



Copyright © 2021 NeuralWorks. Confidential and proprietary

Challenge FullStack Developer

Instrucciones

- Debes entregar tu solución en un repositorio GitHub
- En el repositorio deben estar todos los archivos utilizados para la resolución de tu desafío. - La solución debe estar implementada algún framework moderno de JavaScript o Python para desarrollo web. Por ejemplo: React, Next, Angular, Fast API, Flask, etc.
- Recuerda que no estamos en tu cabeza! Escribe los supuestos que estás asumiendo y documenta tu código. - Para este desafío te recomendamos que describas claramente cómo mejorar cada parte de tu ejercicio en caso de que tenga opción de mejora.
- Debes subir el link al repositorio en el formulario enviado, máximo 5 días de corrido después de haberlo recibido.

Problema

El equipo de NeuralWorks ha tenido problemas con el manejo del tiempo en home office por la pandemia y ha decidido crear NeuralClocks, una herramienta que permita optimizar los tiempos tanto personales como del equipo en si. Se han pensado features cómo modificar las horas de las reuniones entre el equipo para optimizar la intersección de los bloques de trabajo en los pomodoros de cada asistente, unirse al temporizador de alguno de sus compañeros para cuando se realice pair-programming o planificaciones en conjunto, entre otros.

Si bien hay muchas ideas y ambición de lo que se puede construir con NeuralClocks, el paso número uno es crear un timer para poder seguir la metodología Pomodoro. Tu desafío es crear esta aplicación de temporizador Pomodoro.

Puedes usar cualquier librería o herramienta que te guste para ayudarte a completar el desafío. Sus usuarios deberían poder:

1. Establecer un temporizador Pomodoro y temporizadores de pausas cortas y largas
2. Personaliza la duración de cada temporizador
3. Ver una barra de progreso circular que se actualiza cada minuto y representa qué tan lejos de su temporizador están
4. Poder ver un historial de sus pomodoros realizados. Se recomienda el uso de una base de datos y una conexión por API o GraphQL.

Evaluación

- Recuerda escribir los supuestos que tomaste y argumentar tus decisiones.
- Documentar MUY bien tu trabajo. Recomendamos utilizar un README o markdown donde puedas contar y dar a entender tus decisiones y supuestos. Recuerda que no estamos en tu cabeza!

- Criterios a considerar:

338cdbc9caf57f08d0d2a0304a0ea361

- Creatividad en las técnicas y/o herramientas utilizadas.
- Simplicidad y eficiencia.
- Performance.
- Calidad de conclusiones.
- Orden y documentación.