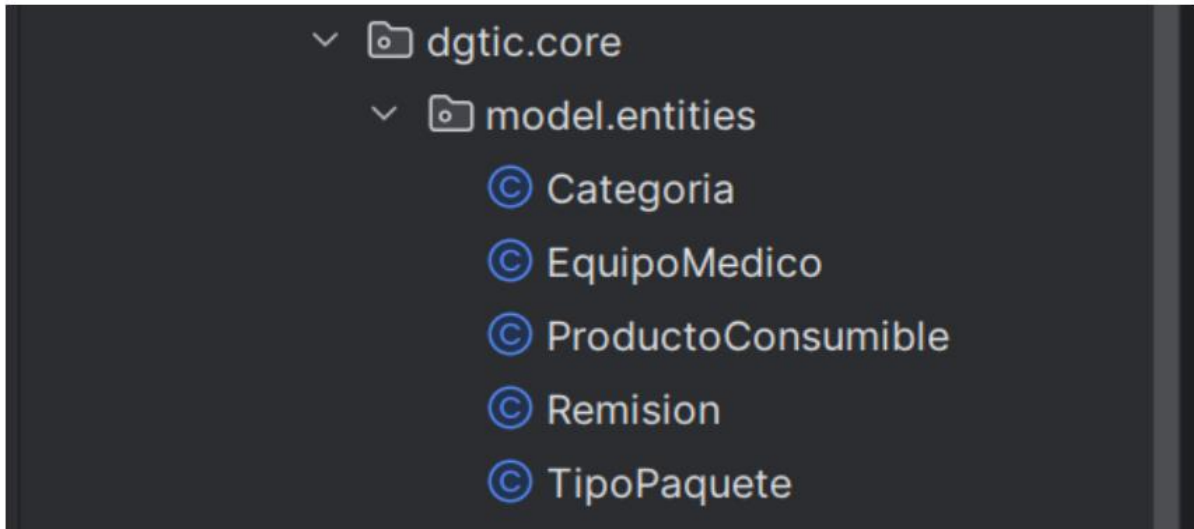


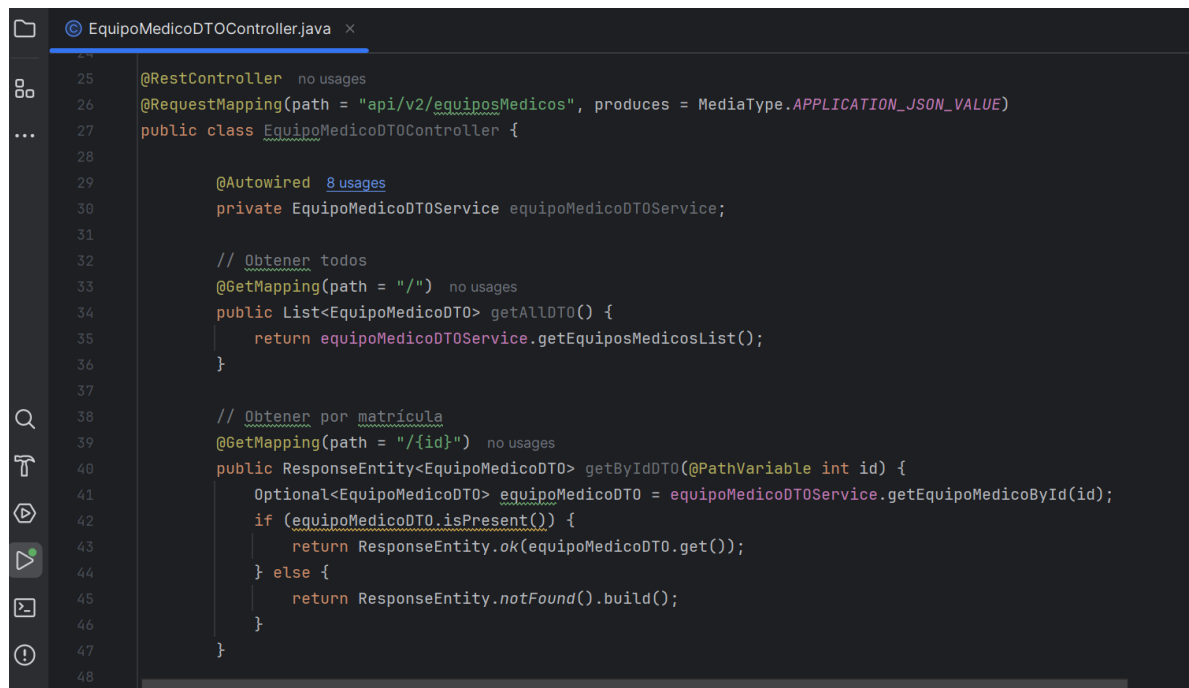
1. Selecciona cuatro tablas de tu proyecto actual.

Se crearon las siguientes entidades con los nombres :



2. Implementa operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para cada una de estas tablas

EquipoMedicoDTOController --- Api principal



```
EquipoMedicoDTOController.java x
27 public class EquipoMedicoDTOController {
48
49     @PostMapping(path = "/"*) no usages
50     public ResponseEntity<EquipoMedicoDTO> createEquipoMedicoDTO(
51         @Valid @RequestBody EquipoMedicoDTO equipoDTO) throws ParseException, URISyntaxException, CategoriaNoExisteExcepcion {
52         EquipoMedicoDTO equipoMedicoDTO = equipoMedicoDTOService.createEquipoMedico(equipoDTO);
53         URI location = new URI(str: "/api/v2/equiposMedicos/" + equipoMedicoDTO.getId());
54         return ResponseEntity.created(location).body(equipoMedicoDTO);
55     }
56
57     @PutMapping(path =("/{id}*)) no usages
58     @
59     public ResponseEntity<EquipoMedicoDTO> updateEquipoDTO(@PathVariable int id, @RequestBody EquipoMedicoDTO equipo) throws ParseException {
60         equipo.setId(id);
61         return ResponseEntity.ok(equipoMedicoDTOService.updateEquipoMedico(equipo));
62     }
}
```

```
EquipoMedicoDTOController.java x
27 public class EquipoMedicoDTOController {
63
64     @PatchMapping(path =("/{id}*)) no usages
65     public ResponseEntity<EquipoMedicoDTO> updatePartialEquipoDTO(@PathVariable int id, @RequestBody EquipoMedicoDTO equipo) throws ParseException {
66         Optional<EquipoMedicoDTO> equipoDB = equipoMedicoDTOService.getEquipoMedicoById(id);
67         if (equipoDB.isPresent()) {
68             EquipoMedicoDTO modificable = equipoDB.get();
69             if (equipo.getNombreEquipo() != null) modificable.setNombreEquipo(equipo.getNombreEquipo());
70             if (equipo.getExistencias() != 0) modificable.setExistencias(equipo.getExistencias());
71             if (equipo.getCategoria() != null) modificable.setCategoria(equipo.getCategoria());
72             if (equipo.getPaquetes() != null) modificable.setPaquetes(equipo.getPaquetes());
73             return ResponseEntity.ok(equipoMedicoDTOService.updateEquipoMedico(modificable));
74         } else {
75             return ResponseEntity.notFound().build();
76         }
77     }
78
79     @DeleteMapping(path =("/{id}*)) no usages
80     public ResponseEntity<?> deleteEquipo(@PathVariable int id) {
81         if (equipoMedicoDTOService.deleteEquipoMedico(id)) {
82             return ResponseEntity.ok().build();
83         } else {
84             return ResponseEntity.notFound().build();
85         }
86     }
}
```

## ProductoConsumibleController

```
ProductoConsumibleController.java x
21 @RestController no usages
22 @RequestMapping(path = "/api/v2/productosConsumibles", produces = MediaType.APPLICATION_JSON_VALUE)
23 public class ProductoConsumibleController {
24
25     @Autowired 7 usages
26     private ProductoConsumibleService productoConsumibleService;
27
28     // Obtener todos
29     @GetMapping(path = "/"*)
30     public List<ProductoConsumible> getAll() {
31         return productoConsumibleService.buscarProductoConsumible();
32     }
33
34     // Obtener por matricula
35     @GetMapping(path =("/{id}*)) no usages
36     public ResponseEntity<ProductoConsumible> getById(@PathVariable int id) {
37         Optional<ProductoConsumible> productoConsumible = productoConsumibleService.buscarProductoConsumibleId(id);
38         if (productoConsumible.isPresent()) {
39             return ResponseEntity.ok(productoConsumible.get());
40         } else {
41             return ResponseEntity.notFound().build();
42         }
43     }
}
```

```

22 public class ProductoConsumibleController {
23
24     @PostMapping(path = "/" no usages
25     public ResponseEntity<ProductoConsumible> createProductoConsumible(
26         @Valid @RequestBody ProductoConsumible productoConsumible) throws ParseException, URISyntaxException, CategoriaNoExisteExcepcion
27     {
28         ProductoConsumible productoNuevo = productoConsumibleService.guardarProductoConsumible(productoConsumible);
29         URI location = new URI("http://localhost:8080/api/v2/productosConsumibles/" + productoNuevo.getIdProductoConsumible());
30         return ResponseEntity.created(location).body(productoNuevo);
31     }
32
33     @PutMapping(path =("/{id}") no usages
34     @
35     public ResponseEntity<ProductoConsumible> updateProductoConsumible(@PathVariable int id, @RequestBody ProductoConsumible producto) throws
36     {
37         producto.setIdProductoConsumible(id);
38         return ResponseEntity.ok(productoConsumibleService.actualizarProductoConsumible(producto));
39     }
40
41 }

```

```

22 public class ProductoConsumibleController {
23
24     @PatchMapping(path =("/{id}") no usages
25     public ResponseEntity<ProductoConsumible> updatePartialProducto(@PathVariable int id, @RequestBody ProductoConsumible producto) throws ParseException, URISyntaxException, CategoriaNoExisteExcepcion
26     {
27         Optional<ProductoConsumible> productoDB = productoConsumibleService.buscarProductoConsumibleId(id);
28         if (productoDB.isPresent()) {
29             ProductoConsumible modificable = productoDB.get();
30             if (producto.getNombreProducto() != null) modificable.setNombreProducto(producto.getNombreProducto());
31             if (producto.getExistencias() != 0) modificable.setExistencias(producto.getExistencias());
32             if (producto.getDescripcion() != null) modificable.setDescripcion(producto.getDescripcion());
33             //if (producto.getPaquetes() != null) modificable.setPaquetes(producto.getPaquetes());
34             return ResponseEntity.ok(productoConsumibleService.actualizarProductoConsumible(modificable));
35         } else {
36             return ResponseEntity.notFound().build();
37         }
38     }
39
40     @DeleteMapping(path =("/{id}") no usages
41     public ResponseEntity<?> deleteProducto(@PathVariable int id) {
42         if (productoConsumibleService.borrarProductoConsumible(id)) {
43             return ResponseEntity.ok().build();
44         } else {
45             return ResponseEntity.notFound().build();
46         }
47     }
48 }

```

## CategoriaController

```

14
15 @RestController no usages
16 @RequestMapping(path = "/api/categorias",
17     produces = MediaType.APPLICATION_JSON_VALUE)
18 public class CategoriaController {
19
20     @Autowired 9 usages
21     private CategoriaService categoriaService;
22
23     @GetMapping(path = "/"
24     public ResponseEntity<List<Categoria>> getAll() {
25         return ResponseEntity.ok(categoriaService.buscarCategorias());
26     }
27
28     @GetMapping(path =("/{id}") no usages
29     public ResponseEntity<Categoria> getById(@PathVariable("id") int id) {
30         Optional<Categoria> categoria = categoriaService.buscarCategoriaId(id);
31         if (categoria.isPresent()) {
32             return ResponseEntity.ok(categoria.get());
33         } else {
34             return ResponseEntity.notFound().build();
35         }
36     }
37
38 }

```

```
18 public class CategoriaController {
38
39     @PostMapping(path = "/") no usages
40     public ResponseEntity<Categoria> createCategoria(@RequestBody Categoria categoria) throws URISyntaxException {
41         Categoria categoriaNueva = categoriaService.guardarCategoria(categoria);
42         URI location = new URI(str: "/api/categorias/" + categoriaNueva.getIdCategoria());
43         return ResponseEntity.created(location).body(categoriaNueva);
44     }
45
46     @PutMapping(path =("/{id}") no usages
47     public ResponseEntity<Categoria> updateFullCategoria(@RequestBody Categoria categoria, @PathVariable("id") int id) throws URISyntaxException {
48         Optional<Categoria> categoriaDB = categoriaService.buscarCategoriaId(id);
49         if (categoriaDB.isPresent()) {
50             return ResponseEntity.ok(categoriaService.actualizarCategoria(categoria));
51         } else {
52             Categoria categoriaNueva = categoriaService.guardarCategoria(categoria);
53             URI location = new URI(str: "/api/categorias/" + categoriaNueva.getIdCategoria());
54             return ResponseEntity.created(location).body(categoriaNueva);
55         }
56     }
57 }
```

```
18 public class CategoriaController {
57
58     @PatchMapping(path =("/{id}") no usages
59     public ResponseEntity<Categoria> updatePartialCategoria(@RequestBody Categoria categoria, @PathVariable("id") int id) throws URISyntaxException {
60         Optional<Categoria> categoriaDB = categoriaService.buscarCategoriaId(id);
61         if (categoriaDB.isPresent()) {
62             Categoria categoriaToUpdate = categoriaDB.get();
63             if (categoria.getNombreCategoria() != null) {
64                 categoriaToUpdate.setNombreCategoria(categoria.getNombreCategoria());
65             }
66             return ResponseEntity.ok(categoriaService.actualizarCategoria(categoriaToUpdate));
67         } else {
68             return ResponseEntity.notFound().build();
69         }
70     }
71
72     @DeleteMapping(path =("/{id}") no usages
73     public ResponseEntity<Boolean> deleteAlumno(@PathVariable("id") int id) {
74         if (categoriaService.borrarCategoria(id)) {
75             return ResponseEntity.ok().build();
76         } else {
77             return ResponseEntity.notFound().build();
78         }
79     }
80 }
```

Run SpringProyectoFinalApplication

## TipoPaqueteController

```
14
15 @RestController no usages
16 @RequestMapping(path = "/api/tipoPaquete",
17     produces = MediaType.APPLICATION_JSON_VALUE)
18 public class TipoPaqueteController {
19
20     @Autowired 0 usages
21     private TipoPaqueteService tipoPaqueteService;
22
23     @GetMapping(path = "/")
24     public ResponseEntity<List<TipoPaquete>> getAll() {
25         return ResponseEntity.ok(tipoPaqueteService.buscarTipoPaquete());
26     }
27
28     @GetMapping(path =("/{id}") no usages
29     public ResponseEntity<TipoPaquete> getById(@PathVariable("id") int id) {
30         Optional<TipoPaquete> tipoPaquete = tipoPaqueteService.buscarTipoPaqueteId(id);
31         if (tipoPaquete.isPresent()) {
32             return ResponseEntity.ok(tipoPaquete.get());
33         } else {
34             return ResponseEntity.notFound().build();
35         }
36     }
37 }
```

```
TipoPaqueteController.java x
18 public class TipoPaqueteController {
19
20     @PostMapping(path = "/") no usages
21     public ResponseEntity<TipoPaquete> createTipoPaquete(@RequestBody TipoPaquete tipoPaquete) throws URISyntaxException {
22         TipoPaquete tipoPaqueteNuevo = tipoPaqueteService.guardarTipoPaquete(tipoPaquete);
23         URI location = new URI(str: "/api/tipoPaquetes/" + tipoPaqueteNuevo.getIdTipoPaquete());
24         return ResponseEntity.created(location).body(tipoPaqueteNuevo);
25     }
26
27     @PutMapping(path =("/{id}") no usages
28     public ResponseEntity<TipoPaquete> updateFullTipoPaquete(@RequestBody TipoPaquete tipoPaquete, @PathVariable("id") int id) throws URISyntaxException {
29         Optional<TipoPaquete> tipoPaqueteDB = tipoPaqueteService.buscarTipoPaqueteId(id);
30         if (tipoPaqueteDB.isPresent()) {
31             return ResponseEntity.ok(tipoPaqueteService.actualizarTipoPaquete(tipoPaquete));
32         } else {
33             TipoPaquete tipoPaqueteNuevo = tipoPaqueteService.guardarTipoPaquete(tipoPaquete);
34             URI location = new URI(str: "/api/categorias/" + tipoPaqueteNuevo.getIdTipoPaquete());
35             return ResponseEntity.created(location).body(tipoPaqueteNuevo);
36         }
37     }
38 }
```

```
TipoPaqueteController.java x
18 public class TipoPaqueteController {
19
20     @PatchMapping(path =("/{id}") no usages
21     public ResponseEntity<TipoPaquete> updatePartialTipoPaquete(@RequestBody TipoPaquete tipoPaquete, @PathVariable("id") int id) throws URISyntaxException {
22         Optional<TipoPaquete> tipoPaqueteDB = tipoPaqueteService.buscarTipoPaqueteId(id);
23         if (tipoPaqueteDB.isPresent()) {
24             TipoPaquete tipoPaqueteToUpdate = tipoPaqueteDB.get();
25             if (tipoPaquete.getNombrePaquete() != null) {
26                 tipoPaqueteToUpdate.setNombrePaquete(tipoPaquete.getNombrePaquete());
27             }
28             return ResponseEntity.ok(tipoPaqueteService.actualizarTipoPaquete(tipoPaqueteToUpdate));
29         } else {
30             return ResponseEntity.notFound().build();
31         }
32     }
33
34     @DeleteMapping(path =("/{id}") no usages
35     public ResponseEntity<Boolean> deleteAlumno(@PathVariable("id") int id) {
36         if (tipoPaqueteService.borrarTipoPaquete(id)) {
37             return ResponseEntity.ok().build();
38         } else {
39             return ResponseEntity.notFound().build();
40         }
41     }
42 }
```

3. Se pondrá a prueba el funcionamiento de las diferentes apis utilizando la herramienta postman

## Pruebas de EquipoMedicoController ---- Obtener Todos

HTTP proyecto / New Request Save Share

GET http://127.0.0.1:8080/api/v2/equiposMedicos/ Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

Body Cookies Headers (5) Test Results 200 OK • 133 ms • 2.39 KB • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔗 📄 🔍

```
1 [
2   {
3     "id": 1,
4     "nombreEquipo": "Shaver",
5     "existencias": 8,
6     "categoria": "Equipos Artroscopicos",
7     "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro, Paquete Artroscopia Tobillo"
8   },
9   {
10    "id": 2,
11    "nombreEquipo": "Bomba de Irrigacion",
12    "existencias": 6,
13    "categoria": "Equipos Insuflacion y lavado",
14    "paquetes": "Paquete Artroscopia Hombro"
15  },
16  {
17    "id": 3,
18    "nombreEquipo": "Maquina De Anestecia",
19    "existencias": 6,
```

## Pruebas de EquipoMedicoController ---- Obtener Por ID

HTTP proyecto / New Request Save Share

GET http://127.0.0.1:8080/api/v2/equiposMedicos/1 Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

Body Cookies Headers (5) Test Results 200 OK • 12 ms • 346 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔗 📄 🔍

```
1 {
2   "id": 1,
3   "nombreEquipo": "Shaver",
4   "existencias": 8,
5   "categoria": "Equipos Artroscopicos",
6   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro, Paquete Artroscopia Tobillo"
7 }
```

## Pruebas de EquipoMedicoController ---- Crear Equipo

The screenshot shows a REST client interface with a "New Request" tab. The request method is "POST" and the URL is "http://127.0.0.1:8080/api/v2/equiposMedicos/". The request body is a JSON object with the following fields: "nombreEquipo", "existencias", "categoria", and "paquetes". The response status is "201 Created", and the response body is a JSON object with the same fields plus an "id" field.

```
1 {
2   "nombreEquipo": "Equipo Test",
3   "existencias": 5,
4   "categoria": "Equipos Artroscopicos",
5   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro, Paquete Artroscopia Tobillo"
6 }
```

```
1 {
2   "id": 27,
3   "nombreEquipo": "Equipo Test",
4   "existencias": 5,
5   "categoria": "Equipos Artroscopicos",
6   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro, Paquete Artroscopia Tobillo"
7 }
```

## Pruebas de EquipoMedicoController ---- Actualizacion Completa de EquipoMedico

The screenshot shows a REST client interface with a "New Request" tab. The request method is "PUT" and the URL is "http://127.0.0.1:8080/api/v2/equiposMedicos/27". The request body is a JSON object with the following fields: "nombreEquipo", "existencias", "categoria", and "paquetes". The response status is "200 OK", and the response body is a JSON object with the same fields plus an "id" field.

```
1 {
2   "nombreEquipo": "Equipo Test Actualizado Completo",
3   "existencias": 6,
4   "categoria": "Equipos Artroscopicos",
5   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro"
6 }
```

```
1 {
2   "id": 27,
3   "nombreEquipo": "Equipo Test Actualizado Completo",
4   "existencias": 6,
5   "categoria": "Equipos Artroscopicos",
6   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro"
7 }
```



## Pruebas de EquipoMedicoController ---- Actualizacion Parcial de EquipoMedico

HTTP proyecto / New Request Save Share

**PATCH** ▼  Send ▼

Params Authorization Headers (9) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {
2   "nombreEquipo": "Equipo Test Actualizado Parcial",
3   "existencias": 1
4 }
```

Body Cookies Headers (5) Test Results ↺ 200 OK • 22 ms • 343 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize **JSON** ▼ 🔗 📄 🔍

```
1 {
2   "id": 27,
3   "nombreEquipo": "Equipo Test Actualizado Parcial",
4   "existencias": 1,
5   "categoria": "Equipos Artroscopicos",
6   "paquetes": "Paquete Artroscopia Rodilla, Paquete Artroscopia Hombro"
7 }
```

## Pruebas de EquipoMedicoController ---- Eliminar EquipoMedico

**DELETE** ▼  Send ▼

Params Authorization Headers (7) **Body** • Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results ↺ 200 OK • 23 ms • 123 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize **Text** ▼ 🔗 📄 🔍

```
1
```



## Pruebas de ProductoConsumibleController ---- Obtener Todos

GET  Send

Params Authorization Headers (7) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK • 312 ms • 1.09 KB

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "idProductoConsumible": 1,
4     "nombreProducto": "Cuchillas artroscópicas",
5     "descripcion": "hace incisiones precisas en la articulación permitiendo la entrada de
6       instrumentos",
7     "existencias": 5
8   },
9   {
10    "idProductoConsumible": 2,
11    "nombreProducto": "Anclas de sutura",
12    "descripcion": "se implantan en el hueso para anclar los hilos de sutura",
13    "existencias": 5
14  },
15  {
16    "idProductoConsumible": 3,
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de ProductoConsumibleController ---- Obtener por Id

GET  Send

Params Authorization Headers (5) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK • 26 ms • 361 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idProductoConsumible": 3,
3   "nombreProducto": "Vaporizador de radiofrecuencia",
4   "descripcion": "utiliza radiofrecuencia para vaporizar tejido no deseado o dañado en la articulación.",
5   "existencias": 4
6 }
```

## Pruebas de ProductoConsumibleController ---- Crear ProductoConsumible

POST http://127.0.0.1:8080/api/v2/productosConsumibles/ Send

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreProducto": "Producto de prueba",
3   "descripcion": "Se utiliza para realizar pruebas en la api ",
4   "existencias": 4
5 }
```

Body Cookies Headers (6) Test Results 201 Created • 213 ms • 352 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idProductoConsumible": 10,
3   "nombreProducto": "Producto de prueba",
4   "descripcion": "Se utiliza para realizar pruebas en la api ",
5   "existencias": 4
6 }
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de ProductoConsumibleController ---- Actualizacion Completa de ProductoConsumible

PUT http://127.0.0.1:8080/api/v2/productosConsumibles/10 Send

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreProducto": "Producto de prueba Actualizacion Completa",
3   "descripcion": "Se utiliza para realizar pruebas en la api ",
4   "existencias": 7
5 }
```

Body Cookies Headers (5) Test Results 200 OK • 29 ms • 328 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idProductoConsumible": 10,
3   "nombreProducto": "Producto de prueba Actualizacion Completa",
4   "descripcion": "Se utiliza para realizar pruebas en la api ",
5   "existencias": 7
6 }
```

## Pruebas de ProductoConsumibleController ---- Actualizacion Parcial de ProductoConsumible

**PATCH** ⌵  **Send** ⌵

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ⌵ Beautify

```
1 {
2   "nombreProducto": "Producto de prueba Actualizacion Parcial",
3   "existencias": 7
4 }
```

Body Cookies Headers (5) Test Results ↺ **200 OK** • 16 ms • 327 B •

Pretty Raw Preview Visualize **JSON** ⌵

```
1 {
2   "idProductoConsumible": 10,
3   "nombreProducto": "Producto de prueba Actualizacion Parcial",
4   "descripcion": "Se utiliza para realizar pruebas en la api ",
5   "existencias": 7
6 }
```

## Pruebas de ProductoConsumibleController ---- Eliminar ProductoConsumible

**DELETE** ⌵  **Send** ⌵

Params Authorization Headers (7) **Body** ● Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results ↺ **200 OK** • 24 ms • 123 B •

Pretty Raw Preview Visualize **Text** ⌵

```
1
```

## Pruebas de CategoriaController ---- Obtener Todos

GET ▼ http://127.0.0.1:8080/api/categorias/ Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

---

Body Cookies Headers (5) Test Results ↺ 200 OK • 10 ms • 574 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ▼ 🔗 📄 🔍

```
1 {  
10   }  
11   "idCategoria": 1,  
12   "nombreCategoria": "Equipos Artroscopicos"  
13 }  
14 {  
15   "idCategoria": 2,  
16   "nombreCategoria": "Equipos de Anestecia"  
17 }  
18 {  
19   "idCategoria": 13,  
20   "nombreCategoria": "Equipos de Anestecia Test"  
21 }  
22 }
```

🔗 📄 🔍 Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de CategoriaController ---- Obtener Por Id

GET ▼ http://127.0.0.1:8080/api/categorias/1 Send ▼

Params Authorization Headers (5) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

---

Body Cookies Headers (5) Test Results ↺ 200 OK • 7 ms • 223 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ▼ 🔗 📄 🔍

```
1 {  
2   "idCategoria": 1,  
3   "nombreCategoria": "Equipos Artroscopicos"  
4 }
```

## Pruebas de CategoriaController ---- Crear Categoria

POST

http://127.0.0.1:8080/api/categorias/

Send

ParamsAuthorizationHeaders (9)Body ●ScriptsSettingsCookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   |   "nombreCategoria": "Categoria para probar "
3   }
```

BodyCookiesHeaders (6)Test Results201 Created • 14 ms • 260 B

PrettyRawPreviewVisualizeJSON

1 {
2 | "idCategoria": 14,
3 | "nombreCategoria": "Categoria para probar "
4 }

## Pruebas de CategoriaController ---- Actualizacion Completa de Categoria

PUT

http://127.0.0.1:8080/api/categorias/14

Send

ParamsAuthorizationHeaders (9)Body ●ScriptsSettingsCookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   |   "nombreCategoria": "Categoria Actualizacion Completa. "
3   }
```

BodyCookiesHeaders (5)Test Results200 OK • 75 ms • 237 B

PrettyRawPreviewVisualizeJSON

1 {
2 | "idCategoria": 14,
3 | "nombreCategoria": "Categoria Actualizacion Completa. "
4 }

PostbotRunnerStart ProxyCookiesVaultTrash

## Pruebas de CategoriaController ---- Actualizacion Parcial de Categoria

**PATCH** ▼  **Send** ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {  
2   "nombreCategoria": "Categoria Actualizacion Parcial. "  
3 }
```

Body Cookies Headers (5) Test Results ↺ **200 OK** • 11 ms • 236 B • ⋮

Pretty Raw Preview Visualize **JSON** ▼

```
1 {  
2   "idCategoria": 14,  
3   "nombreCategoria": "Categoria Actualizacion Parcial. "  
4 }
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de CategoriaController ---- Eliminacion de Categoria.

**DELETE** ▼  **Send** ▼

Params Authorization Headers (7) **Body** ● Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results ↺ **200 OK** • 19 ms • 123 B • ⋮

Pretty Raw Preview Visualize **Text** ▼

```
1
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de TipoPaqueteController ---- Obtener Todos

GET ⌵ http://127.0.0.1:8080/api/tipoPaquete/ Send ⌵

Params Authorization Headers (7) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results ⌵ 200 OK • 299 ms • 414 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1 {
2   {
3     "idTipoPaquete": 1,
4     "nombrePaquete": "Paquete Artroscopia Rodilla"
5   },
6   {
7     "idTipoPaquete": 2,
8     "nombrePaquete": "Paquete Artroscopia Hombro"
9   },
10  {
11    "idTipoPaquete": 3,
12    "nombrePaquete": "Paquete Artroscopia Tobillo"
13  },
14  {
15    "idTipoPaquete": 5,
16    "nombrePaquete": "Paquete Test2"
```

🔗 🏃 ▶ 🌐 🍪 🏠 🗑️

## Pruebas de TipoPaqueteController ---- Obtener Por ID

GET ⌵ http://127.0.0.1:8080/api/tipoPaquete/3 Send ⌵

Params Authorization Headers (7) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results ⌵ 200 OK • 25 ms • 229 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1 {
2   "idTipoPaquete": 3,
3   "nombrePaquete": "Paquete Artroscopia Tobillo"
4 }
```

🔗 🏃 ▶ 🌐 🍪 🏠 🗑️



## Pruebas de TipoPaqueteController ---- Crear TipoPaquete

POST http://127.0.0.1:8080/api/tipoPaquete/ Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombrePaquete": "Paquete De Pruebas TEST"
3 }
```

Body Cookies Headers (6) Test Results 201 Created • 108 ms • 261 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idTipoPaquete": 7,
3   "nombrePaquete": "Paquete De Pruebas TEST"
4 }
```

## Pruebas de TipoPaqueteController ---- Actualizar TipoPaquete

PUT http://127.0.0.1:8080/api/tipoPaquete/7 Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombrePaquete": "Paquete Actualizacon Completa"
3 }
```

Body Cookies Headers (5) Test Results 200 OK • 384 ms • 231 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idTipoPaquete": 7,
3   "nombrePaquete": "Paquete Actualizacon Completa"
4 }
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de TipoPaqueteController ---- Actualizacion Parcial de TipoPaquete

The screenshot shows a REST client interface with a PATCH request to `http://127.0.0.1:8080/api/tipoPaquete/7`. The request body is a JSON object: `{ "nombrePaquete": "Paquete Actualizacon Parcial" }`. The response is a 200 OK status with a response time of 11 ms and a body size of 230 B. The response body is a JSON object: `{ "idTipoPaquete": 7, "nombrePaquete": "Paquete Actualizacon Parcial" }`.

Params Authorization Headers (9) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "nombrePaquete": "Paquete Actualizacon Parcial"
3 }
```

Body Cookies Headers (5) Test Results 200 OK • 11 ms • 230 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "idTipoPaquete": 7,
3   "nombrePaquete": "Paquete Actualizacon Parcial"
4 }
```

Postbot Runner Start Proxy Cookies Vault Trash

## Pruebas de TipoPaqueteController ---- Eliminar TipoPaquete

The screenshot shows a REST client interface with a DELETE request to `http://127.0.0.1:8080/api/tipoPaquete/8`. The response is a 200 OK status with a response time of 17 ms and a body size of 123 B. The response body is empty.

Params Authorization Headers (7) **Body** Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results 200 OK • 17 ms • 123 B

Pretty Raw Preview Visualize Text

```
1
```

4. Consume la API REST principal. El método o tecnología de consumo es libre, pueden usar Spring boot (WebClient)

```
EquipoMedicoWebClientService.java
11 @Service 2 usages
12 public class EquipoMedicoWebClientService {
13
14     @Autowired 5 usages
15     private WebClient webClient;
16
17     public List<EquipoMedicoDTO> getAll(){
18         Mono<List<EquipoMedicoDTO>> equiposMono = webClient.get() RequestHeadersUriSpec<capture of ?>
19             .uri(uri: "/") capture of ?
20             .retrieve() ResponseSpec
21             .bodyToFlux(EquipoMedicoDTO.class) Flux<EquipoMedicoDTO>
22             .collectList();
23         List<EquipoMedicoDTO> equipos = equiposMono.block();
24         return equipos;
25     }
26
27     public EquipoMedicoDTO getEquipoById(Integer id){ 2 usages
28         Mono<EquipoMedicoDTO> equipoMedicoDTOMono = webClient.get() RequestHeadersUriSpec<capture of ?>
29             .uri(uri:("/{id}", id) capture of ?
30             .retrieve() ResponseSpec
31             .bodyToMono(EquipoMedicoDTO.class);
32         EquipoMedicoDTO equipoMedicoDTO = equipoMedicoDTOMono.block();
33         return equipoMedicoDTO;
34     }
35 }
Run SpringProyectoFinalApplication
M10AvProyectoFinal > src > main > java > dgtic > core > clientweb > services > EquipoMedicoWebClientService > crearEquipo 54:6 CRLF UTF-8 4 spaces
```

```
EquipoMedicoWebClientService.java
12 public class EquipoMedicoWebClientService {
13     public EquipoMedicoDTO getEquipoById(Integer id){ 2 usages
14     }
15
16     public EquipoMedicoDTO actualizarEquipo(EquipoMedicoDTO equipoMedicoDTO){ 2 usages
17         return webClient.put().uri(uri:("/{id}", equipoMedicoDTO.getId()) RequestBodySpec
18             .bodyValue(equipoMedicoDTO) RequestHeadersSpec<capture of ?>
19             .retrieve() ResponseSpec
20             .bodyToMono(EquipoMedicoDTO.class) Mono<EquipoMedicoDTO>
21             .block();
22     }
23
24     public EquipoMedicoDTO crearEquipo(EquipoMedicoDTO equipoMedicoDTO) { 1 usage
25         // Realizar una solicitud POST al endpoint externo para crear un equipo
26         Mono<EquipoMedicoDTO> equipoMedicoDTOMono = webClient.post() RequestBodyUriSpec
27             .uri(uri: "/") // Ruta del endpoint del servicio externo
28             .body(Mono.just(equipoMedicoDTO), EquipoMedicoDTO.class) // Enviar el DTO como cuerpo de la solicitud
29             .retrieve() ResponseSpec
30             .bodyToMono(EquipoMedicoDTO.class);
31         // Mapear la respuesta al DTO
32         EquipoMedicoDTO equipoMedicoDTO1 = equipoMedicoDTOMono.block();
33         return equipoMedicoDTO1;
34     }
35 }
Run SpringProyectoFinalApplication
M10AvProyectoFinal > src > main > java > dgtic > core > clientweb > services > EquipoMedicoWebClientService > crearEquipo 54:6 CRLF UTF-8 4 spaces
```

```
EquipoMedicoWebClientService.java
12 public class EquipoMedicoWebClientService {
13
14     public void deleteEquipo(Integer id) { 1 usage
15         webClient.delete() RequestHeadersUriSpec<capture of ?>
16             .uri(uri:("/{id}", id) capture of ?
17             .retrieve() ResponseSpec
18             .toBodilessEntity() Mono<ResponseEntity<...>>
19             .block();
20     }
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
Run SpringProyectoFinalApplication
M10AvProyectoFinal > src > main > java > dgtic > core > clientweb > services > EquipoMedicoWebClientService > crearEquipo 54:6 CRLF UTF-8 4 spaces
```

5. Se implementaron las vistas para la API principal

Vista de todos los equipos Medicos que se tienen

Nombre	Existencias	Categoria			
Shaver	8	Equipos Artroscopicos	Ver	Editar	Eliminar
Bomba de Irrigacion	6	Equipos Insuflacion y Lavado	Ver	Editar	Eliminar
Maquina De Anestecia	6	Equipos de Anestecia	Ver	Editar	Eliminar
Bisturi	10	Equipos Artroscopicos	Ver	Editar	Eliminar
Esponja de Limpieza	6	Equipos Insuflacion y Lavado	Ver	Editar	Eliminar
EquipoTest	15	category	Ver	Editar	Eliminar
Equipo Medico Test 22	9	category	Ver	Editar	Eliminar
Maquina De Anestecia TEST	7	Equipos de Anestecia	Ver	Editar	Eliminar
Shaver 2 Test 2	6	Equipos de Anestecia	Ver	Editar	Eliminar
Shaver 2 hola	9	Categoria Test 3	Ver	Editar	Eliminar

Vista de todos los detalles del Equipo Medico

Navbar

Home

Equipo

Dropdown ▾

Disabled

Shaver

Detalle:

Nombre:

Shaver

Existencias:

8

Categoria:

Equipos Artroscopicos

Paquetes:

Paquete Artroscopia

Rodilla, Paquete Artroscopia

Hombro, Paquete Artroscopia

Tobillo

Ir a todos

Derechos reservasos 2024.

Vista de actualización de un Equipo Médico existente

Navbar

Home

Equipo

Dropdown ▾

Disabled

Search

Search

nombre

Shaver

Existencias

8

Categorías Disponibles

Nombre: Equipos Artroscopicos

Paquetes Disponibles

Nombre: Paquete Artroscopia Rodilla  
Nombre: Paquete Artroscopia Hombro  
Nombre: Paquete Artroscopia Tobillo  
Nombre: Paquete Test2

Guardar

Derechos reservados 2024.

Navbar

Home

Equipo

Dropdown ▾

Disabled

Search

Search

Actualización exitosa :)

nombre

Shaver electrico

Existencias

9

Categorías Disponibles

Nombre: Equipos Artroscopicos

Paquetes Disponibles

Nombre: Paquete Artroscopia Rodilla  
Nombre: Paquete Artroscopia Hombro  
Nombre: Paquete Artroscopia Tobillo  
Nombre: Paquete Test2

Guardar

Derechos reservados 2024.