El objetivo de esta prueba técnica es evaluar tu capacidad para construir una API funcional y precisa que calcule fechas hábiles en Colombia, teniendo en cuenta:

\*Horarios laborales (lunes a viernes, de 8:00 a.m. a 5:00 p.m., con horario de almuerzo de 12:00 p.m. a 1:00 p.m.).

\*Zonas horarias (el cálculo debe hacerse en hora local de Colombia y la respuesta debe ser en UTC)

La API debe recibir una cantidad de horas y/o días hábiles a sumar a partir del momento actual (en hora colombiana), y devolver como respuesta la fecha y hora resultante ya convertida a UTC

**🛠️ Instrucciones**

Desarrolla una API REST que reciba una petición *GET* a cualquier ruta de tu elección. La API debe aceptar los siguientes **parámetros en query string**:

* "days": número de días hábiles a sumar (opcional, entero positivo)
* "hours": número de horas hábiles a sumar (opcional, entero positivo)
* "date": fecha/hora inicial en UTC (ISO 8601) con sufijo Z (opcional). Si se provee, será el punto de partida y se convertirá a hora local de Colombia para aplicar las reglas de negocio; si no se provee, el cálculo inicia desde la hora actual en Colombia.

*Si se envían ambos parámetros, la suma debe hacerse en orden: primero los días, luego las horas.*

*Si no se envía ninguno, la API debe retornar un error.*

**Respuesta esperada:**

Respuesta exitosa (**200 OK**):

JSON, exactamente :

{

"date": "2025-08-01T14:00:00Z"

}

*(clave obligatoria "****date****", valor en UTC ISO 8601 con Z, sin campos extra).*

*Errores (400, 503, etc.):*

*{ "error": "InvalidParameters", "message": "Detalle del error" }*

**Importante**: Toda la solución debe estar implementada en **TypeScript**, incluyendo tipado explícito en todas las funciones, interfaces, tipos de respuesta, estructuras de datos y cualquier otra parte donde sea pertinente. **No se aceptará código en JavaScript ni tipado implícito.**

**📚 Reglas del Negocio**

* El cómputo debe iniciar desde la hora **actual en Colombia** (zona horaria: *America/Bogota*) si el parámetro de fecha no es proporcionado.
* Si el parámetro **date** es proporcionado, se pasará como una fecha en formato UTC y los cálculos de días y horas deben hacerse con la zona horaria de Colombia.
* Los días hábiles son de lunes a viernes.
* El horario laboral va de 8:00 a.m. a 5:00 p.m. (hora de Colombia), con almuerzo de 12:00 p.m. a 1:00 p.m.
* Si la fecha ingresada esta por fuera del horario de trabajo o no es un día laboral debe aproximarse hacia atrás al día y/o hora laboral más cercano.
* Los días festivos colombianos deben excluirse. Puedes consultar un arreglo actualizado de días festivos en el siguiente recurso: https://content.capta.co/Recruitment/WorkingDays.json
* El resultado debe ser retornado en formato UTC (ISO 8601).

**📌 Ejemplos**

**1.** Petición un viernes a las 5:00 p.m. con "hours=1"

Resultado esperado: lunes a las 9:00 a.m. (hora Colombia) → "2025-XX-XXT14:00:00Z" (UTC)

**2.** Petición un sábado a las 2:00 p.m. con "hours=1"

Resultado esperado: lunes a las 9:00 a.m. (hora Colombia) → "2025-XX-XXT14:00:00Z" (UTC)

**3.** Petición con "days=1" y "hours=4" desde un martes a las 3:00 p.m.

Resultado esperado: jueves a las 10:00 a.m. (hora Colombia) → "2025-XX-XXT15:00:00Z" (UTC)

**4.** Petición con "days=1" desde un domingo a las 6:00 p.m.

Resultado esperado: lunes a las 5:00 p.m. (hora Colombia) → "2025-XX-XXT22:00:00Z" (UTC)

**5.** Petición con "hours=8" desde un día laboral a las 8:00 a.m.

Resultado esperado: mismo día a las 5:00 p.m. (hora Colombia) → "2025-XX-XXT22:00:00Z" (UTC)

**6.** Petición con "days=1" desde un día laboral a las 8:00 a.m.

Resultado esperado: siguiente día laboral a las 8:00 a.m. (hora Colombia) → "2025-XX-XXT13:00:00Z" (UTC)

**7.** Petición con "days=1" desde un día laboral a las 12:30 p.m.

Resultado esperado: siguiente día laboral a las 12:00 p.m. (hora Colombia) → "2025-XX-XXT17:00:00Z" (UTC)

**8.** Petición con "hours=3" desde un día laboral a las 11:30 p.m.

Resultado esperado: mismo día laboral a las 3:30 p.m. (hora Colombia) → 2025-XX-XXT20:30:00Z (UTC)

**9.** Petición con "date=2025-04-10T15:00:00.000Z" y "days=5" y "hours=4" (el 17 y 18 de abril son festivos)

Resultado esperado: 21 de abril a las 3:00 p.m. (hora Colombia) → "2025-04-21T20:00:00.000Z" (UTC)

**🤖 Uso de Herramientas de Inteligencia Artificial**

El uso de herramientas de asistencia como ChatGPT, GitHub Copilot, Stack Overflow, etc. **está completamente permitido**.

Lo que nos interesa es que entregues una solución funcional, clara y con la que estés familiarizado.

**Eso sí:** podríamos preguntarte en una entrevista posterior por qué tomaste ciertas decisiones técnicas, así que asegúrate de entender tu propio código.

**✅ Criterios de Evaluación**

* Correctitud en el manejo de fechas y lógica de negocio.
* Implementación clara, modular y mantenible.
* Correcto uso de zona horaria de Colombia para los cálculos y conversión final a UTC.
* Validación de errores (parámetros faltantes, inválidos, etc.).
* **Optimización en el uso de recursos**: eficiencia en memoria, complejidad algorítmica y claridad estructural del código.
* **Uso correcto y consistente de TypeScript**: tipado explícito donde aplique, incluyendo tipos personalizados para estructuras de entrada/salida y lógica interna.

**🚀 Entrega:**

Deberás entregar:

* URL de un repositorio público en GitHub con toda la solución implementada en TypeScript y un README.md con instrucciones claras para instalación y ejecución local.
* La URL exacta de despliegue del endpoint accesible públicamente (Vercel, Railway, Render, etc.).

**Bonus**: se considerará un plus si la solución está desplegada como una función Lambda utilizando AWS CDK.

**⚠️Validación:**

Una vez entregues la URL de despliegue, se verificará de forma automática que tu API cumpla exactamente con lo solicitado.

* Los nombres de los parámetros deben coincidir exactamente (days, hours, date).
* La estructura de la respuesta debe ajustarse al contrato definido, tanto en éxito como en error.
* Los códigos de estado HTTP y el Content-Type deben ser correctos.

**Importante:** Si tu API no cumple con el contrato en cualquiera de estos puntos, la entrega podrá ser rechazada de forma automática.

.