

¡Hola Querid@ Candidat@!

Te damos la bienvenida a **TalaTask**, un emocionante desafío donde pondrás a prueba tus habilidades para construir una API que gestione la asignación de tareas. Este desafío es parte de tu postulación al puesto de desarrollador@ Software Developer Engineer en **Talana**. ¡Haz tu mejor esfuerzo! No te preocupes si no terminas todo, lo importante es que muestres tu capacidad de resolver problemas.

Duración y en qué consta la prueba:

- **Duración:** 72 horas (Si lo terminas antes, genial. Si necesitas un poco más de tiempo, no te preocupes, valoramos más la calidad que la velocidad).
- **En qué consiste:** Desarrollarás una API que asigne tareas a empleados de manera óptima, utilizando la disponibilidad y habilidades de cada uno.

Forma de entrega:

- El proyecto deberá estar en un repositorio compartido.

Lenguajes, frameworks y herramientas:

- Usa el lenguaje que más te acomode. (**Dato útil:** *Python o Go, ¡nos encanta! Pero en realidad, lo que queremos ver es cómo piensas y resuelves problemas, así que elige lo que te haga sentir más cómodo*)

¡Manos a la obra!

En este desafío, deberás crear una API que asigne tareas a un grupo de empleados, optimizando el uso de sus habilidades y tiempo. La empresa tiene una serie de tareas que deben ser completadas en un período específico, y cada empleado tiene una disponibilidad limitada.

¿Qué necesitamos que implementes?

1. Empleados:

- Cada empleado debe tener un nombre, un conjunto de habilidades (por ejemplo, programación, diseño, análisis, coordinación, etc.) y una disponibilidad en horas por día. Además, cada empleado debe tener un conjunto de días específicos en los que está disponible para trabajar.

La idea es que definas quiénes son, qué pueden hacer, cuánto tiempo tienen para trabajar cada día y cuales son sus días disponibles.

2. Tareas:

- Cada tarea tiene un título, una fecha específica y una duración estimada (en horas) y requiere un conjunto de habilidades específicas para completarla. Aquí lo importante es definir qué tareas existen, cuánto tiempo requieren, y qué habilidades son necesarias para hacerlas.

3. Asignación de tareas:

- Desarrollar un algoritmo que asigne las tareas a los empleados disponibles en la fecha específica y de la manera más eficiente posible. Debe asegurarse que las tareas solo se asignen a empleados que estén disponibles tanto en términos de días como de horas, maximizando la utilización del tiempo y asegurando que las tareas se asignen a quienes tienen las habilidades adecuadas. ¡Queremos ver cómo haces que todo encaje!

4. Reporte de asignaciones:

- Este endpoint debe mostrar cómo quedan distribuidas las tareas entre los empleados para una fecha específica, indicando el uso de tiempo y habilidades de cada uno. Queremos ver un resumen claro y ordenado de cómo se resolvió todo.

Notas.

Sabemos que algunos requerimientos pueden parecer un poco ambiguos, y es totalmente a propósito. Queremos ver cómo te desenvuelves tomando decisiones y haciendo supuestos para resolver los desafíos. ¡Confía en tu criterio!

No te compliques demasiado con las reglas, pero asegúrate de que las tareas no superen la disponibilidad horaria de los empleados y que se respeten las habilidades requeridas.

La clave aquí es mantener todo simple y eficiente. No se trata de hacer el sistema más complejo, sino de resolver el problema.

Evaluación:

- Nos enfocaremos en cómo resuelves las asignaciones de tareas, la eficiencia de tu algoritmo y cómo manejas las restricciones.
- Además, en Talana nos preocupamos por que nuestras aplicaciones sean de calidad y se mantengan así, por lo que valoramos mucho que nos muestres cómo testear tu código y cómo aseguras que cada incremento sea más robusto.
- Sabemos que el código ordenado y escalable es fundamental, así que si eres fanático de los patrones de diseño y sabes cómo usarlos para solucionar problemas, ¡nos encantaría verlo reflejado en tu solución!
- Queremos ver tus habilidades en acción, así que el proyecto tiene que estar dockerizado. Asegúrate de incluir un Dockerfile y, si lo necesitas, un archivo docker-compose.yml para que todo funcione de maravilla.
- Y si además logras deployar tu API en alguna nube, ¡te ganarás puntos extra! Y si usas Kubernetes (K8s) para hacerlo, será un gran plus.
- Por último, la documentación también es clave, así que no la descuides. Queremos entender claramente cómo funciona tu solución y cómo implementaste cada parte.

¡Buena suerte! Estamos emocionados por ver cómo resuelves este desafío y cómo optimizas los recursos en TalaTask