



Buenas prácticas servicios REST

Validaciones

Manejo de
Excepciones

Documentación

Pruebas de
Stress

Modelo Richardson

GET/POST/
PUT/DELETE/

Response Status



Validaciones

```
import javax.validation.constraints
```

```
@Size(min = 3, message = "Nombres debe tener minimo 3 caracteres")
```

```
@Min(value = 18, message = "Age should not be less than 18")
```

```
@Max(value = 150, message = "Age should not be greater than 150")
```

```
@Email(message = "Email should be valid")
```



Manejo de errores

```
@ControllerAdvice
@RestController
public class ResponseExceptionHandler extends ResponseEntityExceptionHandler{

    @ExceptionHandler(Exception.class)
    public final ResponseEntity<Object> manejarTodasExcepciones(Exception ex, WebRequest request){
        ExceptionResponse exceptionResponse = new ExceptionResponse(new Date(), ex.getMessage(), request.getDescription(false));
        return new ResponseEntity(exceptionResponse, HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @ExceptionHandler(ModeloNotFoundException.class)
    public final ResponseEntity<Object> manejarModeloExcepciones(ModeloNotFoundException ex, WebRequest request){
        ExceptionResponse exceptionResponse = new ExceptionResponse(new Date(), ex.getMessage(), request.getDescription(false));
        return new ResponseEntity(exceptionResponse, HttpStatus.NOT_FOUND);
    }

    @Override
    protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex,
        HttpHeaders headers, HttpStatus status, WebRequest request) {
        ExceptionResponse exceptionResponse = new ExceptionResponse(new Date(), "Validación fallida", ex.getBindingResult().toString());
        return new ResponseEntity(exceptionResponse, HttpStatus.BAD_REQUEST);
    }
}
```



Documentación (Swagger)

```
@Configuration
@EnableSwagger2
public class SwaggerConfig {

    public static final Contact DEFAULT_CONTACT = new Contact("MitoCode Network", "https://www.mitocode.com",
        "cursos@mitocodenetwork.com");
    public static final ApiInfo DEFAULT_API_INFO = new ApiInfo("Mediapp Api Documentation", "Mediapp Api Documentation", "1.0",
        "PREMIUM", DEFAULT_CONTACT, "Apache 2.0", "http://www.apache.org/licenses/LICENSE-2.0",
        new ArrayList<VendorExtension>());

    @Bean
    public Docket api() {
        return new Docket(DocumentationType.SWAGGER_2).apiInfo(DEFAULT_API_INFO);
    }
}
```

<http://localhost:8080/v2/api-docs>

<http://localhost:8080/swagger-ui.html>

Prueba de stress



RESTful Stress 1.6.0 Update to v1.6.4 Settings History Chart Performance Storage @

options Atom Scenario Massive

iterations delay (ms) timeout (ms) warmup

target

headers ☐ with credentials

```
1 {  
2   "Accept": "application/json",  
3   "Content-Type": "application/json; charset=utf-8"  
4 }
```

request body json form-data x-www-form-urlencoded

```
1 |
```

NoName STORE SETTINGS DOWNLOAD UPLOAD



Response Status

200 - SUCCESS

404 - RESOURCE NOT FOUND

400 - BAD REQUEST

201 - CREATED

401 - UNAUTHORIZED

500 - SERVER ERROR

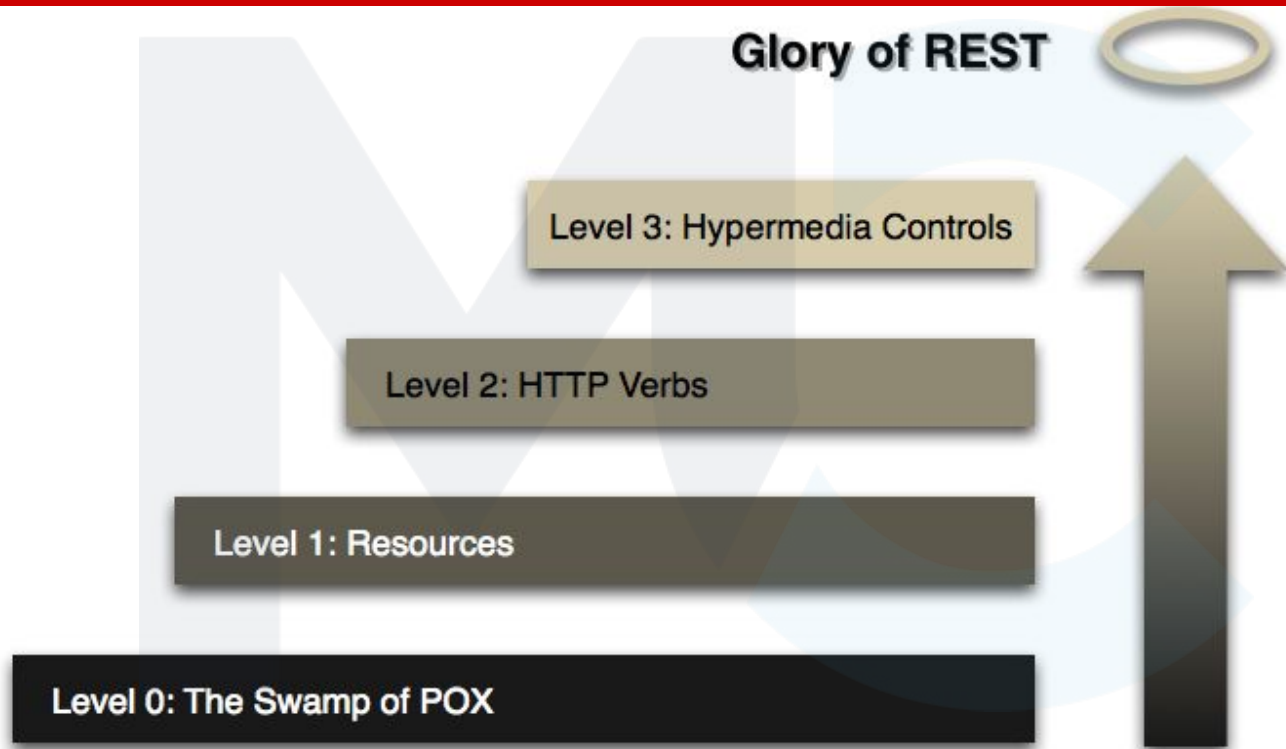


Unos tips más

- No exponer información sensible en las URL
- Usar plurales /pacientes /medicos
/pacientes/1 /medicos/1
- Usar sustantivos para los recursos
- Sustantivo luego verbo (excepciones)
/buscar
/pacientes/1/calificar
/pacientes/calificar



Modelo de Madurez Richardson



Fuente: <https://martinfowler.com/articles/richardsonMaturityModel.html>