

## Sistema gestor de citas:

Se realizará mediante el actor <<Trabajador>>

### Requisitos funcionales:

RF-01: Autenticación/Login de Trabajador.

RF-02: Crear cita seleccionando un inspector y añadir datos del vehículo y su propietario.

RF-03: Eliminar cita de la base de datos.

RF-04: Actualizar cita de la base de datos.

RF-05.1: Exportar la cita a JSON

RF-05.2: Exportar la cita a HTML

RF-06: Exportar todas las citas a JSON

### Requisitos no funcionales:

RNF-01: La creación de una cita se realiza mediante un inspector con menos de 5 citas asociadas, y se deberán introducir los datos del vehículo y su propietario, generándose un informe el cuál estará vacío con el atributo "no apto".

RNF-02: La búsqueda de la cita se podrá realizar mediante la matrícula del vehículo, tipo de vehículo y la fecha de la última revisión.

RNF-03: Al actualizar una cita, se selecciona si es apto o no apto, asignando los datos del informe, y actualizando el informe que se encuentra vacío. y podremos elegir el formato de exportación a JSON o HTML.

### Requisitos de información:

RI-01: La base de datos está creada en MariaDB.

RI-02: La lógica de la aplicación está realizada con Kotlin.

RI-03: La cita tendrá la clave primaria auto numérica.

RI-04: La cita tendrá las fechas de matriculación y última revisión en formato ISO.

RI-05: No se atenderán más de 8 citas en el mismo intervalo de 30 minutos.