DICCIONARIO DE API REST

/\*\*

\*

\* Metodo GET que nos devuelve todos las variantes cargadas en la Tabla

\* Parametrica "Variante"

\*/

@GetMapping("/listVariantes")

public List<Variante> getAllVariantes() {

List<Variante> list = varianteService.findAll();

return list;

}

/\*\*

\*

\* Metodo GET que nos devuelve los detalles de una variante a partir de su

\* nombre

\*/

@GetMapping("/variante")

public ResponseEntity<Variante> getVariante(@RequestParam String nombre) {

Variante variante = varianteService.findByNombre(nombre);

if (variante == null) {

return ResponseEntity.notFound().build();

} else {

return ResponseEntity.ok(variante);

}

}

/\*\*

\*

\* Metodo POST que inserta una variante nueva

\*/

@PostMapping("/addVariante")

public ResponseEntity<Variante> insertarVariante(@RequestParam("id") String id,

@RequestParam("nombre") String nombre, @RequestParam("descripcion") String descripcion,

@RequestParam("costo") String costo) {

Variante variante = varianteService.save(id, nombre, costo, descripcion);

if (variante == null) {

return ResponseEntity.notFound().build();

} else {

return ResponseEntity.ok(variante);

}

}

/\*\*

\*

\* Metodo POST que guarda un nuevo Automovil en la Base de datos

\*

\*/

@PostMapping(path = "/guardarAuto")

public ResponseEntity<Automovil> guardar(@RequestParam("id") String id, @RequestParam("nombre") String nombre,

@RequestParam("opcionales") List<String> opcionales) {

Automovil auto = autoService.save(id, nombre, opcionales);

if (auto == null) {

return ResponseEntity.notFound().build();

} else {

return ResponseEntity.ok(auto);

}

}

/\*\*

\*

\* Metodo PUT que actualiza un nuevo Automovil en la Base de datos

\*

\*/

@PutMapping(path = "/actualizaAuto")

public ResponseEntity<Automovil> actualizar(@RequestParam("id") String id, @RequestParam("nombre") String nombre,

@RequestParam("opcionales") List<String> opcionales) {

Automovil auto = autoService.update(id, nombre, opcionales);

if (auto == null) {

return ResponseEntity.notFound().build();

} else {

return ResponseEntity.ok(auto);

}

}

/\*\*

\*

\* Metodo GET que nos devuelve los detalles de un auto a partir de su nombre

\*/

@GetMapping("/consultarAuto")

public ResponseEntity<Automovil> getAuto(@RequestParam String nombre) {

Automovil automovil = autoService.findByNombre(nombre);

if (automovil == null) {

return ResponseEntity.notFound().build();

} else {

return ResponseEntity.ok(automovil);

}

}

/\*\*

\*

\* Metodo DELETE que nos borra un Auto a partir de su nombre

\*/

@DeleteMapping("/borrarAuto")

public ResponseEntity<String> deleteAuto(@RequestParam String nombre) {

Automovil automovil = autoService.findByNombre(nombre);

// Validamos si el auto existe en la BD

if (automovil == null) {

return ResponseEntity.notFound().build();

} else {

autoService.deleteByNombre(automovil);

return ResponseEntity.ok(nombre);

}

}

}