



Prueba técnica - Hazelcast

17 de Febrero, 2017



Índice

1. DESCRIPCIÓN PROBLEMA	3
2. ENTREGABLES	4
3. DETALLES ENTREGA	5

1. Descripción problema

Hay un select que tarda mucho en devolver los resultados (unos 3 segundos).

```
SELECT *
FROM USUARIOS
WHERE name LIKE '%patr%'
OR surname LIKE '%g%'
OR email LIKE '%o%'
OR phone LIKE '%69587%';
```

Esta select se usa para buscar usuarios registrados en una plataforma para determinar si es necesario o no registrarlo y mover el flujo de la aplicación según el resultado.

El resultado devuelve una lista de usuarios que en cierto modo se parecen al usuario principal que nos proporciona los datos.

Requisitos funcionales.

- Crear una caché de usuarios en hazelcast (<http://hazelcast.org/download/>) y rellenarla con información ficticia con 500k registros.

- La cache debe contener los siguientes campos:

Order	Column Title
1	Name
2	Email
3	Phone
4	Surname
Order	Column Title

- Crear un formulario web de registro con los mismos datos que la cache que conecte con hazelcast para preguntar si existen usuarios que se asemejen al actual.
 - Sino no existe ninguno meter los datos en la cache
 - Si existe devolver un listado para seleccionar el que ya existe.
- La idea final es intentar encontrar al usuario registrado en la caché.
- Por ejemplo si los inputs del formulario contienen más de tres letras una buena aproximación seria que a partir de 3 letras por input ya poder preguntar a la cache en cada pulsación de tecla nueva.

Arquitectura ejemplo

- Caché de hazelcast
- Tomcat que sirve html y que conecta con grid de hazelcast

Esta arquitectura no es más que un ejemplo, hay muchos tipos, descubrir la mejor.

Información

Para generar la base de datos automáticamente podéis usar:

<http://www.generatedata.com/>

Resumen

Resumen de los puntos de la prueba, **requisitos funcionales**.

- Cargar datos en memoria. A tener en cuenta rapidez de carga y diseño.
 - Podéis cargar los datos desde un csv, desde un mysql, etc, no hay restricciones
 - Debéis generar una caché con 500k registros.
- Implementar el modelo de la cache de Usuarios.
- Conectar formulario con la caché.

No es necesario que la interfaz de usuario sea buena, lo importante del ejercicio es el diseño e implementación de la solución. ***Se valorará cualquier aspecto técnico utilizado.***

Para verificar la solución por nuestra parte será necesario que el proyecto se pueda desplegar en una máquina cualquiera (por ejemplo en un tomcat). Pensad que nosotros no tenemos nada y deberemos compilar y ejecutar los pasos que nos indiquéis.

2. Entregables

Para la entrega de la solución es necesario realizarla a través de GitHub o bitbucket y compartir el proyecto con nosotros. Asegúrate que el proyecto contiene la suficiente información para que lo podamos compilar y probar. Cualquier documentación que sea necesaria no te olvides de subirla a GitHub.

Búscanos como:

IS-interviews	en github
pgarcesm	en bitbucket



3. Detalles entrega

Por favor envíanos una vez hayas terminado la URL del proyecto en GitHub o BitBucket antes del 22/02/2015 a las 23:59.

Si decides omitir cualquiera de los requisitos funcionales decirlo claramente.

Desde el diagnóstico inicial a la implantación, estamos siempre.

Y luego iniciamos el siguiente proyecto, y luego otro, y otro...

Gremi d'Hortolans, nº 11,
2ª Planta, Oficina nº 7
Polígono Son Rossinyol -
07009 Palma de Mallorca (Spain)

(+34) 971 770 877
(+34) 971 770 228
innovation@innovation.es
www.innovation.es

Innovation
strategies | Behind.
Beside.
Beyond