

INFORME

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

2 usages
@Column(nullable = false)
@NotEmpty
private String nombre;

2 usages
@Column(unique = true)
@NotEmpty
>Email
private String email;

2 usages
@NotEmpty
private String password;
```

```
2 usages  Adrisher *
@Repository
public interface UsuarioRepository extends CrudRepository<Usuario, Long> {
    1 usage  Adrisher
    Optional<Usuario> findByEmail(String email);
}
```

```
import ...

3 usages 1 implementation 🧑Adrisher *
public interface UuarioService {
    1 usage 1 implementation 🧑Adrisher
    List<Uuario> listar();
    3 usages 1 implementation 🧑Adrisher
    Optional<Uuario> porId(Long id);
    2 usages 1 implementation 🧑Adrisher
    Uuario guardar(Uuario usuario);
    1 usage 1 implementation 🧑Adrisher
    void eliminar(Long id);
    2 usages 1 implementation 🧑Adrisher
    Optional<Uuario> findByEmail (String email);
}
```

```
1 usage 🧑Adrisher
6 @Override
7 public List<Uuario> listar() { return (List<Uuario>) repository.findAll() ; }
0

3 usages 🧑Adrisher
1 @Override
2 public Optional<Uuario> porId(Long id) { return repository.findById(id); }
5


2 usages 🧑Adrisher
6 @Override
7 public Uuario guardar(Uuario usuario) { return repository.save(usuario); }
0


1 usage 🧑Adrisher
1 @Override
2 public void eliminar(Long id) { repository.deleteById(id); }
5

2 usages 🧑Adrisher
6 @Override
7 public Optional<Uuario> findByEmail(String email) {
8     return repository.findByEmail(email);
9 }
0
```

2 usages  Adrishar

```
private static ResponseEntity<Map<String, String>> validar(BindingResult bindingResult) {  
    Map<String, String> errores = new HashMap<>();  
    bindingResult.getFieldErrors().forEach(err ->{  
        errores.put(err.getField(), "El campo" + err.getField() + " " + err.getDefaultMessage());  
    });  
    return ResponseEntity.badRequest().body(errores);  
}
```

 Adrishar

@PostMapping 

```
public ResponseEntity<?> crear(@Valid @RequestBody Usuario usuario, BindingResult bindingResult){  
    if(bindingResult.hasErrors()){  
        return validar(bindingResult);  
    }  
    if(service.findByEmail(usuario.getEmail()).isPresent()){  
        return ResponseEntity.badRequest().body(Collections.  
            singletonMap("Message", "Ya existe el correo"));  
    }  
    return ResponseEntity.status(HttpStatus.CREATED).body(service.guardar(usuario));  
}
```

 Adrishar

@PutMapping("/{id}")

```
public ResponseEntity<?> editar(@Valid @RequestBody Usuario usuario, BindingResult bindingResult, @PathVariable Long id){  
    if(bindingResult.hasErrors()){  
        return validar(bindingResult);  
    }  
    Optional<Usuario> usuarioOptional=service.findById(id);  
    if (usuarioOptional.isPresent()){  
        Usuario usuarioDb=usuarioOptional.get();  
  
        if(service.findByEmail(usuario.getEmail()).isPresent()  
            && !usuario.getEmail().equalsIgnoreCase(usuarioDb.getEmail())){  
            return ResponseEntity.badRequest().body(Collections.  
                singletonMap("Message", "Ya existe el correo"));  
        }  
        usuarioDb.setNombre(usuario.getNombre());  
        usuarioDb.setEmail(usuario.getEmail());  
        usuarioDb.setPassword(usuario.getPassword());  
        return ResponseEntity.status(HttpStatus.CREATED).body(service.guardar(usuarioDb));  
    }  
    return ResponseEntity.notFound().build();  
}
```

POST localhost:8001

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   "nombre": "Adrian Guerrero",
3   "email": "adrianguerrero862@gmail.com",
4   "password": "1234"
5 }
```

Body Cookies Headers (5) Test Results 🌐 Status: 201 Created

Pretty Raw Preview Visualize **JSON** ▾ ↻

```
1 {
2   "id": 2,
3   "nombre": "Adrian Guerrero",
4   "email": "adrianguerrero862@gmail.com",
5   "password": "1234"
6 }
```

```
1 {
2   "nombre": "",
3   "email": "",
4   "password": ""
5 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize **JSON** ▾ ↻

```
1 {
2   "password": "El campopassword no debe estar vacío",
3   "nombre": "El camponombre no debe estar vacío",
4   "email": "El campoemail no debe estar vacío"
5 }
```

```
1 {
2   "nombre": "Juan",
3   "email": "juan.perez@example.com",
4   "password": "123445555"
5 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize **JSON** ▾ ↻

```
1 {
2   "Message": "Ya existe el correo"
3 }
```