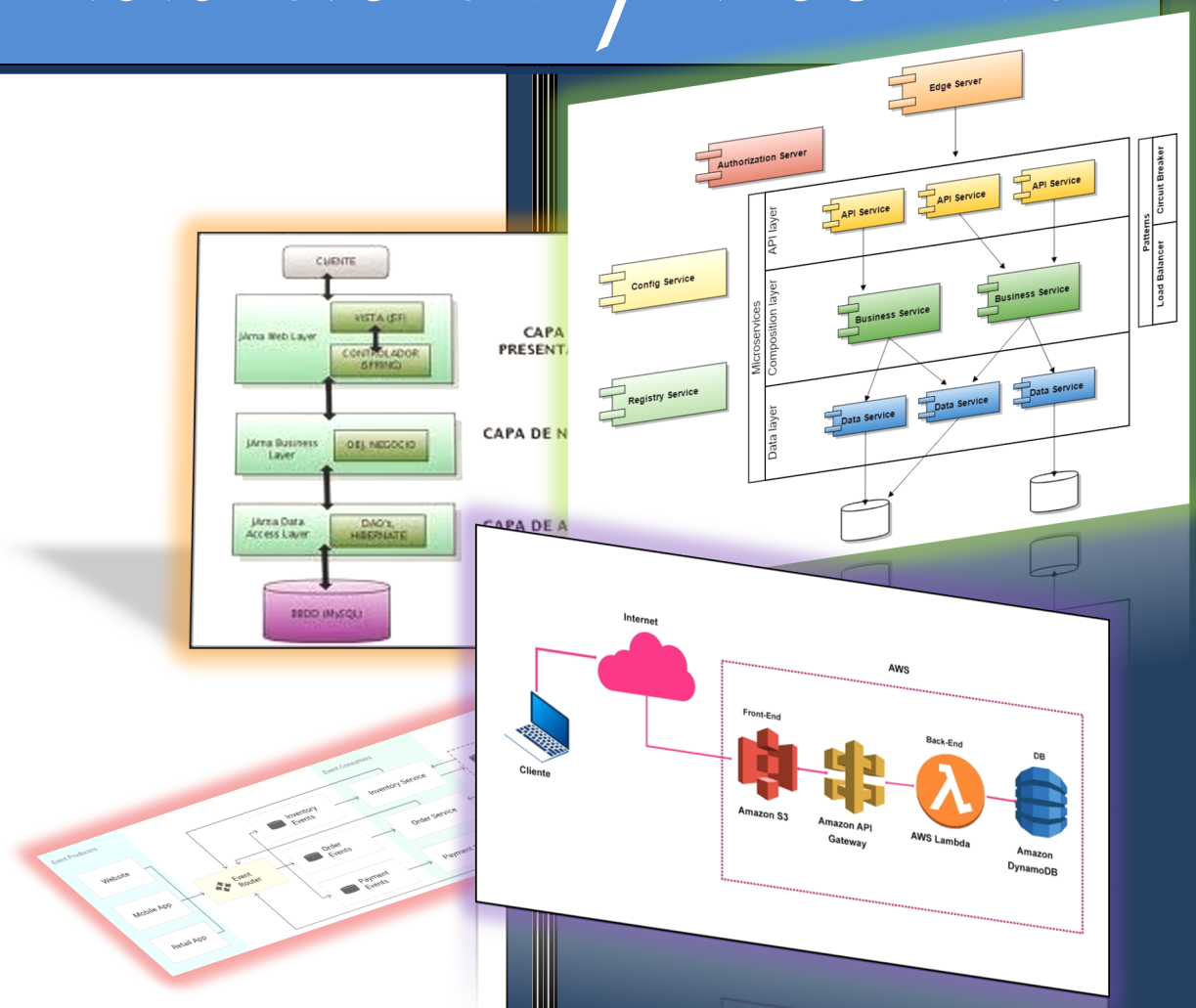
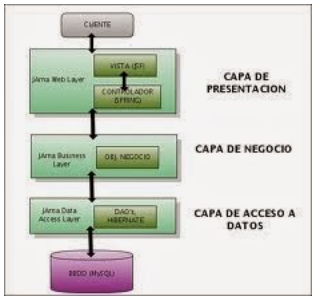
Daniel Navas Borjas
2º DAW
2024

Análisis de arquitectura



GlobalStay

Para el proyecto ficticio de "GlobalStay" he decidido usar la arquitectura "multicapa".



Características:

Está estructurado en 3 capas (presentación, lógica de negocio, datos), esta arquitectura también se le puede llamar monolítica y he estado indagando en internet y todo está en un mismo servidos.

Ventajas e inconvenientes:

Una de las ventajas obvias de esta arquitectura esque nos permite poder tener nuestra base de datos en un servidor centralizado, pero podemos desplegarlos en multiples servidores tambiíen gracias a esto obtenemos escalabilidad (Que es necesario porque dicha empresa es a nivel mundial y puede abrir otro hotel en cualquier momento). Gracias a la tecnología multicapa podemos reutilizar componentes en cada servidor.

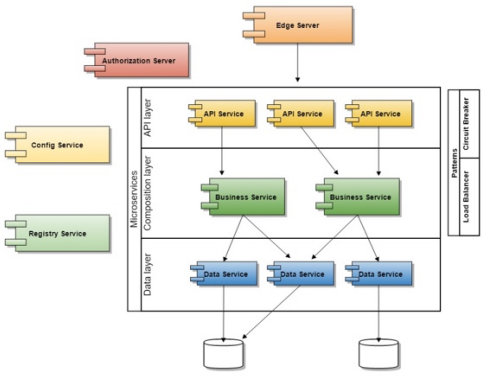
Como inconvenientes decir que tiene un mayor costo de infraestructura (Pero esta empresa es a nivel mundial por lo que se lo pueden permitir), es más complejo, dificultad de depuración y latencia adicional.

Herramientas para desplegar:

- Despliegue en multiples servidores.
- Uso de contenedores.
- En ocasión combinado con microservicios.

Streamlt

Para el proyecto ficticio de "Streamlt" he decidido usar la arquitectura "arquitectura de microservicios".



Características:

Dividen la aplicación en pequeños servicios independientes, y se despliegan en entornos de la nube, permitiendo mayor flexibilidad y escalabilidad.

Ventajas e inconvenientes:

Debemos usar esta arquitectura cuando tengamos actualizaciones y despliegues frecuentes por lo que al tratarse de una plataforman de streaming de video será casi constante.

Alguna de las ventajas de esta arquitectura es que la falla de un microservicio no afecta a otro y esto es esencial en una plataforma de streaming, permite utilizar diferentes tecnologías por lo que no tendremos que anclarnos a una solo, tiene escalabilidad independiente.

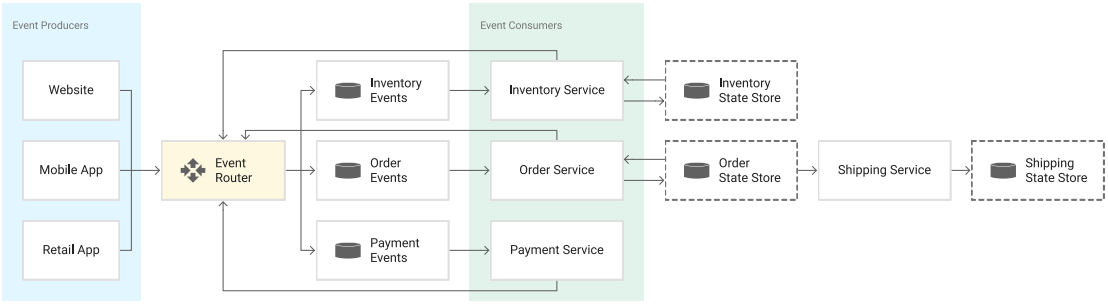
Alguna de los inconvenientes es que es complejo a la hora de gestionarlo, sobrecarga en la gestion y monitorización de multiples y tiene costos potenciales altos.

Herramientas para desplegar:

- Uso de contenedores con Docker.
- Orquestación con kubernetes.
- Integración y Entrega contínua [CI/CD] con Jenkins.
- Servicio de monitorización con ELK Stack o Prometheus.

SmartRecipes

Para el proyecto ficticio de "SmartRecipes" he decidido usar la arquitectura "arquitectura orientada a eventos".



Características:

Los eventos, o cambios en el estado de un sistema, son los que impulsan la comunicación entre componentes de una aplicación. Esto nos permite tener desacoplamiento, reactividad, y flexibilidad.

Ventajas e inconvenientes:

Esta empresa ve indispensable que la generación de recetas basadas en IA deben ejecutarse solamente cuando sea necesario, por lo que he escogido esta arquitectura porque es un sistema que requieren procesamiento en tiempo real.

Alguna de las ventajas es que es altamente escalable, alta eficiencia para manejar eventos en tiempo real (Especificamente para analizar los alimentos que tiene y hacer recetas en tiempo real) y es ideal para aplicaciones distribuidas y sistemas con microservicios.

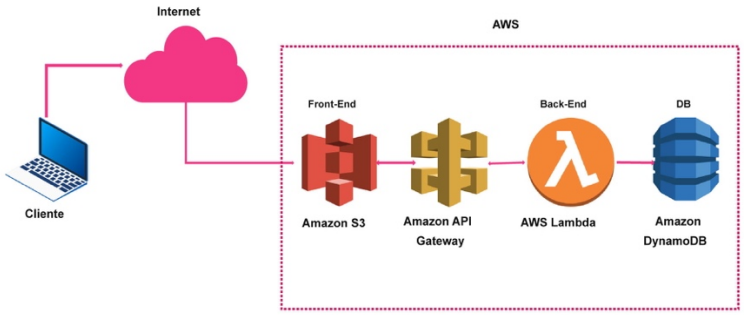
Alguno de los inconvenientes es que es complejo en la gestión y monitoreo de eventos, tiene dificultad a la hora de depuración y trazabilidad de eventos, puede generar latencia si los eventos no se manejan de forma adecuada.

Herramientas para desplegar:

- Despliegue en plataformas en la nube como AWS Lambda, Azure Functions o Google Cloud Functions.
- Uso de servicios de mensajería como Kafka, RabbitMQ o AWS SQS para gestionar colas de eventos.
- Integración continua con herramientas como Kubernetes para gestionar microservicios que procesan eventos.
- Uso de sistemas de monitoreo como ELK Stack o Prometheus para rastrear eventos.

CasinoTech

Para el proyecto ficticio de "CasinoTech" he decidido usar la arquitectura "arquitectura serverles".



Características:

Permite a los desarrolladores escribir y desplegar código sin preocuparse por la infraestructura. Los proveedores de servicios cloud (como AWS Lambda, Azure Functions, o Google Cloud Functions) gestionan automáticamente la infraestructura, escalando la aplicación en función de la demanda.

Ventajas e inconvenientes:

He escogido esta arquitectura ya que es de vital importancia que la seguridad monitorice en tiempo real el casino y dicha arquitectura se puede usar cuando haya que hacer procesamiento en tiempo real.

Alguna de las ventajas es que los servicios se escalan según la demanda, costos basados en el uso (Cuanto más uses más costo gener y biceversa), desarrollo ágil y reducción de la carfa operacional.

Alguno de los inconvenientes es que tiene limitaciones de tiempo de ejecución, hay dependencia del proveedor, el tiempo de arranque de funciones que no se han ejecutado es más lento y suele haber problemas con el manejo del estado.

Herramientas para desplegar:

- Despliegue a través de servicios de plataformas en la nube como AWS Lambda, Azure Functions, Google Cloud Functions.
- Gestión de funciones y eventos mediante framework específicos como Serverless framework.
- Despliegue en combinación con otros servicios en la nube, como bases de datos gestionadas y servicios de cola de mensajes.
- Monitorización mediante AWS CloudWatch o Azure Monitor.