



PROGRAMA EDUCATIVO EN:

Tecnologías de la Información en Desarrollo de Software
Multiplataforma.

ASIGNATURA:

Arquitectura de software.

NOMBRE DE LOS INTEGRANTES:

Oscar Martinez Martinez.
Josmar Olivera Perez.

EMPRESA:

Facebook

FECHA:

07 septiembre del 2025

1. Introducción.

Facebook, que hoy en día se conoce como Meta, es una de las plataformas tecnológicas más influyentes en todo el mundo. Desde que se lanzó en 2004, ha evolucionado de ser una simple red social para universitarios a convertirse en un vasto ecosistema digital que conecta a más de 3 mil millones de usuarios activos cada mes. Esta enorme magnitud presenta desafíos significativos en términos de escalabilidad, disponibilidad, seguridad y experiencia del usuario, lo que convierte su arquitectura de software en un estudio de caso esencial para entender cómo las grandes empresas tecnológicas diseñan y desarrollan sus sistemas.

El crecimiento explosivo de Facebook no solo se debe a su innovación en productos, sino también a su habilidad para adaptar y rediseñar su infraestructura para satisfacer las demandas de millones de interacciones simultáneas. En sus primeros días, la plataforma se construyó como un monolito en PHP, alojado en servidores convencionales. Sin embargo, a medida que aumentó el número de usuarios, publicaciones, fotos y videos, esta arquitectura inicial se volvió insuficiente para garantizar el rendimiento y la disponibilidad que se esperaban. Para enfrentar estos desafíos, Facebook implementó un ecosistema distribuido y altamente escalable, utilizando tecnologías propias y estándares abiertos, lo que le permitió mantener un tiempo de actividad cercano al 100% y ofrecer baja latencia en cualquier parte del mundo.

Uno de los aspectos más destacados de la arquitectura de Facebook es su capacidad de evolución constante. A lo largo de los años, la compañía ha desarrollado tecnologías que se han convertido en referentes en la industria, como React para la creación de interfaces, GraphQL para optimizar las consultas de datos y TAO, su sistema distribuido para gestionar el grafo social. Además, ha implementado soluciones avanzadas de balanceo de carga global, caching distribuido y procesamiento asíncrono de eventos, lo que le permite manejar millones de solicitudes por segundo sin comprometer la experiencia del usuario.

Analizar la arquitectura de Facebook es realmente interesante, ya que es un claro ejemplo de ingeniería a gran escala en su máxima expresión. Al estudiar cómo la compañía aborda desafíos como la escalabilidad, la consistencia de datos, la seguridad y la tolerancia a fallos, podemos extraer valiosas lecciones que son útiles para proyectos de cualquier tamaño. Este informe se enfocará en detallar los estilos arquitectónicos que ha adoptado Facebook, los componentes clave de su infraestructura, así como los requisitos funcionales y no funcionales. También se incluirá un análisis crítico de los atributos de calidad que han sido fundamentales para el éxito de esta plataforma.

Finalmente, se ofrecerán conclusiones sobre las fortalezas, limitaciones y lecciones aprendidas que podrían ser útiles en proyectos futuros, subrayando la importancia de diseñar arquitecturas que logren un equilibrio entre escalabilidad, simplicidad y mantenibilidad.

2. Arquitectura de Facebook

La arquitectura de software de Facebook es un ejemplo asombroso de escalabilidad, resiliencia y eficiencia, diseñada para manejar miles de millones de usuarios activos en todo el mundo. Desde sus inicios, la plataforma ha ido evolucionando constantemente, pasando de un monolito desarrollado en PHP a un ecosistema complejo que integra microservicios, sistemas distribuidos y componentes especializados, todo con el objetivo de asegurar baja latencia, alta disponibilidad y un rendimiento óptimo.

Estilo arquitectónico

- Facebook ha evolucionado bastante. En sus inicios, funcionaba como un gran monolito en un centro de datos propio.
- Hoy en día, ha adoptado una arquitectura de microservicios y servicios distribuidos. Esto significa que cada funcionalidad, ya sea el feed, el chat, los anuncios, etc., opera como un servicio independiente. Esto les permite escalar y actualizar cada parte sin que afecte a las demás.
- Además, utilizan un monorepo que simplifica la gestión del código y la integración continua.
- También aplican técnicas orientadas a eventos y sistemas de colas para el procesamiento asíncrono, lo que optimiza la forma en que manejan tareas en tiempo real.

Componentes principales y funciones.

- Frontend: Utiliza React, una biblioteca que Facebook creó para construir interfaces dinámicas y muy interactivas. También integra GraphQL, que también fue desarrollado por la misma empresa, para hacer consultas de datos de manera más eficiente.
- Balanceo de carga: Facebook utiliza herramientas internas como Proxygen y Katran (basado en eBPF) para distribuir el tráfico a nivel global y asegurar la disponibilidad.
- Almacenamiento y bases de datos:
 - ◆ MySQL es la base relacional principal, empleando técnicas de sharding para dividir la información en varios servidores.
 - ◆ TAO: Un sistema diseñado para gestionar el grafo social con baja latencia.
 - ◆ Haystack: Especializado en el almacenamiento masivo de fotos.
 - ◆ ZippyDB: Basado en RocksDB, optimizado para el almacenamiento distribuido de claves y valores.
 - ◆ Cassandra: Para manejar datos NoSQL con alta escalabilidad.
- Capa de caching: Se hace un uso intensivo de Memcached y Mcrouter para reducir el tiempo de acceso a los datos y mejorar la experiencia del usuario.

- Procesamiento de eventos: Implementa FOQS (Facebook Ordered Queueing Service) y LogDevice para gestionar de manera eficiente colas y registros distribuidos
- Orquestación y despliegue: Facebook utiliza Tupperware, su propio sistema de contenedores, que es similar a Kubernetes, para gestionar microservicios y despliegues masivos.
- Analítica: Herramientas internas como Scuba y Presto permiten realizar análisis en tiempo real y consultas rápidas sobre grandes volúmenes de datos.

API y herramientas para desarrolladores

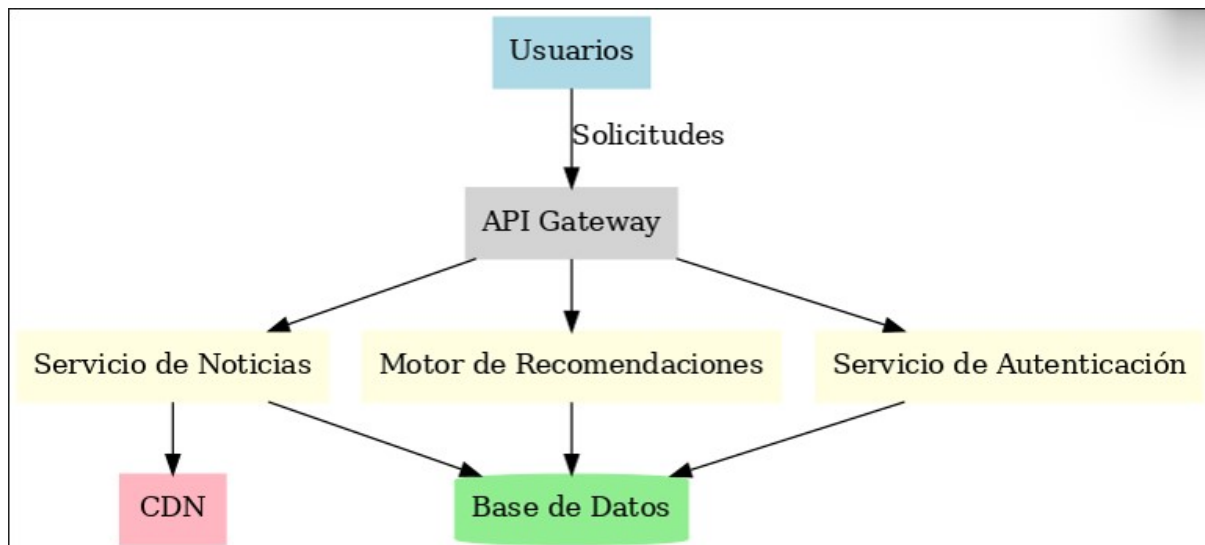
Además de su arquitectura interna, Facebook ofrece APIs que permiten crear soluciones innovadoras:

- Chatbots Inteligentes: A través de las APIs de WhatsApp y Messenger, puedes desarrollar chatbots que automatizan la atención al cliente.
- Aplicaciones de Gestión de Redes Sociales: Las APIs permiten programar publicaciones, analizar métricas y gestionar interacciones.
- Plataformas de Juegos Sociales: Integración con Facebook para crear experiencias de juego compartidas.
- Herramientas de Análisis de Datos: APIs que permiten a las empresas llevar a cabo análisis personalizados sobre interacciones y audiencias.
- Soluciones de Comercio Electrónico: Integración de opciones de compra y pago dentro de aplicaciones, impulsando el e-commerce.

Justificación del diseño.

El tamaño y la complejidad de Facebook requieren una arquitectura distribuida, modular y resiliente, capaz de manejar millones de solicitudes por segundo con baja latencia global, al mismo tiempo que asegura la seguridad, escalabilidad y tolerancia a fallos. La implementación de microservicios, un sistema de caching masivo, bases de datos especializadas y un balanceo inteligente permite que la plataforma mantenga su posición como la red social líder a nivel global.

Diagrama simplificado (C4 – Container level):



3. Requerimientos

3.1 requerimientos funcionales

Los requerimientos funcionales de Facebook se centran en lo que el sistema debe hacer, como crear una cuenta, iniciar sesión, publicar contenido y buscar amigos.

- Gestión de perfiles: Crear y editar un perfil de usuario.
- Autenticación: Iniciar sesión con credenciales y recuperar contraseñas.
- Publicación de contenido: Permite a los usuarios crear y compartir publicaciones.
- Interacción: Funciones para dar "Me gusta", comentar y compartir publicaciones.
- Conexiones sociales: Buscar y añadir amigos, seguir a otras personas.
- Mensajería: Enviar y recibir mensajes a otros usuarios.
- Gestión de grupos y páginas: Crear y administrar comunidades.

3.2 requerimientos no funcionales

Los no funcional aborda aspectos como el rendimiento, la seguridad y la usabilidad. Por ejemplo, un requerimiento funcional podría especificar que un usuario debe iniciar sesión con su correo electrónico y contraseña, mientras que un requerimiento no funcional indicaría que el sistema debe estar disponible el 99.9% del tiempo. Esta distinción es clave en el desarrollo de software, ya que ambas categorías deben considerarse para garantizar un producto que satisfaga las expectativas del usuario.

1. Rendimiento:La rapidez con la que se carga la página, se procesan las solicitudes y se muestra el contenido
2. Disponibilidad:El tiempo que la plataforma está activa y accesible para los usuarios, por ejemplo, que esté el 99.9% del tiempo en línea.
3. Seguridad:Medidas para proteger la información de los usuarios contra accesos no autorizados y otras amenazas.
4. Usabilidad:Qué tan fácil y amigable es la interfaz para que los usuarios naveguen y utilicen las funciones de la plataforma.
5. Escalabilidad:La capacidad del sistema para manejar un número creciente de usuarios y datos sin perder eficiencia.
6. Compatibilidad:Que el software funcione correctamente en diferentes dispositivos, navegadores y sistemas operativos.
7. Fiabilidad:Que el sistema funcione de manera consistente y sin fallos, incluso bajo cargas pesadas.

4. Atributos de calidad y análisis crítico

Los atributos de calidad en Facebook se centran en la relevancia y la experiencia del usuario para la red social y sus anunciantes, medidos por la interacción del usuario como lo son likes y comentarios la calidad del contenido si se oculta, se reporta y el cumplimiento de políticas publicitarias.

Un análisis crítico de Facebook revela que, a pesar de su amplio alcance, la plataforma enfrenta desafíos relacionados con la veracidad de la información, la privacidad de los datos, la generación de adicción y la concentración del poder en pocas manos, factores que impactan la experiencia del usuario y el impacto social de la red.

- Interacciones Sociales:La cantidad de "me gusta", comentarios, compartidos y guardados en una publicación o anuncio indica interés y relevancia.

- Comportamiento del Usuario: La frecuencia con la que los usuarios ocultan, reporta o evitan ver anuncios de un anunciante es una señal directa de baja calidad.
- Calidad de Anuncios: Se evalúa la relevancia y utilidad de los anuncios y sus páginas de destino para el público objetivo, evitando prácticas de baja calidad.
- Transparencia y Confiabilidad: La plataforma implementa herramientas para verificar la integridad de los datos y detectar anomalías en los análisis, lo que ayuda a mantener la calidad y confiabilidad de la información publicada
- Métricas de Rendimiento: Métricas como el alcance, el sentimiento (análisis de emociones en los comentarios) y la cuota de voz son utilizadas para evaluar la calidad y el impacto del contenido

Análisis Crítico de Facebook

Un análisis crítico de Facebook debe considerar su impacto a nivel individual y social.

Veracidad de la información: La facilidad con la que se propaga desinformación y noticias falsas puede tener graves consecuencias sociales y políticas.

Privacidad de los datos:

La recopilación masiva de datos personales y su uso para fines publicitarios plantea preocupaciones sobre la privacidad y la manipulación de los usuarios.

Concentración de poder: La vasta influencia de Facebook en la comunicación y el mercado de la publicidad digital genera un riesgo de monopolio y falta de competencia.

5. Conclusión.

El análisis de la arquitectura de Facebook revela que el éxito de una plataforma global no se basa solo en lo que ofrece a los usuarios, sino también en su capacidad técnica para mantener un sistema que sea altamente escalable, resiliente y eficiente. La transición de un monolito en PHP a una arquitectura distribuida con microservicios, junto con el uso de técnicas avanzadas como el procesamiento asíncrono, caching y orquestación, ha permitido a Facebook ofrecer un rendimiento óptimo, incluso ante una demanda masiva y en constante aumento.

Uno de los aspectos más impresionantes es cómo la empresa ha creado tecnologías propias para enfrentar desafíos específicos, como TAO para gestionar el grafo social, Haystack para almacenar imágenes de manera eficiente y Tupperware para la orquestación de microservicios. Además, la adopción de React y GraphQL ha representado un cambio significativo en la forma de construir interfaces rápidas y optimizar consultas, influyendo incluso en otras empresas y desarrolladores independientes.

Las cualidades que definen esta arquitectura —escalabilidad, disponibilidad, tolerancia a fallos y mantenibilidad— son el resultado de decisiones estratégicas diseñadas para manejar millones de solicitudes por segundo, asegurando baja latencia y seguridad. Sin embargo, estas ventajas vienen con importantes desventajas, como la complejidad operativa, la necesidad de contar con personal altamente especializado y los altos costos de infraestructura a nivel global.

6. Referencias

Arquitectura e ingeniería en comunicación social de Facebook. (s/f). En *Los jóvenes, el servicio de redes sociales en Internet y el futuro* Jesús Galindo Cáceres.

Artiga, R. (2023, octubre 10). *Arquitecturas de software de facebook y WhatsApp: Oportunidades de desarrollo de nuevas herramientas a través de sus APIs*. Tech Blogs by TechAffinity; TechAffinity. <https://techaffinity.com/blog/arquitecturas-de-software-de-facebook-y-whatsapp/>

Facebook company. (2025, junio 18). *Introducing passkeys on Facebook for an easier sign-in*. Meta. <https://about.fb.com/news/2025/06/introducing-passkeys-facebook-easier-sign-in/>

(S/f). Facebook.com. Recuperado el 17 de mayo del 2020, de <https://developers.facebook.com/docs/>

Jain, A. (2025, mayo 7). ¿Qué son los requisitos no funcionales? Tipos, ejemplos y enfoques. Visure Solutions. <https://visuresolutions.com/es/gu%C3%ADa-de-almosna/requerimientos-no-funcionales/>

¿Qué es Facebook y Cómo Funciona? (2024, septiembre 18). Clicacs.com. <https://clicacs.com/que-es-facebook-y-como-funciona/>
Requerimientos no funcionales: Ejemplos. (s/f). Pmoinformatica.com. Recuperado el 8 de septiembre de 2025, de <https://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>

Requisitos no funcionales: ¿Por qué son importantes? (s/f). Gluo. Recuperado el 8 de septiembre de 2025, de <https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes>

Requisitos funcionales. (s/f). Scribd. Recuperado el 8 de septiembre de 2025, de <https://es.scribd.com/document/789045260/Requisitos-funcionales>
Requisitos no funcionales: ¿Por qué son importantes? (s/f). Gluo. Recuperado el 8 de septiembre de 2025, de <https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes>