

DOJO PROGRAMACIÓN REACTIVA

DESCRIPCIÓN

Como programa académico se quiere tener un Servicio REST que permita crear el calendario de un **training activo** para los programas académicos de Sofka U.

DEPENDENCIA

- listar los cursos dentro de un programa académico

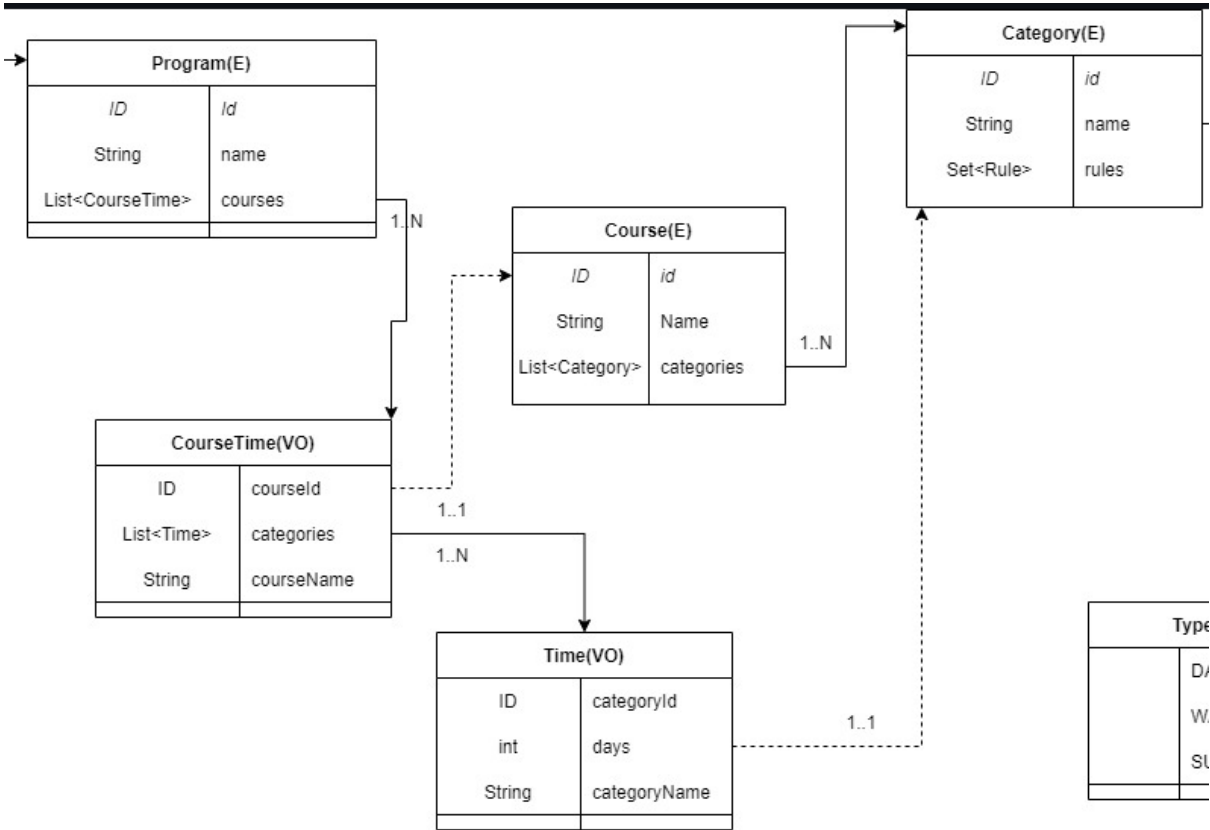
CRITERIOS DE ACEPTACIÓN

- Dado un programa asociado a un training cuando se actualiza o se activa el mismo se debe generar un calendario donde se distribuyen los días de la categoría, excluyendo los fines de semana (sábado y domingo)
- Cada día del calendario debe contener la fecha y asociado todo el objeto categoría.
- Se debe tener un endpoint para poder generar el calendario según sea el training.

DoD

- Pruebas a nivel de dominio al 100% para el backend (cobertura de código).
- Debe tener operaciones reactivas y el servicio debe ser no bloqueante
- Todas las capas del servicios deben ser reactivas
- Se debe almacenar el calendario en una base de datos

MODELO DE DOMINIO



EJEMPLO

LUNES	1	Java básico
MARTES	2	Java básico
MIÉRCOLES	3	Java básico
JUEVES	4	Java avanzado
VIERNES	5	Java avanzado
LUNES	8	Java especialista
MARTES	9	Java especialista
MIÉRCOLES	10	Java especialista
JUEVES	11	Java especialista
VIERNES	12	Resumen
LUNES	15	Certificación

INDICACIONES

1. Descargar el siguiente link
<https://start.spring.io/#!type=maven-project&language=java&platformVersion=2.6.1&packaging=jar&jvmVersion=11&groupId=co.com.sofka&artifactId=calendar&name=calendar&description=Demo%20project%20for%20Spring%20Boot&packageName=co.com.sofka.calendar&dependencies=webflux,data-mongodb-reactive>
2. Crear un repositorio git
3. Refactorizar la base de pruebas unitarias para que sea reactivo
4. Crear primer solución temporal (servicio) para refactorizar y ver cobertura de código
5. Exponer el servicio en un controlador (tener un endpoint para generar el calendario)
6. Crear otro servicio que guarde los datos que fueron generados - OPCIONAL
7. Exponer un servicio que me muestre el calendario generado - OPCIONAL

Repositorio: <https://github.com/Sofka-XT/calendar-sofkau>

Resolver los siguientes TODOs:

```
co.com.sofka.calendar.services 10 items
└─ SchedulerService.java 3 items
   (26, 7) // TODO: deben retornar un flux de programDate Flux<ProgramDate>
   (32, 11) // TODO: debe pasarlo a reactivo, no puede trabaja elementos bloqueantes
   (33, 11) // TODO: trabajar el map reactivo y no deben coleccionar
└─ SchedulerServiceTest.java 7 items
   (33, 11) // TODO: modificar el test para que el act sea reactivo, usando stepverifier
   (41, 11) // TODO: hacer una subscripción de el servicio reactivo
   (44, 56) Assertions.assertEquals(13, response.size()); // TODO: hacer de otro modo
   (45, 81) Assertions.assertEquals(getSnapResult(), new Gson().toJson(response)); // TODO: hacer de otro modo
   (56, 11) // TODO: hacer de otro modo
   (58, 71) schedulerService.generateCalendar(programId, startDate); // TODO: hacer una subscripción de el servicio reactivo
   (61, 95) Assertions.assertEquals("El programa academnico no existe", exception.getMessage()); // TODO: hacer de otro modo
```