

Challenge Técnico: Pipeline de Contenido con IA

(Newsletter-to-Post)

Esto es una prueba realista, en el sentido de que se asemeja a una tarea que podrías acometer mañana como parte de Magnetu. Se trata de una prueba de concepto, como tantas otras que hacemos habitualmente, con el objetivo de valorar soluciones y expandir el negocio. Queremos ver cómo te desenvuelves desde la recepción de una idea hasta su implementación técnica básica.

1. El Contexto

En Magnetu ayudamos a agencias y marcas a brillar en LinkedIn. Muchos de nuestros clientes reciben newsletters valiosas y queremos automatizar el proceso de convertir ese conocimiento en posts de LinkedIn listos para publicar.

2. El Reto (PoC)

Debes construir una pequeña aplicación (o servicio) que realice el siguiente flujo:

1. **Input:** Recibir el contenido de un email (texto plano o HTML). *Nota: Puedes simular esto con un endpoint POST que reciba un JSON.*
2. **Procesamiento (IA):** Enviar ese contenido a un LLM (OpenAI/Claude/Gemini) con un prompt que lo transforme en un post de LinkedIn atractivo (formato corto, con emojis y hashtags).
3. **Output & Storage:** Guardar en una base de datos (MongoDB preferiblemente) el post generado, el texto original y el **conteo de tokens** utilizados.
4. **Visualización (Opcional pero valorado):** Una interfaz mínima para ver el historial de posts generados.

3. Stack Sugerido (Alineado con Magnetu)

- **Backend:** Node.js / Express.
- **Base de datos:** MongoDB (Mongoose).
- **IA:** Conexión con cualquier API de LLM. No es necesario que tengas una cuenta de pago en OpenAI. Para esta prueba puedes utilizar cualquier API gratuita. Te recomendamos:
 - Google AI Studio (Gemini): Gratis y fácil de configurar en ai.google.dev.
 - Groq: Modelos Llama 3.

- OpenRouter: Acceso a múltiples modelos gratuitos en un solo endpoint.
- **Lenguaje:** TypeScript (altamente recomendado).

4. ¿Qué evaluaremos?

- **Calidad del Código:** Estructura modular y legibilidad.
- **Gestión de Errores:** ¿Qué pasa si la API de la IA falla?
- **Tracking:** Capacidad para medir el uso de recursos (tokens).
- **Visión de Producto:** Al final del ejercicio, queremos conocer tu opinión sobre:
 - ¿Cómo harías este sistema escalable para 1,000 newsletters diarias?
 - ¿Qué riesgos técnicos ves (latencia, límites de API)?
 - Estimación de tiempo para llevar esto a una versión "Production Ready".

5. Logística y Transparencia

- **Tiempo estimado:** 2-4 horas. No buscamos perfección, sino ver cómo priorizas.
- **Entrega:** Un repositorio de GitHub (público o privado).
- **Propiedad Intelectual:** Este código es **solo para evaluación**. Magnetu no utilizará este código en su producto; es una prueba de concepto aislada.
- **Defensa en vivo:** Tendremos una sesión de 30 min para revisar el código, hacer un pequeño cambio en vivo y probar el envío de un email real a tu sistema.

6. Alcance, Tiempo y Compromiso

Queremos ser muy claros y respetuosos con tu tiempo:

- Tú decides el alcance: El tiempo sugerido mínimo es de 2 horas, pero tienes total libertad para dedicarle más si deseas profundizar en algún área (UI, seguridad, despliegue, etc.).
- Evaluación proporcional: Evaluaremos la prueba en función de la extensión que decidas darle:
 - Si decides hacer "poco": Seremos muy exigentes con la calidad de ese poco. Esperamos un código impecable, una arquitectura robusta y una lógica sin fisuras.
 - Si decides hacer "mucho": Valoraremos positivamente tu ambición, capacidad de ejecución y las conclusiones globales que extraigas del proceso, incluso si algún detalle técnico queda como "borrador".
- La utilidad es clave: Si por falta de tiempo no logras completar el flujo, asegúrate de que lo que entregues sea de utilidad. Por ejemplo, un backend perfectamente documentado con

una estrategia de escalabilidad clara tiene mucho más valor para nosotros que un sistema completo que no funciona o está mal estructurado.