

LEVANTAR VUELO

TP Grupal - IASC 1C2023

CONTEXTO

DOMINIO

REQUERIMIENTOS

NO REQUERIMIENTOS

LAS TECNOLOGÍAS

FORMA DE ENTREGA Y EVALUACIÓN

CONTEXTO

Se necesita diseñar e implementar una aplicación que permita **gestionar la compra de pasajes de diferentes aerolíneas** en tiempo real y sincronizada entre múltiples usuarios.

La empresa *Levantar vuelo* se dedica a la reserva y compra de pasajes de vuelos a través de una plataforma en línea. Esta empresa ofrece a los clientes la posibilidad de comparar precios, horarios y opciones de vuelo de diferentes aerolíneas para encontrar la mejor opción para sus necesidades de viaje por un tiempo limitado.

Cuando decimos que la aplicación debe funcionar en tiempo real nos referimos a que todas las operaciones que realiza un usuario deberían ocurrir de forma rápida (al menos de forma perceptible para el usuario que las está llevando a cabo), y no depender sustancialmente de las condiciones de la red ni de la cantidad de personas interesadas en reservar ese pasaje.

DOMINIO

La plataforma que nos pide la empresa Levantar vuelo maneja ofertas de lugares en vuelos, donde las aerolíneas publican parte de sus asientos disponibles. Así mismo, al ser una oferta es por tiempo limitado por lo que una vez que los usuarios interesados reciben la alerta de la publicación de un vuelo que coincide con su búsqueda, miles de usuarios tratan de reservar y elegir el asiento así compran el vuelo a precios increíbles.

Tengan en cuenta que si el usuario reserva el vuelo es porque ya se hizo reserva del monto en su medio de pago, es decir, no tenemos que preocuparnos de que no pague y mismo si se cancela el sistema ya sabe como devolver el dinero. No es parte de la solución tener en cuenta esto.

¿Qué nos permite hacer nuestro sistema?

- Las aerolíneas pueden publicar vuelos.
- Los interesados pueden crear alertas según su búsqueda.

- El sistema notifica según las alertas creadas.
- Los interesados/compradores reservan vuelos.
- Una vez reservado deben elegir el tipo de asiento para así poder finalizar la compra. Sin asiento la reserva no se concreta, por lo que podría llegar a cancelarse si finaliza el tiempo de oferta o si todos los asientos son vendidos.
- Los vuelos tienen tiempo límite de oferta, por lo cual si las reservas al finalizar el tiempo de venta no se concretaron se cancelan y se cierra el vuelo ya que pasan a que los asientos se vendan por otras plataformas a mayor precio.
- Los vuelos también se cierran una vez que todos sus asientos quedaron vendidos.
- Siempre que un vuelo es cerrado, se les notifica a todas las personas que reservaron el vuelo y se cancelan sus reservas que no terminaron de concretarse.

Notificación de vuelo interesante

- Los interesados generan alertas de interés de vuelos con la siguiente información:
 - Fecha y/o más de interés.
 - Lugar de origen.
 - Lugar de destino.
- Una aerolínea publica un nuevo vuelo en el sistema
 - La información que la aerolínea debe proporcionar al sistema es:
 - Tipo de avión, por ejemplo, Boeing 737, Airbus 320, Embraer 190.
 - Cantidad de asientos por tipo (ventana, pasillo y/o entre asientos).
 - Fecha y horario del vuelo.
 - Lugar de origen.
 - Lugar de destino.
 - Tiempo de oferta.
- El sistema alerta a todos los interesados que coinciden sus criterios de búsqueda con los de la aerolínea pública.

Usuario interesado en un vuelo - Caso base

Una vez transcurrido el escenario 1:

- El interesado reserva el vuelo
- Selecciona los asientos:
 - Ubicación de asiento: es ventana, pasillo o cualquier opción, con posibilidad a que toque asientos entre dos asientos.
 - Cada asiento puede ser de un tipo diferente de elección: 2 ventanas + 1 pasillo.
 - Cuando son múltiples asientos, podemos pedir que sean contiguos o no.
- La elección de asientos genera un pedido interno de asientos, el sistema valida que estén disponibles, sino vuelve a pedir que se seleccionen:
- Una vez que la elección de asientos está confirmada, queda confirmada la compra del pasaje.

Cancelación de reserva por completitud.

Nos puede pasar que muchos usuarios tengan configurada la misma alerta e ingresen todos al sistema, como nos dicen las métricas de negocio, no todos los interesados en una alerta compran, entonces siempre la aerolínea manda a todos los interesados, y permite reservar más de los que caben en el avión, ya que el momento de la compra solo sucede una vez definido el asiento. La empresa siempre quiere ganar, su objetivo es que los vuelos se vendan completos lo más rápido posible.

Ejemplo, una vez transcurrido el escenario 1, el vuelo X tiene disponibles 2 asientos donde solo 1 es ventana y el otro pasillo.

- El interesado A y el B reservan el vuelo X
- Sabemos que los usuarios A y B les interesa ventana,
- El usuario B selecciona un asiento con ventana, y compra su boleto.
- Llega el usuario C interesado en el vuelo X sin un gusto particular de asiento, puede reservar el vuelo X.
- El usuario A, demorando por elegir, trata de seleccionar el asiento de ventana, el sistema se le rechaza ya que B lo compró, por lo que se le invita a seleccionar otro asiento.
- El interesado C puede seleccionar su asiento, como no le interesa uno particular, lo hace rápidamente y compra su boleto.
- Al usuario A se le notifica que ese vuelo ya quedó cerrado por lo cual se le cancela la reserva.

REQUERIMIENTOS

Para la prueba de concepto de nuestra arquitectura se nos han presentado ciertos requerimientos para asegurar su eficiencia y eficacia.

Almacenamiento en memoria

La operatoria del servicio de reserva/compra de vuelos debe **ocurrir en memoria**, evitando en todo momento la persistencia de información en disco, a fin de garantizar un desempeño óptimo y cumplir con las exigencias legales de privacidad y seguridad de los datos que nos impone la empresa.

Escalabilidad y monitoreo

La aplicación de reservas experimenta períodos de alta demanda durante las vacaciones, lo que resulta en **ráfagas de tráfico intensas**. Debido a que las ofertas son cortas en el tiempo, contamos con aumento significativo en la carga de trabajo, por lo cual requiere que el sistema pueda escalar en nodos, capaz de manejar estas cargas y luego reducir la cantidad cuando finalicen dichos períodos.

Para poder reconocer cuándo ocurren estos picos de demanda, es necesario contar con algún tipo de monitoreo que permita identificarlos.

La forma de agregar/quitar nodos puede ser manual o automática (esta última opción la consideramos un **bonus**).

Redundancia de datos y tolerancia a fallos

Es mandatorio que **el estado esté distribuido** entre los distintos nodos, ya que si tenemos alguna falla en nuestro sistemas tenemos que poder resolverla en el menor tiempo posible. Es decir, **si alguno de los nodos de nuestra aplicación se cae debemos asegurarnos poder restaurarlo** a partir de los restantes (sin perder información sobre las reservas).

Despliegue

Si bien *Levantar vuelo* considera **deseable** que el despliegue se haga mediante contenedores Docker, este punto es opcional.

NO REQUERIMIENTOS

Interfaz de usuario

La interfaz para publicar vuelos, crear alertas y reservar vuelos puede ser la más sencilla que permita cumplir con los requerimientos del TP. Por ejemplo, una API REST es más que suficiente para nuestros fines. Las notificaciones (ante alertas de vuelos y ante reservas finalizadas/cancelaciones) quedan a criterio del grupo.

Consideraciones de seguridad

En esta primera etapa no se tendrán en cuenta cuestiones de seguridad; se asumirá que todos los clientes y servidores están dentro de una red segura.

LAS TECNOLOGÍAS

Se podrá utilizar cualquier tecnología que aplique alguno de los siguientes conceptos vistos en la cursada:

- Paso de mensajes basado en actores
- Corrutinas
- Continuaciones explícitas (CPS)
- Promises

Obviamente, lo más simple es basarse en Elixir/OTP, Ruby, Python, o Node.js, que son las tecnologías principales que vimos en la materia.

Otras opciones son tecnologías basadas en Scala/Akka, Go, Clojure y Rust, pero en estos casos podremos dar menor soporte.

Tengan en cuenta que hay tecnologías y/o herramientas que pueden serles de utilidad a la hora de encarar el desarrollo del trabajo práctico, aunque ninguna es imprescindible para cumplir con los requerimientos del TP:

- Push notifications
- Websockets
- Server-sent events

Por cualquier duda en la elección de la tecnología pueden ponerse en contacto con el tutor de su grupo.

FORMA DE ENTREGA Y EVALUACIÓN

Se deberá construir el sistema descrito, tanto el servidor como clientes de prueba. No es obligatoria la construcción de casos de prueba automatizados, pero constituye un **gran plus**.

Se evaluará que:

- El sistema cumpla con los requerimientos planteados.
- Haga un uso adecuado de la tecnología y los conceptos explicados en la materia.
- La arquitectura y el estado de la aplicación sean distribuidos.
- La forma en que justifica la solución: tanto del modelo de concurrencia utilizado, aspectos de CAP que le dieron prioridad, cómo la manipulación de los datos, la recuperación de nodos ante fallos, entre otros aspectos.

Es recomendable para la entrega, tener documentación para poder comunicar las decisiones de la solución, como por ejemplo, la **arquitectura propuesta**, mencionando los **componentes** y la **comunicación** entre los mismos. Consejo: un diagrama puede facilitar la comprensión de la arquitectura propuesta.