ITV PEPILANDIA

Tenemos nuestra central de ITV en Pepilandia.

Nuestra central tiene una serie de **trabajadores** de los cuales nos interesa saber su nombre, teléfono, email (único), nombre de usuario (único), contraseña (cifrada) fecha de contratación. A su vez cada trabajador tiene una especialidad que es: ELECTRICIDAD, MOTOR, MECANICA e INTERIOR. Su **salario** depende de dicha especialidad ELECTRICIDAD(1800), MOTOR(1700), MECANICA(1600) e INTERIOR(1750) **unido a un 100€ más por cada tres años trabajados de antigüedad**.

Además, tenemos un **responsable de ITV** que es un trabajador y a parte tiene un plus de dirección de +1000€.

Nuestro taller **gestiona las citas en intervalos de 30 min**, y cada cita será atendida por un trabajador sin en ese momento tiene hueco. **Este trabajador, no podrá atender más de 4 citas por intervalo y no podremos tener más de 8 citas en el mismo intervalo**. Para atender una cita necesitamos los datos del **vehículo**: marca, modelo, matrícula, fecha de matriculación, fecha de última revisión. Además, necesitamos los datos del **propietario/a**: DNI, nombre, apellidos, teléfono.

Nuestra ITV, por inspección, emitirá un **informe** favorable o no, donde se describa los datos obtenidos en: frenado (entre 1 y 10 con dos decimales), contaminación (entre 20 y 50 con dos decimales), frenado (apto o no) y luces (apto o no). Este informe debe reflejar los datos del trabajador que lo ha realizado, del vehículo y del propietario/a.

A parte queremos obtener:
- El trabajador que más gana sin ser responsable.
- El salario medio de todos los trabajadores que no son responsables.
- El salario medio de todos los trabajadores agrupados por especialidad.
- La el trabajador/a con menos antigüedad.
- Trabajadores ordenados por especialidad y ordenados por antigüedad.

Se pide:

Según tengas pendiente:
- Carga de datos mediante CSVS y backup usando JSON.
- Implementación mediante SQL y SpringData, realizando el diagrama de clases y explicando el paso a tablas de la solución.
- Implementación mediante NoSQL usando MongoDB y SpringData MongoDB, realizando el diagrama de clases y explicando la obtención de colecciones existentes.
- API REST Segura para gestión de empleados, y citas y sistema de notificaciones.

Obligatoriamente:
- Caché de entidades relevantes.
- Obtención de tiempo real mediante notificaciones de las citas existentes (especialmente las de hoy). Se deberá lanzar al comienzo del programa y "reaccionar" a los cambios que se realicen posteriormente.
- Enfoque Railway Oriented Programming.
- Test de todos los elementos usados.

NOTA: Se debe mostrar el funcionamiento de todas las restricciones impuestas en este enunciado y consultas propuestas.

Entrega: Documentación, repositorio, vídeo en Youtube, entrevista en clase. 14 de junio en clase.