



# MANUAL DE USO **RIAB**

Sosa Candela, Mejia Sol, Mejia Micaela

# MODELOS

RIAB incluye una BD. llamada riab, que contiene 4 modelos:  
**modelHistorial**, **modelMascotas**, **modelRescatista** y  
**modelPersonas**.

## HISTORIAL

- ID
- CASTRADO
- OPERADO
- DISCAPACIDAD
- RABIA
- ENFERMEDADES
- DESPARASITADO

## MASCOTAS

- ID
- NOMBRE\_APODO
- ESPECIE
- RAZA
- COLOR
- ESTADO\_SALUD
- ANIO\_NACIMIENTO

---

## RESCATISTA

- DNI
- NOMBRE
- APELLIDO
- TELEFONO
- EMAIL
- DIRECCIÓN
- GÉNERO
- PASSW

## PERSONAS

- DNI
  - NOMBRE
  - APELLIDO
  - TELEFONO
  - EMAIL
  - DIRECCIÓN
  - GÉNERO
-

# CONTROLADORES

RIAB incluye una 4 controladores: histoControlador, mascotasControlador, personasControlador y rescatistasControlador. Cada uno contorá con sus funciones correspondientes.

## HISTORIAL

- OBTENETODOS : En esta función obtendremos todos los historiales médicos que se encuentren en la BD y también utilizando el método **findAll()**.
- OBTENER: Obtendremos un historial médico buscandolo por ID a través del método **findByPk()**.
- CREAR: Crearemos un nuevo historial médico respetando los campos puestos en el modelo del mismo y sus respectivas validaciones. Utilizamos el método **create()**.
- ACTUALIZAR: Actualizaremos los historial para cuando se cambie algo utilizando el método **update()**.
- BORRAR: Borraremos un historial, buscandolo primero por su ID con el método **findOne()** y luego terminamos con la utilización del método **destroy()**.

## MASCOTAS

- OBTENERMASCOTAS: Obtendremos a todas las mascotas utilizando el método **finAll()**.

- OBTENERMASCOTASID: Obtendremos a las mascotas pero buscandolas por ID, utilizando el método **findByPk()**.
- CREARMASCOTAS: Crearemos a las mascotas respetando los campos puestos en el modelo y sus respectivas validaciones. Utilizamos el método **create()**.
- ACTUALIZARMASCOTAS: Actualizaremos la lista de mascotas utilizando el método **update()**.
- BORRARMASCOTAS: La función del borrado de mascotas contendrá la utilización del método **destroy()**, pero antes debemos encontrar el id de la mascota con el método **findOne()**.

## PERSONAS

- OBTENETODOS: Obtendremos a todas las personas con el método **findAll()**.
- OBTENER: En esta función obtendremos a las personas per buscandolas por DNI con el método **findByPk()**.
- CREAR: Crearemos a las personas respetando los campos proporcionados a su modelo y validaciones colocadas. Utilizamos el método **create()**.
- ACTUALIZAR: Actualizaremos la lista de personas con el método **update()**.
- BORRAR: Borraremos a las personas buscandolas primero por su DNI con el método **findOne()** y luego la borraremos con el método **destroy()**.

## RESCATISTAS

- OBTENETODOS: Obtendremos a todos los rescatistas utilizando el método **findAll()** para buscar a todos.
- OBTENER: Obtendremos a una persona específica buscandola por su DNI con el método **findByPk()**.
- CREAR: Crearemos a las personas respetando los campos puestos en su modelo y sus correspondientes validaciones, con la utilización del método **create()**.
- ACTUALIZAR: Actualizaremos a los rescatistas utilizando el método **update()**.
- BORRAR: Borraremos a los rescatistas primero buscandola por su DNI con la utilización del método **findOne()** y luego eliminaremos con el método **destroy()**.
- LOGIN: En esta función se utilizara el campo passw del modelo Rescatista, ya que con este campo se proporcionará el ingreso como rescatista.

# RUTAS

RIAB cuenta con un archivo JS donde se encuentran las rutas de cada uno de los modelos hablados anteriormente. Luego conectamos las Rutas con sus despectivos llamados.