

Desafío Técnico - Ingeniero de Software (Assessment Center + Demo Final)

Objetivos

- Se comunica y estructura ideas al trabajar con un par técnico y un rol de negocio.
- Integra feedback en tiempo real.
- Presenta de forma clara y estratégica una solución a una audiencia diversa.
- Demuestra razonamiento técnico y orientación al usuario.

Estructura del Assessment

| Etapas | Duración | Participantes | Propósito |
|--|-----------------|--|--|
| 1. Presentación de la estructura del desafío | 5 min | Personas | Personas realiza una presentación breve de los postulantes y luego expone la estructura del desafío en cuanto a etapas y tiempos. |
| 2. Presentación y discusión del caso | 15 min | Tech Lead + Postulante | Producto presenta el caso a los postulantes, se aclaran dudas (Tech Lead) y se les da tiempo cronometrado para discutir entre ellos posibles soluciones. |
| 3. Desarrollo del Caso: IoT | 45 min | Postulante + Tech Lead | Cada postulante desarrolla su solución, pudiendo hacer consultas de manera abierta a los Tech Lead. |
| 4. Presentación de soluciones - Demo Final | 25 min | Postulante (solo) ante RR.HH., CS, Producto y Tech | Evaluar comunicación, claridad, empatía y dominio de la solución. |

Desarrollo de Caso: IoT en Tiempo Real

Contexto:

Gauss Control monitorea la seguridad vial de conductores de flotas mineras. Cada vehículo envía datos de velocidad vía protocolo MQTT con la siguiente estructura:

```
{  
  "vehicle_id": "VEH123",  
  "timestamp": "2025-08-14T08:01:00Z",  
  "speed": 72  
}
```

El candidato debe implementar un sistema que reciba los mensajes en **tiempo real** y que genere alertas según reglas de seguridad.

Objetivos de la prueba

1. Levantar un **broker MQTT local** con Docker.
2. Suscribirse a los mensajes de velocidad (**vehicles/speed**) y procesarlos **inmediatamente**.
3. Generar alertas si un vehículo excede el límite de velocidad (60 km/h).
4. Generar alertas críticas si un vehículo excede el límite en **3 o más mensajes consecutivos**.
5. Manejar mensajes malformados o datos faltantes.
6. Persistir los datos crudos para poder hacer analítica posterior.
7. Opcional: Publicar las alertas en otro tópico MQTT (**vehicles/alerts**) para consumo por otro servicio.

Requerimientos mínimos

Broker MQTT local

- Sugerencia: Usar **eclipse-mosquitto** o similar con Docker o **docker-compose.yml**.
- Debe estar accesible en **localhost:1883**.

Suscripción y procesamiento en tiempo real

- Conectarse al tópico **vehicles/speed**.
- Analizar cada mensaje al recibirlo.
- Si **speed > 60**, generar alerta simple:

ALERT: VEH123 exceeded speed at 72 km/h at 08:01:00

- Si **3 o más mensajes consecutivos** de un vehículo exceden el límite, generar alerta crítica:

CRITICAL ALERT: VEH123 exceeded speed consecutively 3 times

Robustez

- Manejar mensajes malformados (JSON inválido, campos faltantes).
- Considerar la posibilidad de pérdida de conexión al broker.

Bonus / Extensiones

- Publicar alertas en otro tópico MQTT (**vehicles/alerts**).
- Mejoras a criterio del candidato.

Micro-cronograma sugerido:

- **0–15 min:** Producto y Tech Lead presentan el caso.
- **15–60 min:** Implementación junto a Tech Lead.

Roles y dinámicas:

Tech Lead

- Apoya decisiones técnicas (estructura, patrones, modularidad).
- Da feedback y valida prioridades técnicas.
- Hace preguntas abiertas para evaluar claridad y comunicación.
- No lidera la solución; el postulante marca el rumbo.

Producto

- Presenta la historia de usuario (ver ejemplo abajo).
- Aclara criterios de aceptación.
- Evalúa cómo el candidato pregunta, válida y prioriza.

Ejemplo de historia de usuario:

En nuestra plataforma, un visualizador de sensores en tiempo real, permite a los clientes detectar y reaccionar rápidamente ante eventos críticos en terreno, tales como: Alertas de velocidad, en especial si estas son reiteradas, mejorando la seguridad y eficiencia operativa.

Presentación Multidisciplinaria - Demo Final (25 min)

Narrativa de presentación:

Problema detectado → Solución implementada → Decisiones técnicas → Qué falta / mejoras futuras.

Roles en la preparación:

- **Postulante:** Lidera la narrativa y define ideas clave.
- **Tech Lead:** Refuerza claridad técnica y asegura precisión.
- **Producto:** Traduce lo técnico a lenguaje de negocio y orienta hacia el valor para el usuario.

| Rol | ¿Qué hace? |
|-------------------------|---|
| Postulante | Lidera el contenido: estructura narrativa, ideas clave, mensajes principales. |
| Tech Lead (acompañante) | Apoya en clarificar términos técnicos, sugiere estructura lógica, desafía al postulante a pensar en audiencias no técnicas. |

Recuerda considerar lo siguiente:

- El candidato lidera la narrativa. El postulante se presenta solo, pero puede referenciar trabajo con Tech Lead y Producto.
- El Tech Lead refuerza la claridad técnica (¿cómo explicamos esto?).
- Producto ayuda a bajar la explicación a términos comprensibles para otras áreas.

Además, respecto a cada rol, ten en cuenta:

- Audiencia: RR.HH., Customer Success, Producto y Tech Lead (oyente).
- Panel formula preguntas desde su perspectiva:
 - **Producto:** Priorización y enfoque en el usuario.
 - **Tech:** Calidad y escalabilidad.

Preguntas de diseño/reflexión (Producto/Tech):

- Que explique la arquitectura de la solución
- ¿Qué fué lo más complejo? ¿Qué le faltó?
- ¿Cómo escalarías este sistema para millones de vehículos y millones de mensajes por día?
- **¿Cómo manejarías mensajes duplicados o pérdida de conexión del broker? → Opcional**

- **RR.HH.:** Actitud frente al trabajo en equipo, adaptabilidad.
 - ¿Cómo trabajaste en equipo durante la prueba y qué aprendiste?
 - Pedir feedback del proceso, ¿Cómo se sintieron realizando el desafío?
- **CS:** Impacto en clientes
 - ¿Cómo esta funcionalidad mejoraría la experiencia del cliente en terreno?

El panel hace preguntas desde distintos ángulos.

Criterios de Evaluación:

| Habilidad | ¿Por qué es valiosa? | Indicadores |
|------------------------------|---|---|
| Co-creación | Evalúa cómo integran ideas en equipo. | Integra ideas y colabora con otros roles. |
| Escucha activa | Se ve si toman en cuenta las sugerencias del otro. | Considera sugerencias y ajusta su enfoque. |
| Calidad y traducción técnica | Evitar cometer errores, buenas prácticas de desarrollo y comunicación con áreas de negocio. - Código limpio, modular y documentado (10%). - Manejo de errores y robustez (25%). - Levantar broker MQTT correctamente (10%) - Suscripción y procesamiento en tiempo real (25%). - Detección de exceso simple y consecutivo (30%). | Código legible, buenas prácticas y manejo mínimo de errores. Explica en términos claros para el negocio. |
| Empatía comunicacional | Clave para explicar soluciones a clientes o áreas no técnicas. | Adapta su mensaje según la audiencia. |

Entregables del candidato:

1. Funcionalidad parcial o completa.
2. Capturas o notas si no se completa la implementación.
3. Presentación verbal clara con narrativa alineada.

Hint: Imagina que manejamos cientos de camiones en faena y debemos detectar riesgos en tiempo real. Prioriza lo esencial para dar valor rápido, valida con Producto qué es más crítico, y deja tu código listo para que otro compañero pueda continuar mañana..

¡Vamos con todo!

¡Mucho éxito en este desafío!