El empleado inicia el trabajo y un cronometro empieza contar el tiempo transcurrido desde que empieza su trabajo. El empleado ejecuta las tareas por paso entonces es necesario tener claro el performance del empleado en cada paso para calcular el performance por tarea. Al final de su trabajo del día es recuperada la información del cronometro para guardar el tiempo transcurrido en total para el completado de su trabajo del día.

Inicia el trabajo - Inicia el timer

**Completa un paso** - Guarda el timestamp y el tiempo transcurrido en ese paso.

Es necesario tener guardado un objeto que permita entender el performance total de cada día

Dado lo anterior los updates de tiempo transcurrido en casa paso deberán de ser guardados en un objeto como el siguiente:

Las tareas deben estar mappeadas para entender la relación entre la orden y la tarea que se ejecuta, se requiere trackear el perfomance del empleado por lo que es necesario que el sistema sea capaz de:

Ver el tiempo transcurrido por tarea en un día especifico

Ver el promedio de tiempo transcurrido por tarea por empleado

Ver los pasos de una tarea, y el tiempo transcurrido en cada una de ellas

Ver una orden, las tareas que se ejecutaron para esa orden, y el tiempo transcurrido total en esa orden, el tiempo por tarea en esa orden, y el tiempo por cada paso de las tareas de esa orden

Al final, es necesario tener acceso a todos estos parámetros para poder calcular el performance del empleado trackeando el tiempo por orden, por empleado, por orden, por tarea y por paso

Las tareas estarán en la base de datos como una colección estática únicamente para referenciarla en el campo de performanceHistory del empleado, donde se almacenará toda la información de la ejecución de tareas de los empleados:

Referencia al tipo de tarea

timeStamp de inicio

Referencia a los pasos de la tarea y su tiempo transcurrido (arreglo de objetos con \_Id del step y el tiempo transcurrido de ejecucion)

Tiempo total transcurrido de ejecución

timeStamp de finalización

{

“seeding”: {

“stepName”: {

“CompletionTimeStamp”: new Date(),

“timeTaken”: time,

}

}

}