

Proyecto



Curso de escalabilidad v2, día 6

Recapitulemos

Sistemas distribuidos. Escalado y replicación

Pruebas de carga. Optimización de rendimiento

Uso de métricas. Incógnitas desconocidas

Monitorización y observabilidad

Análisis de incidencias

Contadores

Uno de los problemas más difíciles
en programación distribuida

Mantiene una cuenta incremental

Requiere un bloqueo

Proyecto: Turnomátic distribuido

Nuestro turnomátic está saturado

Queremos ampliarlo para admitir más clientes

De paso, vamos a modernizarlo

Especificación técnica

El servicio recibe un identificador

Devuelve un contador unívoco y consecutivo

Cada llamada aumenta el contador en uno
y devuelve el total

Cada valor debe ser único

Cada cliente debe recibir un número consecutivo

API

URL: `http://localhost:7017/turno/:id`

Recibe un `:id` único: `grupoN`, donde N es tu grupo

Devuelve un JSON:

```
{  
  id: "grupoN",  
  turno: number  
}
```

Balanceo

Debes provisionar al menos dos máquinas

El tráfico debe estar balanceado en servidor

Sugerencias: ELB, balanceo por DNS

Para nota: autoescalado

Almacenamiento

Para almacenar los contadores puedes usar Redis

Sugerencia: utiliza el Redis disponible en ...

Monitorización

El sistema debe estar monitorizado

Elige el sistema que prefieras

Alertas: CPU excesiva, errores en el sistema

Para nota: instrumenta con [Honeycomb.io](https://honeycomb.io)

Pruebas de carga

Realiza pruebas de carga contra tu sistema

Obtén la carga máxima que puedes aceptar

Optimiza tu sistema

¡Enhorabuena!

