



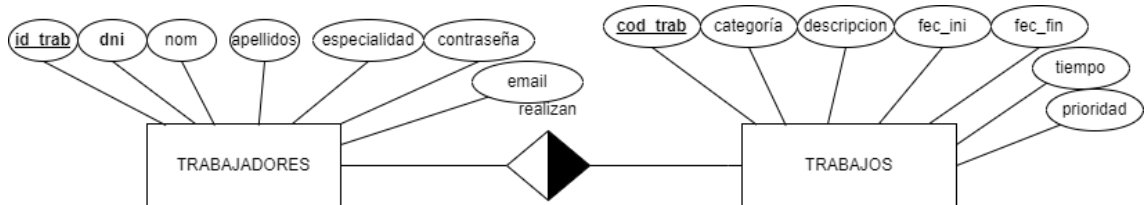
# PRÁCTICA 4 - SPRING

DAMDAW – ACCESO A DATOS

Hector Molina Beneyto  
[hectormolina@iessanvicente.com](mailto:hectormolina@iessanvicente.com)

## INTRODUCCIÓN

A partir de la siguiente base de datos que se muestra a continuación, que contiene trabajadores y trabajos, de la cual se adjunta script de generación en la tarea, vamos a hacer un programa con Spring Tool Suite que trabaje sobre esa base de datos mediante el uso de Servicios Web.



La base de datos simula una base de datos sencilla de una empresa que contiene trabajadores, a los que se le pueden asignar trabajos concretos, que en este caso van a ser sobre el mantenimiento de hotel que tenéis en la práctica de Despliegue de Aplicaciones Web.

## DESARROLLO API

De las tablas indicadas anteriormente se tendrá que permitir las operaciones básicas de un CRUD (insertar, consultar, actualizar y borrar). Esta es la parte de la API que puede probarse con Postman/Similares. Para la construcción de los servicios Web debes hacerlo como se explica en el documento “Spring MVC. Construyendo una API”

Además del CRUD de las tablas, se deberán desarrollar una serie de métodos en la API, de tal manera que se cumplan los requisitos del resto de aplicaciones de la infraestructura del proyecto conjunto. Las funcionalidades a añadir son:

- Obtener trabajos pendientes (login para el trabajador en la app móvil)

- Pasándole por parámetros el id de usuario y la contraseña, si coinciden en la BD, devolverá una lista de todos los trabajos pendientes de ese trabajador (es decir, con el fec\_fin a null).
- Obtener trabajos finalizados
  - Función similar al de pendientes, pero mostrando los trabajos finalizados por ese trabajador.
- Finalizar trabajo
  - Pasándole la fecha actual, actualizará ese trabajo en la tabla, poniéndole esa fecha en la columna fec\_fin, de tal manera que se marque ese trabajo como finalizado.
- Asignar trabajo
  - Pasándole el id de un trabajador, se le asigna ese trabajo al trabajador, actualizando la columna clave ajena que hay en la tabla trabajo. Este método deberá comprobar que la categoría del trabajo y la especialidad del trabajador coinciden, y de no ser así devolverá un error.
- Crear trabajo con trabajador asignado
  - Crear un trabajo con un id de trabajador ya asignado, que se le pasará por parámetros.
- Listar tareas sin asignar
  - Este método mostrará el detalle de todas las tareas sin asignar, mostrando toda la información de cada uno de ellos.
- Listar tareas sin realizar (de todos los trabajadores)
  - Mostrará el detalle de todas las tareas sin finalizar de todos los trabajadores.
- Listar tareas realizadas (de todos los trabajadores)
  - Devolverá el detalle de todas las tareas realizadas por todos los trabajadores.
- Listar tareas por trabajador entre fechas
  - Mostrará todas las tareas finalizadas por un trabajador concreto, entre dos fechas específicas.
- Listar tareas de un trabajador ordenadas por prioridad

- Mostrará todas las tareas pendientes de un trabajador concreto, ordenadas por prioridad (De la 1 -urgente- a la 4 -menos urgente-)
- Listar tareas de un trabajador de una prioridad concreta
  - Mostrará las tareas pendientes de un trabajador, solo de la prioridad que se le pase en la petición (Por ejemplo, solo las de prioridad 1 para las más urgentes)

## DESARROLLO VISTAS

También crearemos páginas html (vistas) para poder hacer las operaciones del CRUD. Para ello, debe crearse una web en Thymeleaf (o cualquier otra tecnología que conozca el alumno, como Angular por ejemplo) donde estén los diferentes menús (trabajadores y trabajos) y en ellos se permita realizar todas las operaciones asociadas del CRUD de cada tabla de la base de datos (y de la tabla que los relaciona evidentemente). Para la parte de Thymeleaf puedes basarte en el pdf “Spring MVC. Construyendo una API” que aparece en el tema. Además, se deberá utilizar algún framework como bootstrap o similar, para dar estilo a las vistas, de tal manera que se puntuará la elegancia y estilo de estas, además de que funcionen correctamente.

## DOCUMENTACIÓN

Para simular completamente el trabajo colaborativo en el entorno laboral, una de las tareas mas importantes de un desarrollador que realiza trabajos que van a ser utilizados o manipulados por otra gente, es la realización de documentación oficial que pueda consultarse en caso de formación o de dudas sobre el uso de dichos servicios. Habrá que realizar la documentación tanto de la API como de las vistas, mostrando un breve resumen de cada una de las funcionalidades de la aplicación, así como un ejemplo de uso en desarrollo, así como un ejemplo de

entrada y de salida de cada una de las funcionalidades. Para las vistas, también habrá documentación, mostrando capturas sobre su utilización y una explicación de su uso.

## **CRITERIOS DE CALIFICACIÓN**

\*Por concretar\*