

## Time Tracking Estimación de la tarea GTTT 18

Esto es un esbozo de los cambios aproximados que se tendrán que implementar para poder crear un nivel inferior al de Registro, que será la llamada Tarea.

Con esta nueva fórmula no trabajarán todos los departamentos ni tampoco todos los proyectos, con lo que va a ser opcional, por lo que para los proyectos que no contengan tareas, este nuevo cambio debe de ser transparente, pero a la vez funcional para los que sí que la tengan.

### Base de Datos

Se tendrá que crear otra tabla para listar todas las tareas y asociarlas al proyecto correspondiente:

Task	
PK	ID
FK1	project_ID
	name
	status

Siendo:

- `ID` int NOT NULL AUTO\_INCREMENT
- `project\_ID` int NOT NULL
- `name` varchar(128) NOT NULL
- `status` char(1) NOT NULL DEFAULT '1',
  - FOREIGN KEY (project\_ID) REFERENCES Project(ID)

Además, se tendrá que crear una nueva columna en la tabla Register para definir la tarea asociada a esa entrada de horas:

- `task\_ID` int NOT NULL DEFAULT 0
  - FOREIGN KEY (task\_ID) REFERENCES Task(ID)

### Objetos de Datos

Se tendrá que crear un nuevo TaskDTO con los mismos campos que la tabla.

Es el encargado de mapear los resultados de la llamada a BD

## Objeto de Negocio

También habrá que crear un TaskBO que será el objeto con el que se trabaja en el back-end que tendrá los mismos datos que su DTO.

También modificar el RegisterBO añadiendo el campo task\_ID y lo mismo para el objeto ProjectBO, que se tendrá que añadir un nuevo campo: `List<TaskBO>`

El nuevo campo del ProjectBO será útil a la hora de mostrar o no las tareas que tiene este proyecto y el campo que pertenece al RegisterBO será el encargado de guardar en BD esta especificación: para el proyecto 'x' (como estaba hasta ahora) está definido (si fuera el caso) la imputación de las horas en la tarea 'y'.

## Pantalla Record Time

Para mostrar esta página se trabaja con el objeto de negocio PeriodBO, que a su vez tiene una `List<RegisterBO>`.

Por un lado, en esta pantalla, se muestran los proyectos asociados a cada usuario a través de una lista de objetos ProjectBO (en el momento en que se entra en la página, esta lista se pasa a través del objeto Model) pero cuando se navega entre diferentes usuarios (para el caso de los managers) esta lista se refresca mediante llamadas Ajax.

Entonces, aquí entra en juego el nuevo campo del ProjectBO, la lista de TaskBO. Si esta lista estuviera vacía, el comportamiento sería como hasta entonces, pero si estuviera cargada, en el momento de seleccionar un proyecto

Mon 13	Tue 14	Wed 15	Thu 16	Fri 17	Sat 18	Sun 19
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
0	0	0	0	0	0	0

Total Week Hours: 0

Ese select tendría que retraerse y mostrar en la misma fila un nuevo select con la lista de tareas asociadas a ese proyecto.

Para eso se tiene que modificar la tabla 'period' para la primera vez que se entra y luego recargar esto dinámicamente en el método `loadUserRegisters(...)`.

Cuando el usuario clique en el botón 'Save', se tiene que hacer los cambios necesarios en el DAO para poder guardar en BD esta nueva información (de esto se encarga el `savePeriod(...)` del `TimeServiceImpl`).

## Pantalla Approve per Week

Aquí, algo parecido, pero solo se muestran los proyectos/tareas. Entonces, para los proyectos que tengan una tarea asignada, se deberá mostrar el nombre del proyecto y el de la tarea.

## Pantalla Approve Month / Monthly Summary

A diferencia de las 2 vistas anteriores, aquí el contenido que se muestra es siempre estático. Para cambiar de un mes a otro, o en el caso de los managers, de un usuario a otro, no hay una llamada asíncrona, si no que vuelve a entrar en el método principal del controlador, recarga la información con el usuario y mes y la muestra.

Esto son dos pantallas diferentes lo que pasa que en la primera se aprueban las horas y en la segunda se puede generar el Excel. Se tendrá que adaptar los métodos que ataquen a la base de datos necesarios y lo del report se define a continuación.

## Administración

Se tendrá que crear un nuevo apartado donde se puedan crear nuevas tareas asignándolas a los proyectos ya existentes.

Si las tareas son relativamente estables y no muy cambiantes se puede gestionar desde el mismo módulo de Administración, sino se tendría que plantear de añadir en la página de user Info.

De una manera u otra, lo que se tiene que hacer es un Modal para poder crear las tareas (parecido a lo que hay para crear nuevos proyectos desde Admin) y otro para poder modificarlas, por si se tiene que cambiar algo o dar de baja.

## Reports

Se tendrá que crear una nueva columna en el reporte de los excels (AdministrationServiceImpl.java) modificando el método createDetailedSheet y el método que alimenta el excel que es el getDaysFromNaturalMonthSortedByDay(...)

Por lo que se tiene que añadir un nuevo campo para poder listar las tareas asociadas a los proyectos.

Ahora mismo, para un día, como puede haber varios proyectos, esto queda reflejado en el atributo `List<String> projectNames`

Para poder mostrar las tareas, en el caso de que hubieran, se podría crear u otra lista o un objeto que incorporara Project y Task.