

Challenge Técnico / Backend - Menta Tickets

¡Hola!

Nuevamente gracias por tu tiempo en la entrevista donde pudimos conocernos. Tal como adelantamos, tenemos un proceso corto pero eficaz para nuestros candidatos a sumarse al equipo de Menta.

1	Entrevista de presentación	30m - 1h	✓
2	Challenge técnico asincrónico ¹	Max 3h	⌚
3	Entrevista técnica final con Líderes de equipo	1h - 2h	⌚
-	Tiempo total deseado del proceso	≈6h máx	⌚

Diseñamos estos espacios para ser respetuosos con el tiempo de cada persona

Asumimos el compromiso de siempre comunicar resultados finales de la selección. También nos comprometemos a no demorar más de 10 días² el proceso completo.

Somos una startup, está en nuestro ADN movernos rápido

A tener en cuenta:

- El tiempo de resolución recomendado es de 3 horas. Entendemos que puede demorar más pero pedimos que no se exceda mucho más de ese tiempo. Podes hacerlo cuando quieras, e incluso con intervalos. Manejalo a tu gusto.
- Nuestro stack principal es Typescript pero puedes desarrollarlo en **cualquier lenguaje de programación**. El que más te guste o sientas más seguridad. Si Typescript no te suena familiar y te invita a un desafío, te invitamos a que lo hagas en ese lenguaje
- Valoramos la aplicación de todo conocimiento que tengas, es tu momento de romperla y dar el máximo hasta en lo más mínimo.
- No existe una única forma de resolverlos los problemas, tampoco esperamos soluciones completas. Lo importante es entender la forma en la cual encaraste el problema.
- Cualquier consulta puedes escribirle a tu punto de contacto con Menta Tickets por donde creas conveniente.
- Buena suerte :D



1. El tiempo recomendado de resolución del challenge es de 3 horas. Puede demorar más, o incluso menos.
2. Contando de lunes a viernes.

CHALLENGE TÉCNICO

Los caracteres que utilizamos todos los días para escribir, tanto en español, inglés y otros lenguajes que comparten las mismas raíces tienen una particularidad: puede ser de trazos cerrados o bien de trazos abiertos.

¿A qué nos referimos cuando hablamos de trazos cerrados? Básicamente que hay partes del carácter que quedan aisladas del exterior, habitualmente con una circunferencia, y esta puede ser rellena o vacía.

Por ejemplo, la letra **a** tiene un trazado completo que forma un círculo en su interior.

Otro ejemplo es la **B**, que tiene **dos** espacios cerrados. La letra U, por el contrario, ninguno. Y así con todos los caracteres que habitualmente utilizamos para escribir.

Nos interesa contar la cantidad de “espacios cerrados” tiene una cadena de texto que ingresamos en nuestro programa. Puede ser una oración completa o incluso un párrafo.

Consideraciones principales

1. La entrada de texto únicamente será de caracteres que se utilicen en el español (letras [incluso con tildes o símbolos (á, é)], números y algunos símbolos [[aclarados al pie](#)]).
2. Ej: Frente a la cadena de texto de entrada: “El 37% de los humanos está bancarizado” el resultado debería ser 15.
3. Puntos que se encuentren en las letras también deberían sumar: (ejemplo la **i** o la **j**)

Si sobra tiempo y quieres dar un extra mile:

- Montar un servicio API REST para enviar una cadena mediante HTTP y obtener una respuesta codificada en JSON con el resultado. Utiliza la implementación que creas conveniente.
- Realizar manejo de errores
- Realizar una ideación extensible que no solo permita soportar español, sino lenguajes como el francés o el alemán (ej), sin volver a compilar el programa para agregar un nuevo lenguaje.

Y si aún querés más:

- Deploya la API REST en algún proveedor de cloud computing y generá una URL estática para consumir el servicio públicamente desde internet.

Símbolos permitidos: @ # \$ % & ' () = ? _ - el resto de los símbolos **no** son permitidos.

¡Gracias por tu tiempo y muchos éxitos!