



# Manual técnico

08/02/2022

—

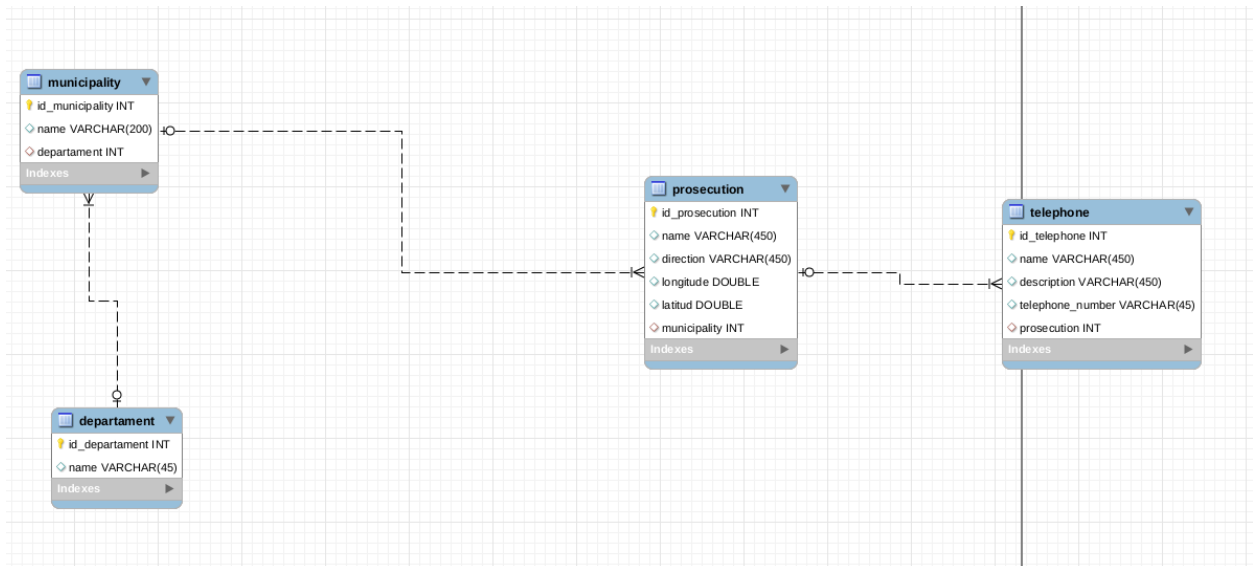
Daniel Alberto Gonzalez Gonzale  
Ministerio público

## Herramientas:

- Java 11
- Spring boot
- React
- Mysql

## Base de datos

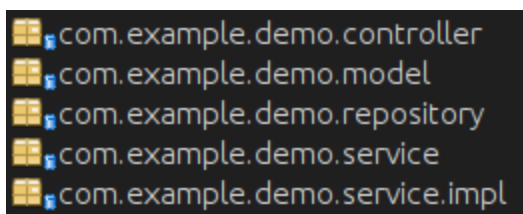
Para la base de datos se trabajo con mysql, la herramienta de workbeanch y phpmyadmin para ayudarnos en la creación, edición e implementación de la base de datos siendo este el diagrama que representa el trabajo:



## Backend

Para el backend se utilizó spring boot como framework ya existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE.

Dentro del proyecto se encuentran 5 paquetes importantes los cuales son:



Los cuales son utilizados para la arquitectura de desarrollo para servicios REST donde se exponen los siguientes:

```
@GetMapping("/{id}")
public ResponseEntity<Prosecution> read(@PathVariable Integer id) {
    Optional<Prosecution> oDepartament = prosecutionService.findById(id);
    if (!oDepartament.isPresent()) {
        return ResponseEntity.notFound().build();
    }

    return ResponseEntity.ok(oDepartament.get());
}

@GetMapping
public ResponseEntity<List<Prosecution>> readAll() {
    List<Prosecution> departaments = StreamSupport.stream(prosecutionService.findAll().spliterator(), false)
        .collect(Collectors.toList());
    return ResponseEntity.ok(departaments);
}

@PostMapping
public ResponseEntity<Prosecution> create(@RequestBody Prosecution bridge) {
    System.out.println("Creando.....");
    return ResponseEntity.status(HttpStatus.CREATED).body(prosecutionService.save(bridge));
}

@PutMapping("/{id}")
public ResponseEntity<Prosecution> update(@RequestBody Prosecution bridge, @PathVariable Integer id) {
    Optional<Prosecution> bridgeOptional = prosecutionService.findById(id);
    if (!bridgeOptional.isPresent()) {
        return ResponseEntity.notFound().build();
    }
    BeanUtils.copyProperties(bridge, bridgeOptional.get());

    return ResponseEntity.status(HttpStatus.CREATED).body(prosecutionService.save(bridgeOptional.get()));
}

@DeleteMapping("/{id}")
public ResponseEntity<?> delete(@PathVariable Integer id) {
    if (!prosecutionService.findById(id).isPresent()) {
        return ResponseEntity.notFound().build();
    }
    prosecutionService.deleteById(id);
    return ResponseEntity.noContent().build();
}
```

Siendo dos de lectura, uno de escritura, uno de modificación y uno de eliminación por parte del objeto prosecution.

## Frontend

Este fue creado con la librería React y consta con los siguientes métodos:

```
peticionGet = () => { ...
}

getAllDepartaments = () => { ...
}

peticionPost = async () => { ...
}

peticionPut = () => { ...
}

peticionDelete = () => { ...
}

modalInsertar = () => { ...
}

seleccionarEmpresa = (prosecution) => { ...
}

handleChange = async e => { ...
}

//change
handleChangeSelect = (newValue, actionMeta) => { ...
};
```

Métodos para la petición post, put, get, delete para poder crear, editar, obtener y eliminar datos a través de los servicios antes mencionados.

Método handle Change que sirven para estar escuchando constantemente los cambios en los inputs de nuestra app web.

Método de seleccionar Empresa que sirve para cambiar los datos de nuestro formulario al momento de actualizar.