

REBRICA DEL PROYECTO FINAL DE CURSO

Sección	:	Única
Curso	:	Reactive Developer: Spring WebFlux & MongoDB
Instructor	:	Aristedes Novoa Arbildo
Horario	:	Martes y Jueves de 9:00pm a 11:00pm
Alumno	:	

I. CONSIDERACIONES GENERALES

- El desarrollo del proyecto es personal, sin embargo, está permitido colaborar con sus compañeros de clase o la asesoría de cualquier otro profesional con experiencia en desarrollo con Spring WebFlux, RxJava, Vert.x u otro framework reactivo.
- Está permitido usar libros, tutoriales, scripts y presentaciones para revisar y repasar conceptos y casos a reutilizar.
- Reutilizar proyectos previos desarrollados en clase o descargados de internet, incluyendo librerías y rutinas de código; siempre que no violen la propiedad de derecho del autor.

II. ACTIVIDADES A REALIZAR

- Seleccionar un caso de gestión de un proceso de negocio que incluya documentos relacionados y una jerarquía de 3 a más niveles de profundidad con elementos individuales o colecciones. Diseñar un diagrama de contexto u otro para esquematizar la organización de los documentos y colecciones.
- Crear un repositorio en github, gitlab o cualquier otra plataforma de gestión de código fuente para versionar el código fuente de sus proyectos.
- El caso debe incluir las principales funcionalidades una aplicación reactiva utilizando Spring WebFlux, considerando entre los más importantes los siguientes (cada componente vale 2 puntos):

#	Componente	Consideraciones	Cumple ¹		Comentarios de revisión
			Si	No	
1	Base de datos NoSQL	Crear la base de datos en MongoDB, con sus respectivos documentos y colecciones, incluyendo la carga de datos manual a través de archivos .json personalizados y documentados.			
2	API REST v1 – Controller	Crear una Api REST reactiva basada en Controller que implemente los métodos GET (URI amigable-path y query params), POST, PUT, PATH y DELETE. La API debe ser idempotente.			
3	API REST v2 – Functional Endpoint	Adaptar la API REST v1 utilizando Functional Endpoint (handlers & routers), incluyendo el uso de lambdas. Además de incluir un documento como atributo interno.			
4	API REST v3 –Functional Endpoint	Implementar mejoras a la API REST v2 - aplicando DTOs; utilizar por lo menos 1(una) librerías de mapeo (MapStuct, ModelMapper u otros) y crear un mapper personalizado.			
5	Procesos de Negocio	Crear las API REST reactivas que soporten un proceso de negocio (maestro detalle), el proceso debe implementar cálculos, validaciones, transformaciones, etc.			
6	Gestión de Validaciones y Excepciones	Implementar validaciones estándares y personalizadas (usar Jackarta o Spring Validation) que permitan garantizar la consistencia del las API y generar mensajes personalizadas al cliente. Centralizar la gestión de las excepciones			
7	Documentación	Implementar la documentación de las APIs utilizando Swagger, crear constantes para los valores comunes y utilizar archivos properties para facilitar su modificación.			

8	Consumo	Realizar el consumo de una API REST reactiva utilizando WebClient con sus respectivas configuraciones. EL consumo puede realizarse en una aplicación de consola, otra API o a través de pruebas unitarias(JUnit)			
9	Seguridad	Reutilizar el paquete de seguridad con Spring Security (proporcionado en clase) en su proyecto y de acuerdo a su caso práctico, personalizar los parámetros generales y otros componentes de seguridad.			
10	Otros	Otras funcionalidades o uso de componentes de Spring WebFlux como valor agregado que el alumno aporte en su entrenamiento			

III. PRESENTACION

El alumno deberá realizar la presentación de su proyecto con la implementación de las funcionalidades solicitadas en la fecha establecida por la Coordinación Académica de Galaxy Training.