

## Intern Full Stack Developer: Caso Práctico

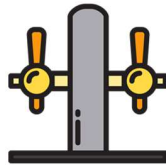
### Statement

Como parte del proceso para un Intern Full Stack & AI Developer, te proponemos un caso práctico para evaluar tu capacidad de diseñar e implementar un flujo funcional extremo a extremo: desde procesar un vídeo hasta exponer y visualizar el resultado.

### Contexto

Disponemos de un tirador doble de cerveza (dos grifos). El propietario quiere saber cuántas cervezas se sirven para poder compararlo con las que figuran en caja.

Se te facilita una carpeta Drive con vídeos reales del tirador en operación. Tu solución debe contar las cervezas servidas en cualquiera de los dos grifos a partir de un vídeo subido por el usuario y mostrar el resultado en una interfaz sencilla.



### Scope del caso

Para mantener el alcance acotado, céntrate exclusivamente en el conteo de tiradas (no en volumen o tamaño del vaso).

Tu sistema debe ser capaz de:

1. Detectar y contar cada cerveza servida por grifos A y B durante la duración de un vídeo.
2. Persistir el resultado (p. ej., en SQLite/Postgres) para consultas posteriores.
3. Levantar un backend simple que permita:
  - Subir un vídeo (mp4/mov).
  - Lanzar el proceso de conteo.
  - Consultar los conteos por vídeo/fecha/grifo.
4. Mostrar en una interfaz mínima (frontend) que:
  - Permite subir el vídeo de prueba.
  - Muestra los conteos por grifo y el total.
  - Indica el estado del procesamiento (pendiente/en curso/completado o error).

### Objetivo de validación durante la presentación

En la sesión de defensa:

- Subiremos un nuevo vídeo (de la misma carpeta facilitada u otro equivalente).
- Verificaremos que tu app:
  - 1) Acepta la subida,
  - 2) Procesa el vídeo,
  - 3) Devuelve el número correcto de cervezas servidas en cada grifo y el total.

### Material proporcionado

- Carpeta [Drive](#) con vídeos reales del tirador (dos grifos). Úsalos para desarrollar y probar.

### Requisitos funcionales mínimos

- Conteo por eventos: cada vez que comienza y termina una tirada se contabiliza 1 unidad para el grifo correspondiente.
- Multigrifo: identificar si el evento ocurre en el Grifo A o Grifo B.
- Robustez básica: tolerar o reducir falsos positivos por manos, vasos en espera, goteos residuales, etc.
- Procesamiento offline: el conteo se hace sobre un vídeo subido (no hace falta tiempo real).

### Requisitos no funcionales

- Simplicidad y claridad del código y la arquitectura (nivel Intern).
- README con pasos para lanzar backend y frontend (Docker o comandos simples).
- Tiempo de proceso razonable para vídeos como los facilitados en un portátil estándar.
- Persistencia mínima (p. ej., SQLite con migraciones simples).

### Frontend mínimo

- Formulario para subir vídeo.
- Botón Procesar y spinner/estado.
- Tarjetas con Conteo Grifo A, Conteo Grifo B, Total.
- (Opcional) Timeline con eventos.

### Criterios de evaluación

1. Funcionalidad.
2. Claridad.
3. Robustez.
4. Arquitectura.
5. Datos.
6. Trade-offs.

### Entrega y presentación

Para la realización del caso, dispones de un plazo de hasta 72 horas desde la confirmación de recepción del Caso práctico.