

INTRODUCCIÓN

INTRODUCCIÓN AL CURSO

Daniel Blanco Calviño

ARQUITECTURA DE SOFTWARE

Representa todas las decisiones de diseño de nuestro sistema que afectan a su estructura, comportamiento y a la interacción de sus componentes internos. Es una **descripción a alto nivel de todo lo que comprende el sistema.**





REQUISITOS FUNCIONALES

- Recogen todas las **funcionalidades**.
- Pasos a seguir:
 - **Identificar a los actores.**
 - Administrador, vendedor, comprador.
 - **Recopilar casos de uso.**
 - Pago del contenido del carrito por parte de un comprador.
 - Añadir un nuevo producto al catálogo por parte de un vendedor.
 - **Diagramas de flujo.**
 - Interacciones entre los distintos componentes y actores en cada caso de uso.

1.insertar diagrama flujo

REQUISITOS NO FUNCIONALES

- Factores que **más impactan** a la arquitectura del sistema:
 - No es lo suficientemente **rápido**.
 - No es suficientemente **seguro**.
 - No soporta bien tantos **usuarios / peticiones al mismo tiempo**.
 - No es sencillo **añadir o modificar funcionalidades**.
 - No es lo suficientemente **fiable**

REQUISITOS NO FUNCIONALES

También llamados requisitos de calidad. **Deben ser medibles.**



“La web debe estar disponible funcionando correctamente la gran mayoría del tiempo”

REQUISITOS NO FUNCIONALES

También llamados requisitos de calidad. **Deben ser medibles.**



“La web debe estar disponible funcionando correctamente la gran mayoría del tiempo”



“La web tiene que responder rápido a los usuarios”

REQUISITOS NO FUNCIONALES

También llamados requisitos de calidad. **Deben ser medibles.**



“La web debe estar disponible funcionando correctamente la gran mayoría del tiempo”



“La web tiene que responder rápido a los usuarios”



“La web debe responder en un tiempo inferior a los 300ms el 90% de las peticiones e inferior a 1s el 99% de las peticiones.”

RESTRICCIONES

- **Técnicas.**
 - Utilizar un lenguaje o framework específico.
 - APIs o servicios externos.
- **Legales.**
 - Transacciones en el sector bancario.
 - GDPR en la Unión Europea.
- **De Negocio.**
 - Presupuesto.
 - Tiempo.
 - Procesos internos.

IMPORTANCIA DE ELEGIR UNA BUENA ARQUITECTURA

- Posibilidad de **escalar nuestro sistema** de unos pocos usuarios a **millones de usuarios concurrentes**.
- **Facilidad de incluir nuevas funcionalidades** que a su vez escalen bien.
- Si no elegimos una buena arquitectura, **podemos perder meses de trabajo**.
- **Rediseñar una arquitectura es un trabajo muy costoso**, especialmente en sistemas a gran escala. Supondrá mucho **tiempo y dinero** a la empresa.

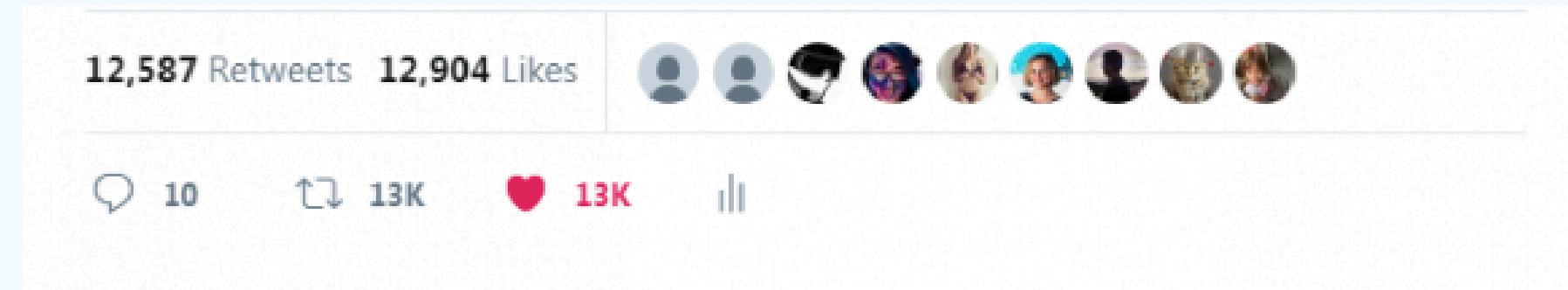
IMPORTANCIA DEL CONTEXTO



Plataforma aparentemente "sencilla":

- Publicar pequeños textos (tweets).
 - Pueden incluir imagen o vídeo.
- Acciones sobre los tweets.
 - Like.
 - Retweet / Repost.
 - Comentar.
- Posibilidad de seguir a otros usuarios.

IMPORTANCIA DEL CONTEXTO



Escalar un sistema a millones de usuarios concurrentes no es sencillo e **implica solventar muchos problemas.**

CONTENIDO DEL CURSO

- **Introducción.**
 - Rendimiento.
 - Escalabilidad.
 - Fiabilidad.
 - Mantenibilidad.
- **Conceptos clave en el diseño de sistemas a gran escala.**
 - Load Balancer.
 - API Gateway.
 - Message brokers.
 - Cache.
 - CDN (Content Delivery Network).
 - Data Centers.

CONTENIDO DEL CURSO

- **Almacenamiento de datos.**

- Bases de datos relacionales vs NoSQL.
- Índices y desnormalización de BBDD.
- Replicación de datos.
- Particionado de Base de Datos (Sharding).
- Teorema CAP.

- **Patrones de arquitectura de software.**

- Arquitecturas multi-layer y multi-tier.
- Microservicios.
- CQRS y Event Sourcing.

CONTENIDO DEL CURSO

- **Ejercicios prácticos.**
 - URL Shortener.
 - Sistema de Chat.
 - E-Commerce.
 - Twitter.