# ·RF:

·RF-01: obtener todos los informes.

·RF-02: añadir un nuevo informe.

·RF-03: editar un informe seleccionado.

·RF-04: borrar un informe seleccionado.

·RF-05: filtrar los informes según la fecha en la que se produjo (es un intervalo de tiempo).

·RF-06: filtrar los informes según la matrícula introducida.

·RF-07: filtrar los informes según el tipo del vehículo seleccionado.

·RF-08: exportar informe, asociado a una cita seleccionada, a JSON.

·RF-9: exportar informe, asociado a una cita seleccionada, a HTML.

·RF-10: mostrar ventana de ayuda.

·RF-11: exportar trabajadores a un fichero CSV.

·RF-12: exportar informes a un fichero JSON.

·RF-13: obtener todos los trabajadores.

·RF-14: obtener todos los vehículos.

·RF-15: añadir un nuevo vehículo.

·RF-16: editar un vehículo seleccionado.

·RF-17: borrar un vehículo seleccionado.

·RF-18: obtener todos los propietarios.

·RF-19: añadir un nuevo propietario.

·RF-20: editar el propietario seleccionado.

·RF-21: borrar el propietario seleccionado.

·RF-22: añadir una nueva estación.

·RF-23: editar una estación seleccionada.

·RF-24: borrar una estación seleccionada.

·RF-25: filtrar los vehículos según la matrícula introducida.

·RF-26: filtrar los vehículos según el tipo del motor.

·RF-27: filtrar los propietarios según el dni introducido.

·RF-28: obtener una cita según su id.

·RF-29: obtener un vehículo según su id.

·RF-30: obtener un propietario según su id.

·RF-31: obtener un trabajador según su id.

·RF-32: obtener una estación según su id.

·RF-33: crear un nuevo trabajador.

·RF-34: editar un trabajador.

·RF-35: borrar un trabajador.

·RF-36: obtener un trabajador.

·RF-37: realizar inspección de vehículo.

# ·RNF:

·RNF-01: aplicar un estilo similar a la página web y a la alicación.

·RNF-02: utilizar una BBDD relacional y que sea MariaDB.

·RNF-03: hacer la aplicación con JavaFX.

# ·RI:

·Estación ITV:

-idEstacion: es el identificador único de la estación ITV, es un Long.

-nombre: es una variable del tipo String, no puede estar vacía.

-dirección: es una variable del tipo String, no puede estar vacía.

-numeroTelefono: es una variable de tipo String, no puede estar vacía. Es un String ya que el número puede empezar con 0.

-correoElectrónico: es un String, no puede estar vacío.

·Trabajador:

-idTrabajador: es el identificador único del trabajador, es un Long, no puede estar vacia.

-idEstacion: es el id de la estación a la que pertenece el trabajador, es un Long, no puede estar vacía.

-nombre: es el nombre real del trabajador, es un String, no puede estar vacía.

-telefono: es un String, no puede estar vacio.

-email: es un string, no puede estar vacío y debe ser único.

-nombreUsuario: es el nombre de usuario asociado al trabajador, es un String y no puede estar vacío.

-contraseña: es la contraseña de la cuenta de usuario asociada al trabajador, es un String que se guardará cifrado en la BBDD, no puede estar vacía.

-fechaContratacion: es un LocalDate, no puede estar vacía.

-especialidad: es una List<Especialidad>, Especialidad es un Enum, no puede estar vacía.

-salario: es un Double, será un campo calculado a partir de la/s epecialidad/es, más los años de antigüedad.

-idResponsable: es el identificador del trabajador que es responsable del resto, es un Long, no puede estar vacío.

-informes: es un List<Informe>, que sirve para tener la citas/informes almacenadas, puede estar vacio.

·Vehiculo:

-matricula: es un String, el identificador único del vehículo y no puede estar vacío.

-marca: es un String, no puede estar vacía.

-modelo: es un String, no puede estar vacío.

-fechaMatriculacion: es un LocalDate, no puede estar vacío.

-fechaUltimaRevision: es un LocalDateTime, puede estar vacío.

-tipoMotor: es un Enum, no puede estar vacío y no se podrá tomar un valor diferente.

-tipoVehiculo: es un Enum, no puede estar vacío y no se podrá tomar un valor diferente.

-propietario: es el Propietario del vehículo, no puede estar vacío.

·Propietario:

-dni: es el identificador único del propietario, es un String, no puede estar vacío.

-nombre: es un String, no puede estar vacío.

-apellidos: es un String, no puede estar vacío.

-telefono: es un String, no puede estar vacio.

-email: es un String, no puede estar vacío.

·Informe:

-idInforme: es un Long, no puede estar vacio.

-fechaInicio: es un LocalDateTime, no puede estar vacio.

-fechaFinal: es un LocalDateTime, no puede estar vacio.

-favorable: es un Enum que puede ser “apto” o “no apto”, no puede estar vacío.

-frenado: es un Double, que puede ir de 0.0 a 10.0, no puede estar vacío.

-contaminación: es un Double, que puede ir de 20.0 a 50.0, no puede estar vacio.

-interior: es un Enum que puede ser “apto” o “no apto”, no puede estar vacío.

-luces: es un Enum que puede ser “apto” o “no apto”, no puede estar vacío.

-idTrabajador: es el identificador del Trabajador que atendió la cita sobre la que se saca el informe, no puede estar vacío.

-vehiculo: es el Vehiculo que fue atendido en la cita, no puede estar vacío, además el Vehiculo ya contiene la información del Propietario, que tampoco puede estar vacío.

# ·Tecnologías utilizadas:

·Como IDE hemos utilizado IntelliJ.

·Como lenguaje de programación Kotlin.

·Como herramienta de front-end en programación, hemos usado JavaFX junto con Scene Builder.

·Para la creación de la página web se a usado HTML, CSS y JavaScript.

·Se ha utilizado VisualStudioCode para la creación de la página web.

·Se ha utilizado una BBDD MariaDB, con el lenguaje MySQL.

·Se ha utilizado XAMPP para levantar el servidor de MariaDB.

·Se ha utilizado DataGrip para la creación de la BBDD.

·Se ha utilizado Draw.io para la creación de los distintos diagramas pedidos.

·Se ha utilizado GitHub como sistema de control de versiones.