Categorias | Forum Serivice | - Creación y administración de foros.  - Estadísticas generales del foro. |
| Category Service | - Organización y búsqueda de categorías.  - Creación y gestión de categorías. |
| Publicaciones y Contenido | Post Service | - Creación, edición y eliminación de publicaciones.  - Subida y gestión de archivos adjuntos.  - Indexación para búsqueda avanzada. |
| Comment Service | - Comentarios y respuestas en publicaciones.  - Edición y eliminación de comentarios.  - Moderación básica (detección de spam). |
| Moderación y Reportes | Moderation Service | - Gestión de reportes de contenido y usuarios.  - Políticas de comunidad (detección automática de lenguaje ofensivo).  - Bloqueo de usuarios y acciones disciplinarias (eventos con user-service). |
| Report Service | Gestión de reportes |
| Interacciones Sociales | Reaction Service | - Likes, dislikes, métricas.  - Tendencias y popularidad de contenido. |
| Follow Service | - Seguimiento de publicaciones y usuarios. |
| Soporte y monitoreo | Support Service | Gestión de tickets de soporte. |
| Monitoring Service | Recolección y visualización de métricas del sistema. |
| Anuncios y Notifiaciones | Announcement Service | - Publicación y segmentación de anuncios.  - Historial de anuncios y comunicados. |
| Notification Service | - Notificaciones en tiempo real.  - Alertas personalizadas por usuario.  - Envío de correos/SMS push. |
| Historial y Auditoria | Activity Log Service | - Logs de acceso y cambios de usuarios. |
| Audit Service | - Registro centralizado de eventos del sistema.  - Cambios en permisos, intentos de acceso no autorizados o modificaciones en configuraciones importantes. |
| Seguridad y Backups | Security Service | Protección y gestión de credenciales. |
| Backup Service | - Copias automatizadas de bases de datos.  - Restauración ante fallos. |

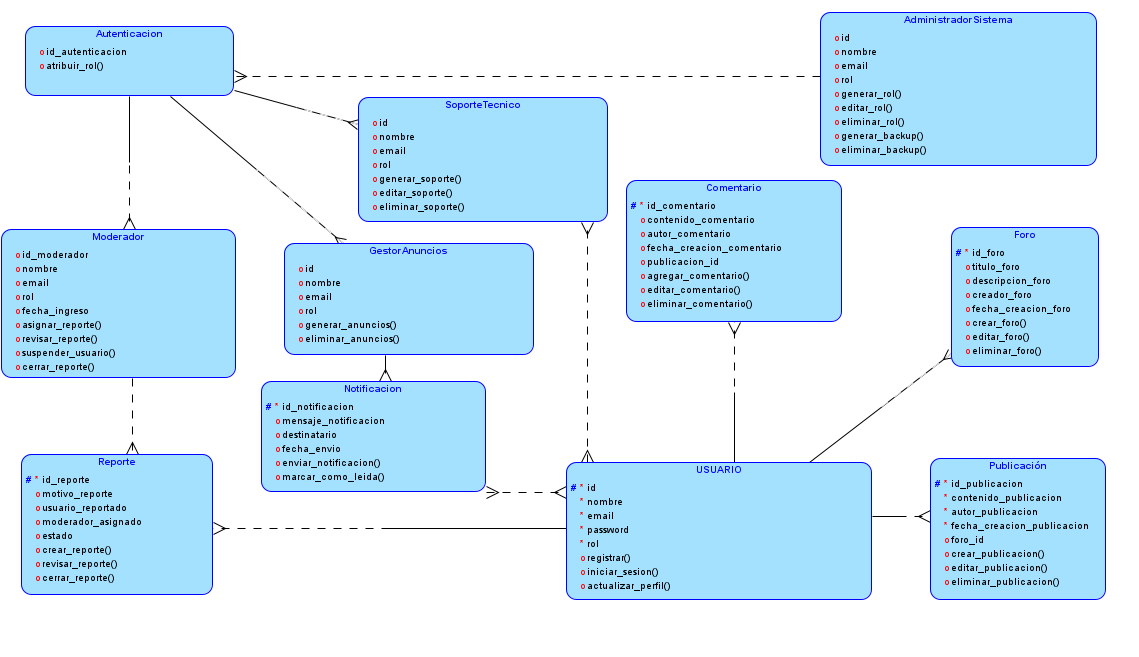
## **Diagrama de Actores de Alto Nivel**



## **Diagrama de Casos de Uso**



## **Diagrama de Clases**



# PLANIFICACIÓN DE LA MIGRACIÓN

### **Fases de Migración**

**Fase 1: Análisis y Planificación**

* 1. Evaluación de las herramientas actuales y limitaciones.
  2. Diseño de la arquitectura de microservicios.

**Fase 2: Desarrollo de Servicios Críticos**

* 1. Implementación de autenticación y gestión de usuarios.
  2. Creación del sistema de foros y publicaciones.

**Fase 3: Implementación de Funcionalidades Avanzadas**

* 1. Desarrollo del sistema de reportes y moderación.
  2. Integración del sistema de notificaciones y anuncios.

**Fase 4: Pruebas e Implementación Gradual**

* 1. Pruebas unitarias y de integración.
  2. Implementación progresiva con usuarios seleccionados.

**Fase 5: Despliegue y Monitoreo**

* 1. Lanzamiento de la plataforma.
  2. Monitoreo y optimización post-lanzamiento.

### **Consideraciones de Compatibilidad**

* Interoperabilidad con herramientas existentes hasta completar la migración.
* Integración con bases de datos ya en uso.

### **Identificación de Riesgos y Plan de Mitigación**

* **Riesgo**: Fallos en la autenticación.
  + *Mitigación*: Pruebas exhaustivas y backup de datos de usuarios.
* **Riesgo**: Falta de adopción por parte de usuarios.
  + *Mitigación*: Capacitación y documentación clara.

# PLANIFICACIÓN ÁGIL Y CRONOGRAMA

### **Enfoque Ágil**

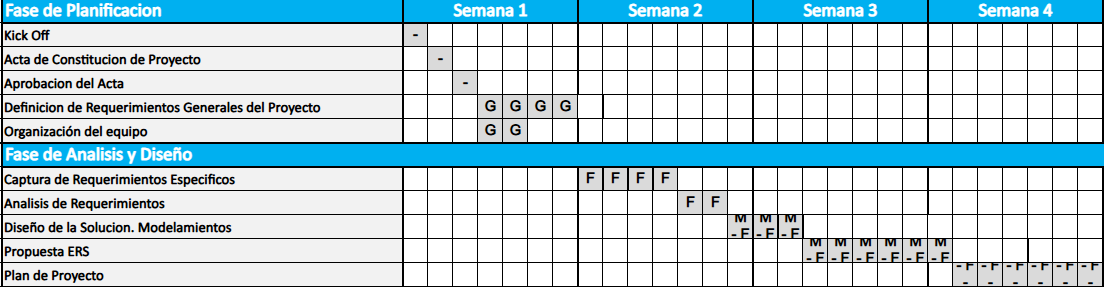
El desarrollo se basará en Scrum, el cual es un marco que se utiliza para gestionar proyectos de manera colaborativa e incremental. Se basa en ciclos de trabajo o “Sprint”, en los que se entregan partes del proyecto.  
  
Para este trabajo, se contemplarán iteraciones de 2 semanas, revisiones constantes y entregables incrementales.

### **Roles:**

En este proyecto, usando la metodología Scrum contemplaremos 3 roles principales. Los roles son:

* **CEO (Gabriel Vidal):** Dirige toda la empresa.
* **PO (Francisco Argandoña)**: Define el producto y su evolución.
* **CTO (Mauricio Mateluna):** Dirige la tecnología y desarrollo.

Se contempló la siguiente carta gantt para las primeras 4 semanas, que abarcan la Fase 1 de Planificación y Diseño.



# CONCLUSIONES Y RECOMENDACIONES

El desarrollo de la nueva plataforma para ConnectForo SPA, basada en una arquitectura de microservicios por dominios, representa un avance significativo en la modernización y escalabilidad del sistema. La migración desde una arquitectura monolítica hacia una estructura modular y desacoplada ha permitido abordar las limitaciones técnicas y operativas que presentaba el sistema anterior. Entre los logros más destacados se encuentran:

1. **Mayor Escalabilidad y Flexibilidad**: La adopción de microservicios ha permitido escalar componentes de manera independiente según la demanda, optimizando el uso de recursos y mejorando el rendimiento del sistema.
2. **Resiliencia y Tolerancia a Fallos**: El aislamiento de los microservicios garantiza que un fallo en un componente no afecte al sistema en su totalidad, mejorando la disponibilidad y la experiencia del usuario.
3. **Agilidad en el Desarrollo**: Los equipos pueden trabajar en paralelo en diferentes microservicios, acelerando el ciclo de desarrollo y permitiendo despliegues independientes.
4. **Mejora en la Seguridad y Cumplimiento Normativo**: La segmentación por dominios facilita la implementación de políticas de seguridad específicas y el cumplimiento de regulaciones como la Ley 19.628 sobre protección de datos personales.

## **Lecciones Aprendidas**

Durante el proceso de diseño e implementación, se identificaron varias lecciones clave:

* **Importancia del Diseño de Arquitectura de un Sistema**: Una planificación adecuada de los microservicios y su comunicación es esencial para evitar dependencias innecesarias y garantizar un sistema modular.
* **Importancia del Diseño Basado en Dominios (DDD)**: La estrategia de agrupar microservicios por dominios demostró ser la más adecuada para reflejar la estructura del negocio y reducir el acoplamiento entre componentes.
* **Desafíos en la Comunicación entre Microservicios**: La gestión de APIs y la sincronización de datos entre servicios requirió un enfoque cuidadoso para evitar latencias y garantizar la consistencia.
* **Cultura de Colaboración**: La transición a microservicios demandó una mayor coordinación entre equipos y una cultura de trabajo colaborativo, respaldada por herramientas como Trello y GitHub.

## **Potencial Escalabilidad del Sistema**

La arquitectura propuesta está diseñada para crecer de manera sostenible, tanto en capacidad como en funcionalidades. Entre las características que facilitan su escalabilidad se incluyen:

* **Escalabilidad Horizontal**: Cada microservicio puede replicarse según la demanda, utilizando servicios en la nube como AWS para gestionar recursos dinámicamente.
* **Integración con Tecnologías Emergentes**: La modularidad del sistema permite incorporar herramientas de inteligencia artificial (IA) para moderación automática, análisis de datos avanzados o chatbots de soporte.
* **Adaptabilidad a Nuevos Canales**: La plataforma puede extenderse para soportar aplicaciones móviles, APIs públicas o integraciones con redes sociales sin requerir cambios profundos en la arquitectura.

## **Posibles Mejoras Futuras**

Para asegurar el crecimiento continuo y la excelencia técnica, se recomienda considerar las siguientes mejoras:

1. **Adopción de Event-Driven Architecture**: Mejorando la eficiencia en procesos como notificaciones o actualizaciones en tiempo real.
2. **Enriquecimiento con IA y Machine Learning**:
   1. Moderación automática de contenido mediante NLP (Procesamiento de Lenguaje Natural).
   2. Recomendaciones personalizadas de contenido basadas en el comportamiento de los usuarios.
3. **Expansión de la Plataforma**:
   1. Integración con sistemas de pago para monetización de contenidos premium.
   2. Desarrollo de una API pública para permitir a terceros crear extensiones o integraciones propias.

La migración a una arquitectura de microservicios posicionaría a ConnectForo SPA como una plataforma moderna, escalable y preparada para los desafíos futuros. Con las mejoras planificadas y un enfoque en la innovación continua. Con esto ConnectForo se encuentra con un sistema confiable para su gestión de comunidades digitales personalizadas.

# ANEXOS

<https://www.linkedin.com/pulse/domain-driven-design-why-do-companies-need-approach-1e>

<https://medium.com/bimar-teknoloji/understanding-clean-architecture-and-domain-driven-design-ddd-24e89caabc40>

https://www.geeksforgeeks.org/domain-driven-design-ddd/