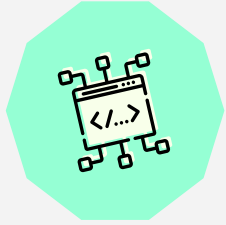


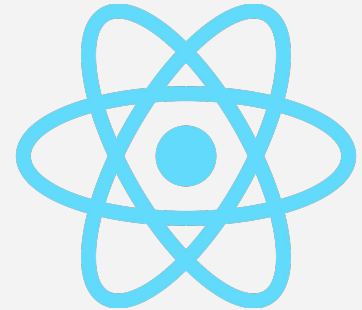
Resultado de la investigación

Grupo 4

- Patrón orientado a eventos
- Framework: Spring Boot
- Front-End: React
- Lenguaje: Java
- Base de Datos: MongoDB

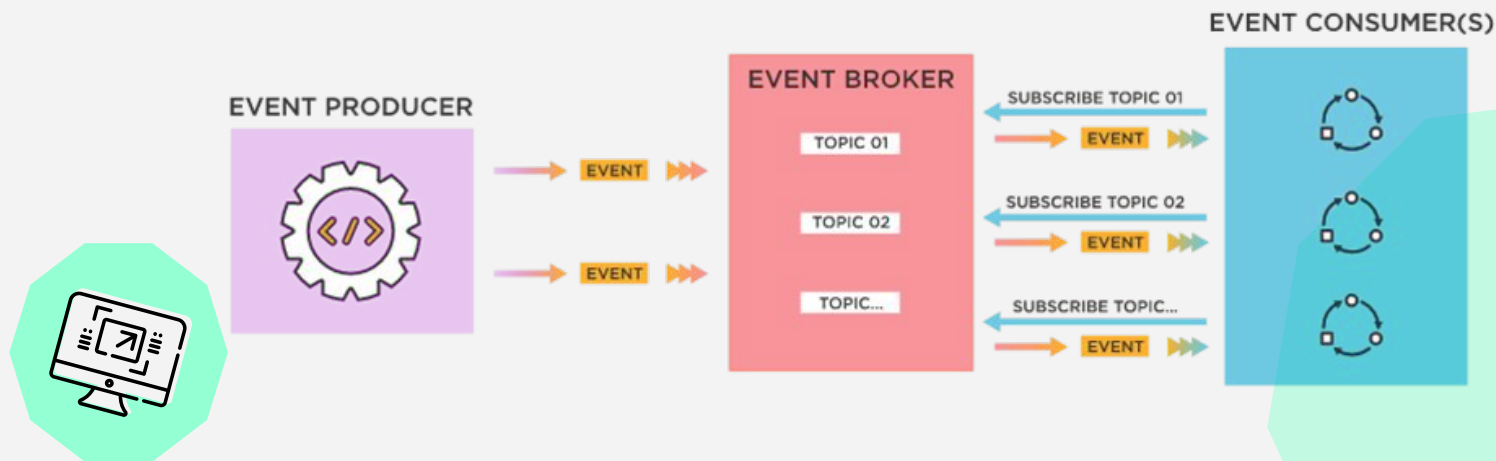


— Definición, Historia y Evolución

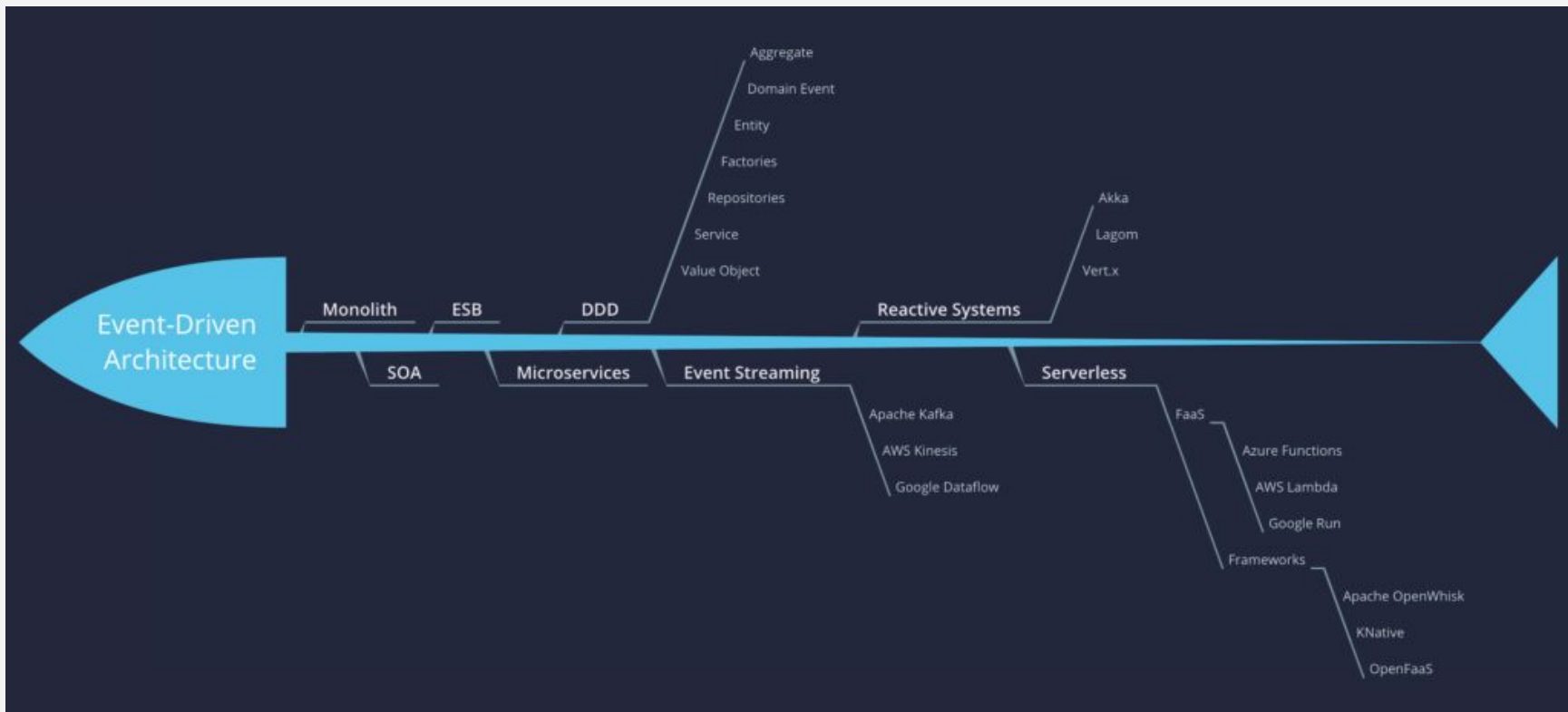
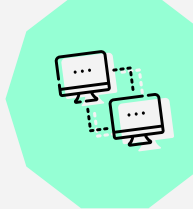


Patrón Orientado a eventos

Es un patrón arquitectónico en el que los componentes de software se comunican a través de eventos.



Evolución y Historia

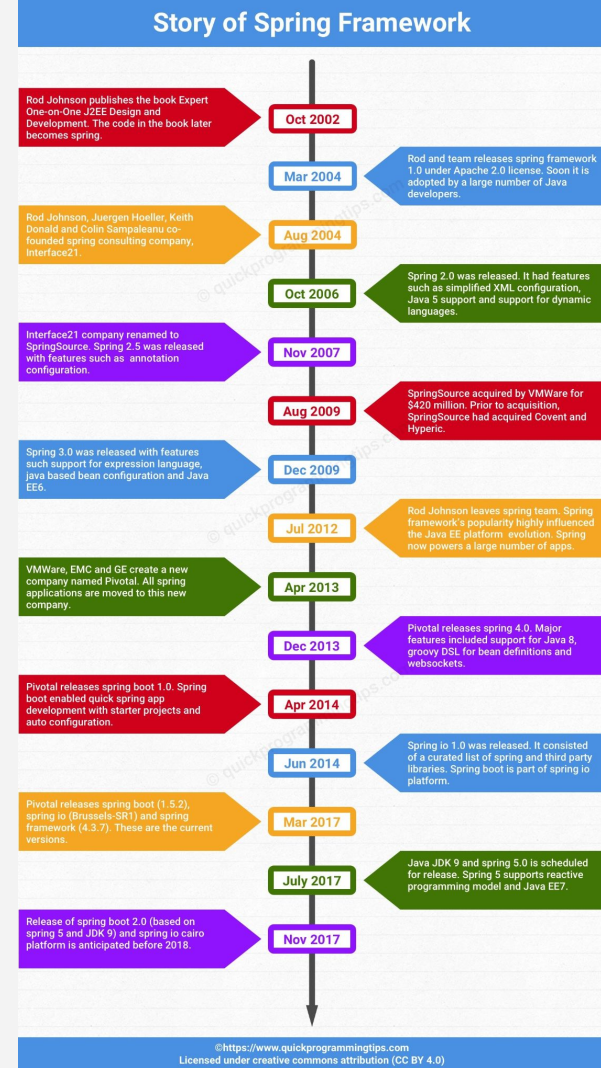


Spring Boot

Spring Boot es un framework de código abierto basado en Java que se utiliza para crear aplicaciones independientes basadas en Spring de nivel de producción con una configuración mínima.

Algunas características:

- Configuración automática de Spring.
- Servidores integrados para aplicaciones web.
- Gestión simplificada de dependencias
- Interfaz de línea de comandos para generar nuevos proyectos y ejecutar aplicaciones.

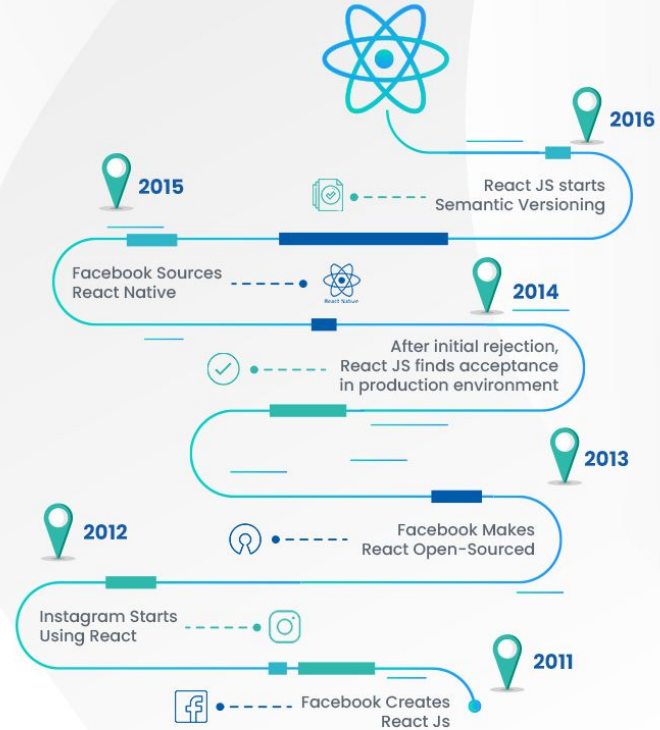


React

React es una biblioteca JavaScript de código abierto para crear interfaces de usuario. Fue desarrollada por Facebook y publicada por primera vez en 2013.

React permite a los desarrolladores crear interfaces de usuario dinámicas e interactivas mediante una sintaxis declarativa y una arquitectura basada en componentes.

THE GRAPHICAL GROWTH OF REACT JS

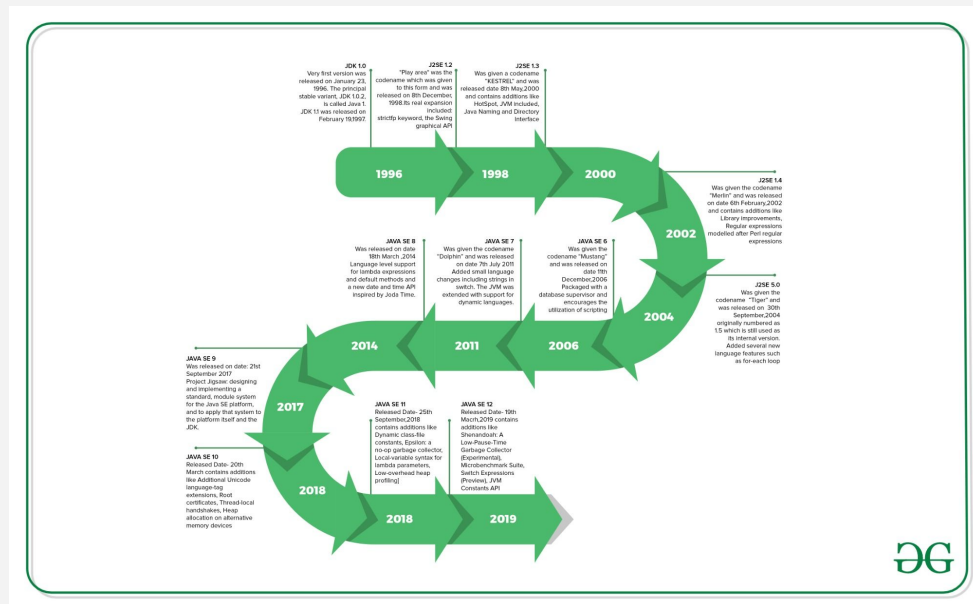


Java

Java es un lenguaje de programación de propósito general diseñado para ser rápido, seguro e independiente de la plataforma

Algunas características:

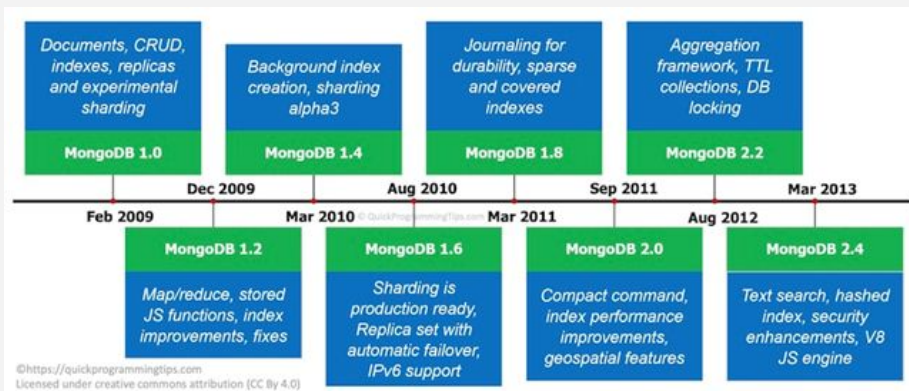
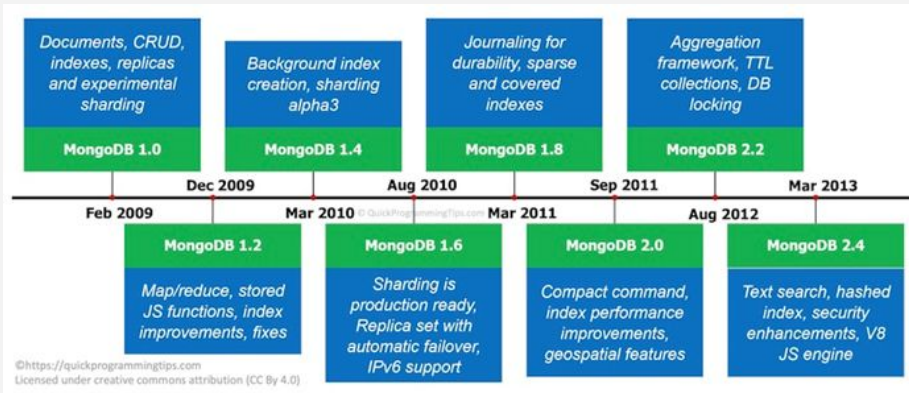
- Gestión automática de la memoria
- Gestión de excepciones
- Multihilo
- Funciones de seguridad
- Compatibilidad con servicios web y redes



MongoDB

MongoDB es una base de datos NoSQL orientada a documentos que almacena datos en un formato similar a JSON.

Es una base de datos de código abierto, escalable y de alto rendimiento, diseñada para manejar grandes volúmenes de datos no estructurados o semiestructurados.



— Relación entre los temas



01 SpringBoot

Desarrollar aplicaciones web y servicios RESTful

03 Java

Lenguaje de programación popular utilizado para desarrollar aplicaciones empresariales

02 React

Construir interfaces de usuario interactivas y de alta calidad

04 MongoDB

MongoDB es una base de datos NoSQL que se utiliza en aplicaciones modernas

— Relación entre los temas



Patrón de arquitectura (EDA) y las tecnologías

Se utilizan comúnmente juntos en el desarrollo de aplicaciones modernas.



Es una forma de

diseñar sistemas que se basa en el intercambio de eventos entre diferentes componentes del sistema



SpringBoot y Java

Proporcionan una plataforma sólida para implementar una arquitectura orientada a eventos.



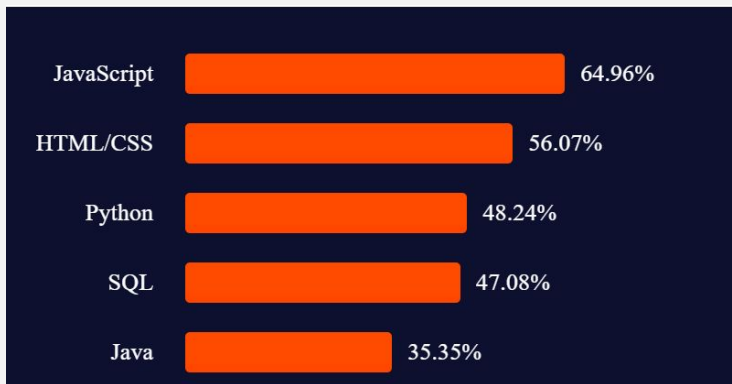
EDA y las tecnologías

Utilizan comúnmente juntas para construir aplicaciones escalables, flexibles y modulares.

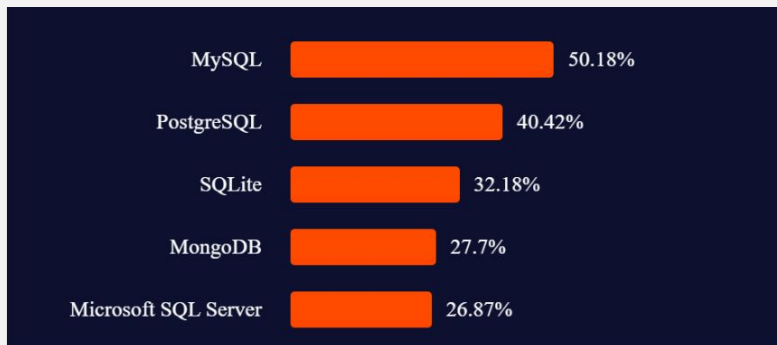


Encuestas de Stack OverFlow 2021

Tecnologías

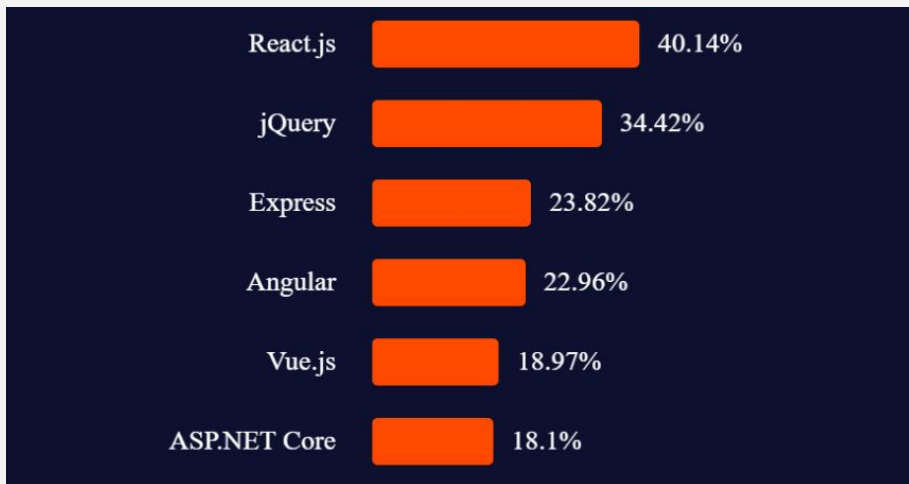


Bases de datos



Encuestas de Stack OverFlow 2021

Frameworks
Web



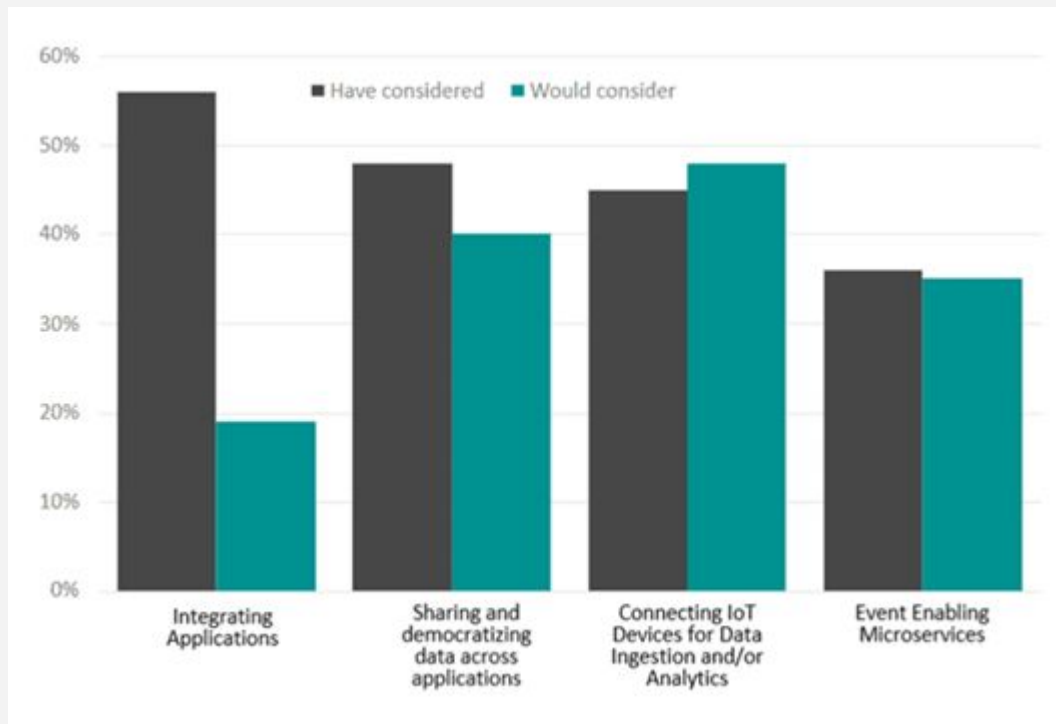


Situaciones donde se — aplican

Casos de uso - Arquitectura orientada a eventos

Los 4 principales casos de uso de la arquitectura basada en eventos según una encuesta realizada en 2021 fueron:

- Integración de aplicaciones.
- Compartir y democratizar los datos entre aplicaciones.
- Conexión de dispositivos IoT para el análisis de datos.
- Microservicios habilitados para eventos.



— Aplicaciones del stack



Aplicaciones de Internet de las Cosas



Arquitectura de microservicios



Aplicaciones web escalables y en tiempo real que requieren un procesamiento de datos basado en eventos.





Ventajas

Orientado a eventos

Arquitectura Orientada a Eventos



Arquitectura

dirigida por eventos es
asíncrona sin bloqueo.

Los servicios

no necesitan saber o
depender de otros servicios

Más fácil

buscar cuellos de botella en
un servicio específico y
escalar dicho servicio.

Servicios se pueden

actualizar, testear y
desplegar
independientemente

Adquiere

rendimiento muy alto a
través de sus capacidades
asíncronas

Escalabilidad

se logra de forma natural en este
patrón mediante procesadores de
eventos altamente independientes y
desacoplados.

Spring Boot



Gratis

creado para simplificar el desarrollo de aplicaciones con Spring Framework bajo licencia Apache 2.

Permite

Hacer Webs y microservicios

Gran ecosistema

documentación bien estructurada y completa

Funciona sobre la JVM:

permite desarrollar nuevas aplicaciones, migrar o reutilizar código

Rápido

desarrollar de inmediato y desplegar la aplicación sin complicaciones

Políglota

gracias al uso de la JVM

React



DOM virtual

poder generar el DOM de forma dinámica

Componentes

aplicaciones más escalables y fáciles de mantener.

Amplia comunidad

gran cantidad de paquetes y librerías externas

Isoformismo

crear código que se ejecuta tanto en contextos de servidor como de cliente

Flujo de datos unidireccional

datos solo se pueden modificar en una dirección.

Aprendizaje

Como está escrito en JavaScript la curva de aprendizaje es bastante rápida

Java



Fácil de aprender

es fácil de escribir, compilar, depurar y aprender

Independiente de

La plataforma. Tiene la capacidad de moverse fácilmente de un sistema informático a otro

Capacidades

De plataforma cruzada

Orientado a objetos

crear programas modulares y código reutilizable.

Robusto

Funciones de

seguridad

MongoDB



Ideal para

Pocos recursos de computación.

Documentación

Posee una documentación muy buena, muy amplia y detallada

Coste Bajo

lo único que se paga es por el soporte, en caso de necesitarlo.

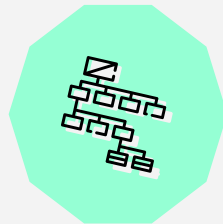
Complemento

Perfecto para JavaScript, se podrá utilizar toda la potencia de sus funciones y operadores en MongoDB.



Desventajas

— Desventajas



Arquitectura orientada a eventos

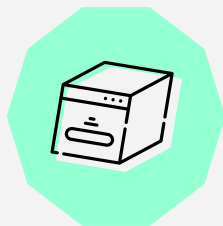
- Posibilidad de desborde
- Potencial imprevisión de escalabilidad
- Pobre comprensibilidad
- No hay garantía del lado del publicador, que el suscriptor responderá al evento
- No hay mucho soporte de recuperación en caso de falla parcial

SpringBoot

- Se actualiza muy constantemente.
- Para aplicaciones escalables el catálogo de wicket podría quedarse corto.
- Entre más componentes más desorganizado y complejo

React

- Falta de documentación oficial.
- No hay un patrón de desarrollo.
- Se necesita mucho tiempo para dominarlo



Java

- Rendimiento más lento
- Puede requerir más memoria que otros lenguajes.
- La programación en Java tiene sobrecarga de código



MongoDB

- No es una base de datos adecuada para aplicaciones con transacciones complejas.
- Es una tecnología joven
- No tiene Joins para consultas

Caso de estudio

Se implementó un programa sencillo de una transferencia bancaria donde la página web tiene tres secciones:

- **Información de la cuenta:** se muestra la información de la cuenta que está haciendo la transferencia. Se incluyen datos como número de cuenta, nombre del titular, fecha de nacimiento del titular, fecha de apertura de la cuenta, tipo de cuenta, tipo de moneda y la cantidad existente en la cuenta.
- **Formulario de transferencia:** se muestra un pequeño formulario donde se pide el número de cuenta al cual transferir y la cantidad que se quiere transferir.
- **Lista de transferencias:** se muestra el historial de transferencias de la cuenta. Cada transferencia muestra su id correspondiente, el número de cuenta al que se le envió el monto, la cantidad de monto y el día en que se realizó la transferencia.

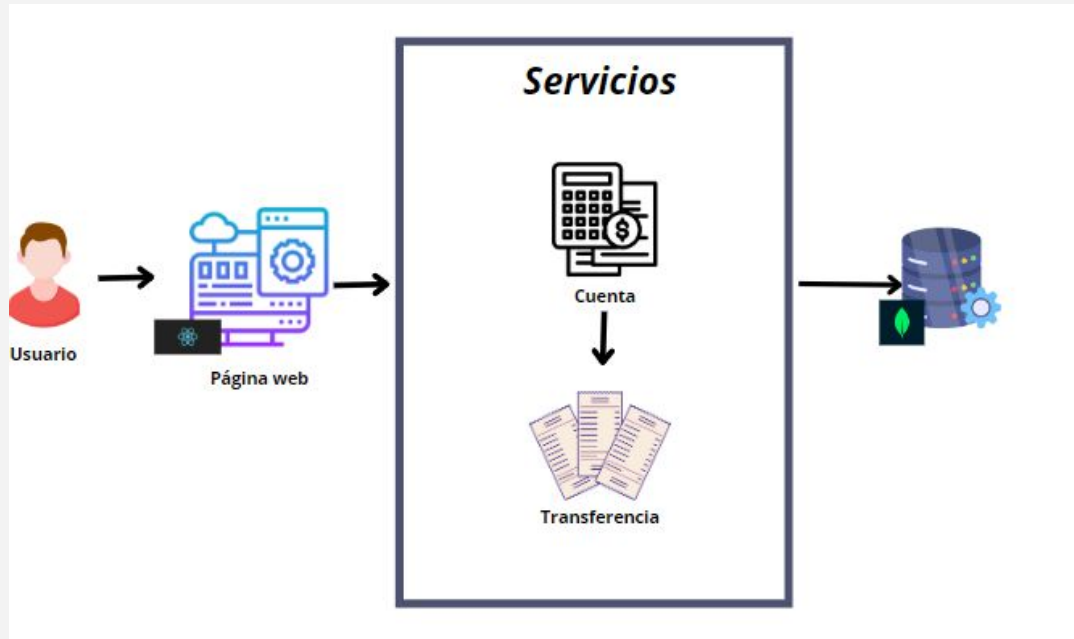




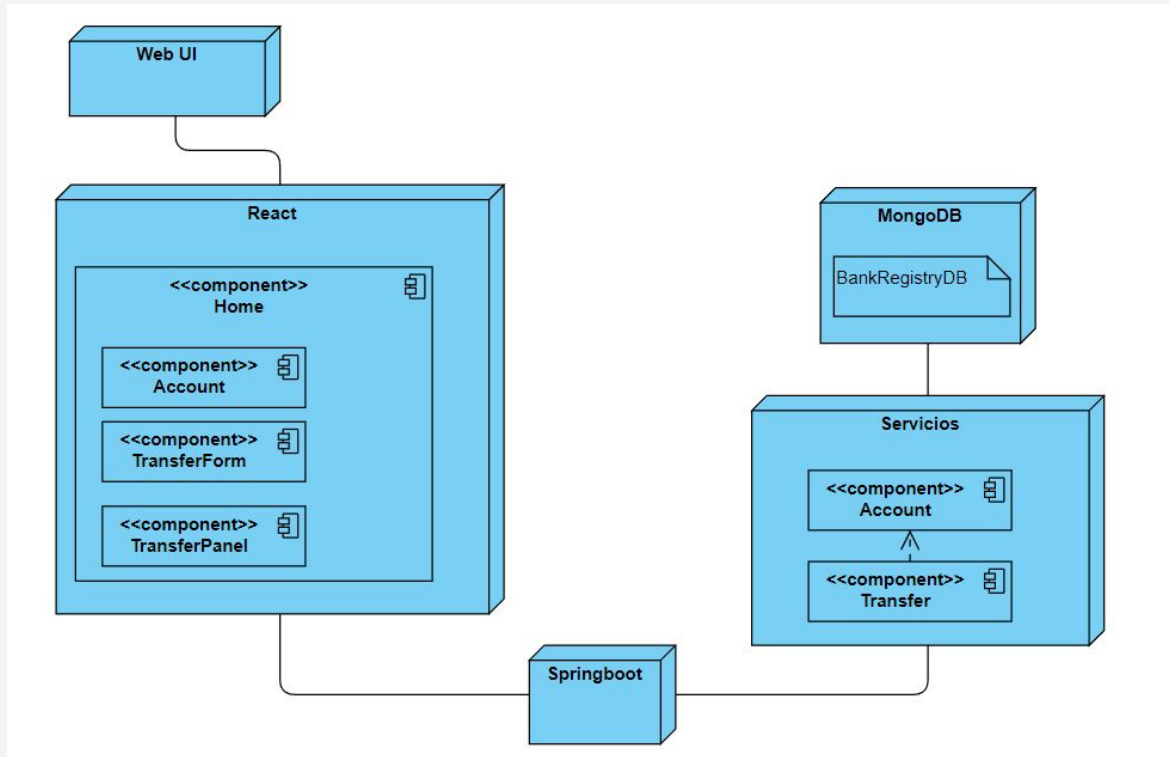
Representación

— *uml*

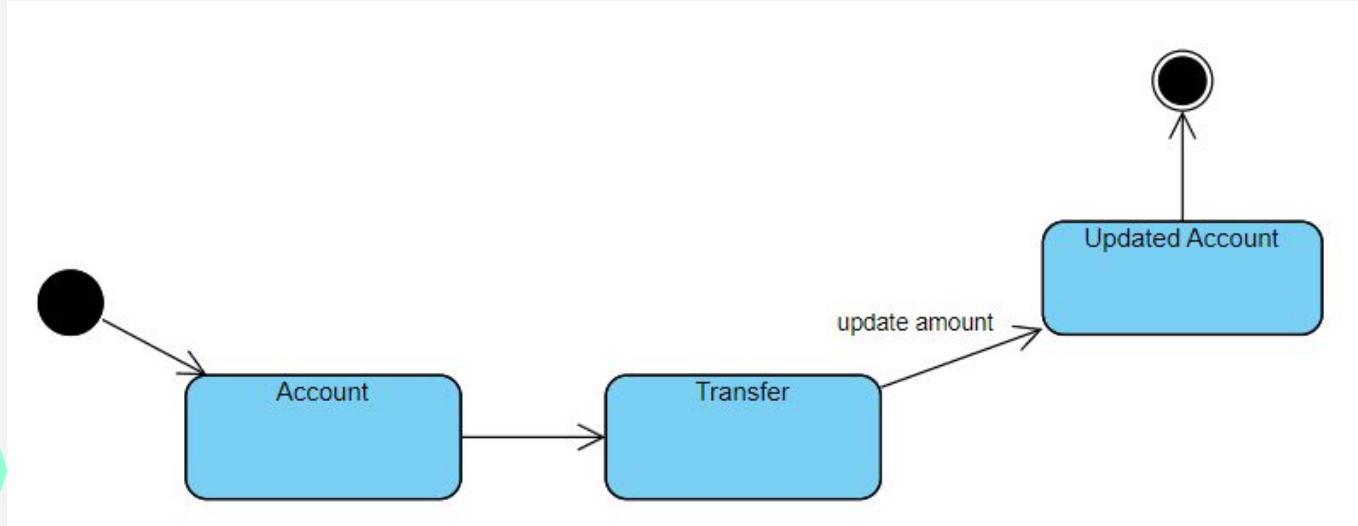
Arquitectura de alto nivel



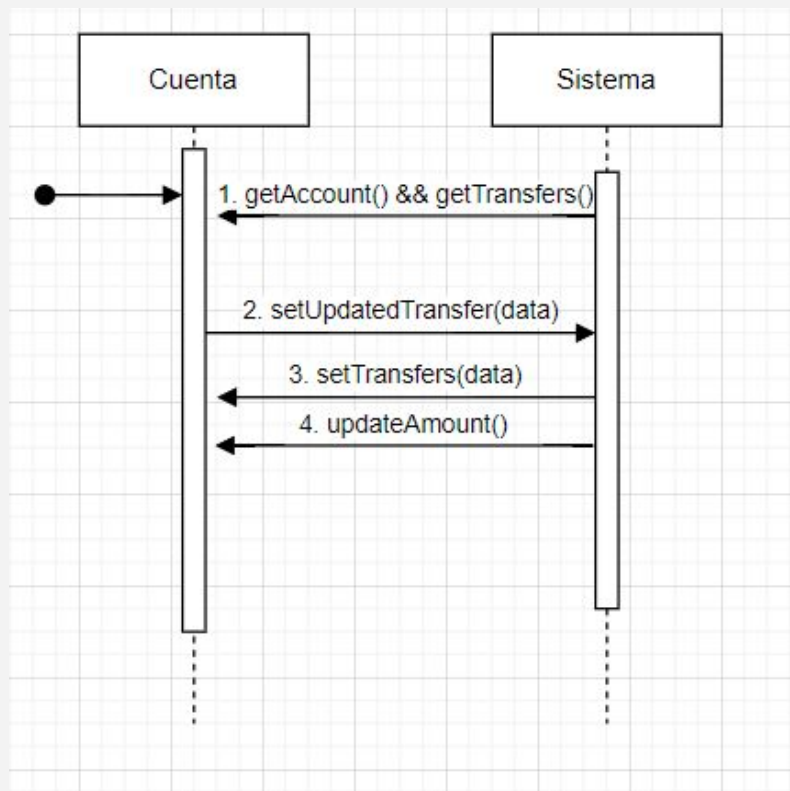
Estructura



Comportamiento



Interacción





Principios SOLID Asociados

Responsabilidad única

Cada componente tiene una única responsabilidad.

Open-Closed

La arquitectura de eventos podría permitir la adición de nuevos componentes sin alterar el funcionamiento existente

Segregación de interfaces

En el programa se han utilizado interfaces para definir los servicios ofrecidos por el backend. Los componentes que utilizan los servicios solo necesitan conocer la interfaz relevante y no la implementación completa.

Atributos de calidad



Escalabilidad

Escalabilidad horizontal y vertical del sistema

Confiabilidad

Lo permite el uso de una base de datos confiable y la arquitectura "Event Driven"

Rendimiento

MongoDB permite una mayor velocidad de acceso y procesamiento de datos

Mantenibilidad

"Event Driven" y el diseño basado en principios SOLID permiten una mayor facilidad de mantenimiento y actualización del sistema.

Usabilidad

React en el front end permite una mayor usabilidad y facilidad de uso para los usuarios finales.

Ejemplo práctico y funcional

<https://github.com/MartinChafloque/BankRegistry/releases/tag/v1.0>



— Conclusiones

- Es una buena elección a la hora de desarrollar una aplicación escalable y de alta calidad que maneja grandes cantidades de datos no estructurados.
- El stack presentado va ligado a más que todo los siguientes requisitos: escalabilidad, interacción con múltiples componentes, requerimientos empresariales, interfaz de usuario interactiva y el manejo de datos no estructurados.



Lecciones aprendidas



- Desacoplamiento de componentes: La arquitectura de bus de eventos promueve la separación de las responsabilidades entre componentes, lo que facilita el mantenimiento y mejora la escalabilidad del sistema. Aprenderás a diseñar componentes que se comuniquen a través de eventos, reduciendo la dependencia directa entre ellos.
- Desarrollo de aplicaciones web modernas: Con React, aprenderás a crear aplicaciones web de una sola página (SPA) utilizando componentes modulares y la arquitectura Flux. React es una biblioteca popular para el desarrollo de interfaces de usuario eficientes y dinámicas.
- Integración de tecnologías y lenguajes: La combinación de Java (para la lógica del servidor y los microservicios), Spring Boot (para facilitar el desarrollo de aplicaciones Java), React (para las interfaces de usuario) y MongoDB (para la gestión de datos) te enseñará cómo integrar diferentes tecnologías y lenguajes de programación en un proyecto. Esto te dará una comprensión más profunda de cómo los diferentes componentes pueden trabajar juntos en una aplicación moderna y escalable.
- Patrones de diseño y buenas prácticas: Al trabajar con estas tecnologías, estarás expuesto a diversos patrones de diseño y buenas prácticas en la industria del desarrollo de software. Aprenderás a implementar patrones de diseño adecuados para resolver problemas comunes y a aplicar buenas prácticas de desarrollo, como la programación orientada a objetos, la inyección de dependencias y la separación de responsabilidades.

Thanks! —



CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik

