

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática


## **DP2-Reporte de Test**



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

<b><u>Group:</u></b>	C1.016
<b><u>Repository:</u></b>	<a href="https://github.com/luchersol/Acme-SF-D02-24.1">https://github.com/luchersol/Acme-SF-D02-24.1</a>
<b><u>Student #3</u></b>	
	
<b>UVUS:</b> edurobrus	
<b>Name:</b> Robles Russo, Eduardo	
<b>Email:</b> edurobrus@alum.us.es	
<b><u>Date:</u></b>	Sevilla Mayo 26, 2024

1. Pruebas funcionales	2
2. Pruebas de rendimiento	11

## 1. Pruebas funcionales

### 1.1 training-module

#### Safe:

Se crea un nuevo módulo de entrenamiento desde developer1. Cosas que se tuvieron en cuenta para realizar los test:

Momentos:

- Momento Mínimo: 2000/01/01 00:00
- Momento Actual: 2022/07/30 00:00
- Momento Base: 2022/07/29 23:00
- Se emiten advertencias para momentos que están por debajo del momento mínimo o actual.

URLs:

- URL Base: <http://www.lorem-ipsum.org>
- Se emiten advertencias para URLs más cortas de 7 caracteres o más largas de 255 caracteres.

Direcciones de correo electrónico:

- Correo electrónico base: [acme@lorem-ipsum.org](mailto:acme@lorem-ipsum.org)
- Se emiten advertencias para direcciones de correo electrónico más cortas de 6 caracteres o más largas de 254 caracteres.

Para la entidad TrainingModule, parece tener los siguientes atributos:

- código: Una cadena única que coincide con el patrón `[A-Z]{1,3}-[0-9]{3}`
- creationMoment: Una marca de tiempo que debe estar en el pasado o en el presente
- detalles: Una cadena no vacía con una longitud máxima de 100
- difficultyLevel: Un nivel de dificultad no nulo
- updateMoment: Una marca de tiempo que debe estar en el pasado o en el presente, debe de ser posterior a la creación del módulo pero como lo pone el sistema no hay ninguna validación.

- link: Una URL válida
- estimatedTotalTime: Un doble no nulo entre 0.0 (inclusive) y 100.0 (inclusive)
- draftMode: Un booleano no nulo

## Hacking:

### Diferente Rol:

Eliminar Módulo de Entrenamiento:

1. Borrar un módulo existente publicado de otro usuario/rol.
2. Borrar un módulo existente no publicado de otro usuario/rol con hijos publicados.
3. Borrar un módulo existente no publicado de otro usuario/rol con hijos no publicados.
4. Borrar un módulo que no existe de otro usuario/rol.

Listar Módulo de Entrenamiento:

5. Listar módulos no publicados de otro usuario/rol.
6. Listar módulos que no existen de otro usuario/rol.

Mostrar Módulo de Entrenamiento:

7. Mostrar un módulo existente publicado de otro usuario/rol.
8. Mostrar un módulo existente no publicado de otro usuario/rol.
9. Mostrar un módulo que no existe de otro usuario/rol.

Crear Módulo de Entrenamiento:

10. Crear un módulo que existe para otro usuario/rol.
11. Crear un módulo que no existe para otro usuario/rol.

Actualizar Módulo de Entrenamiento:

12. Actualizar un módulo existente publicado de otro usuario/rol.
13. Actualizar un módulo existente no publicado de otro usuario/rol.
14. Actualizar un módulo que no existe de otro usuario/rol.

Publicar Módulo de Entrenamiento:

15. Publicar un módulo existente ya publicado de otro usuario/rol.
16. Publicar un módulo existente no publicado con hijos publicados de otro usuario/rol.
17. Publicar un módulo existente no publicado con hijos no publicados de otro usuario/rol.
18. Publicar un módulo que no existe de otro usuario/rol.

## **Mismo Rol, Diferente Usuario:**

Eliminar Módulo de Entrenamiento:

19. Borrar un módulo existente publicado de otro usuario.
20. Borrar un módulo existente no publicado de otro usuario con hijos publicados.
21. Borrar un módulo existente no publicado de otro usuario con hijos no publicados.
22. Borrar un módulo que no existe de otro usuario.

Listar Módulo de Entrenamiento:

23. Listar módulos no publicados de otro usuario.
24. Listar módulos que no existen de otro usuario.

Mostrar Módulo de Entrenamiento:

25. Mostrar un módulo existente publicado de otro usuario.
26. Mostrar un módulo existente no publicado de otro usuario.
27. Mostrar un módulo que no existe de otro usuario.

Crear Módulo de Entrenamiento:

28. Crear un módulo que existe para otro usuario.
29. Crear un módulo que no existe para otro usuario.

Actualizar Módulo de Entrenamiento:

30. Actualizar un módulo existente publicado de otro usuario.

- 31. Actualizar un módulo existente no publicado de otro usuario.
- 32. Actualizar un módulo que no existe de otro usuario.

Publicar Módulo de Entrenamiento:

- 33. Publicar un módulo existente ya publicado de otro usuario.
- 34. Publicar un módulo existente no publicado con hijos publicados de otro usuario.
- 35. Publicar un módulo existente no publicado con hijos no publicados de otro usuario.
- 36. Publicar un módulo que no existe de otro usuario.

## **Mismo Rol, Mismo Usuario, Cosas No Permitidas:**

Eliminar Módulo de Entrenamiento:

- 37. Borrar un módulo existente publicado.
- 38. Borrar un módulo existente no publicado con hijos no publicados.
- 39. Borrar un módulo que no existe.

Listar Módulo de Entrenamiento:

- 40. Listar módulos que no existen.

Mostrar Módulo de Entrenamiento:

- 41. Mostrar un módulo que no existe.

Crear Módulo de Entrenamiento:

- 42. Crear un módulo que ya existe.

Actualizar Módulo de Entrenamiento:

- 43. Actualizar un módulo existente publicado.
- 44. Actualizar un módulo que no existe.

Publicar Módulo de Entrenamiento:

45. Publicar un módulo existente ya publicado.
46. Publicar un módulo existente no publicado con hijos no publicados.
47. Publicar un módulo que no existe.

## Coverage:

En el paquete `acme.features.developer.trainingModule` solo salen en amarillo los assert `object != null`

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
acme.features.developer.trainingModule	94,5 %	1.244	72	1.316
> DeveloperTrainingModuleCreateService.java	94,5 %	274	16	290
> DeveloperTrainingModuleDeleteService.java	93,2 %	221	16	237
> DeveloperTrainingModulePublishService.java	92,9 %	210	16	226
> DeveloperTrainingModuleUpdateService.java	94,9 %	297	16	313
> DeveloperTrainingModuleListMineService.java	93,9 %	62	4	66
> DeveloperTrainingModuleShowService.java	97,3 %	144	4	148
> DeveloperTrainingModuleController.java	100,0 %	36	0	36

## Capturas validate:

```
@Override
public void validate(final TrainingModule object) {
    assert object != null;
    Project project;
    // Validate code
    if (!super.getBuffer().getErrors().hasErrors("code")) {
        TrainingModule existing;
        existing = this.repository.findOneTrainingByCode(object.getCode());
        super.state(existing == null, "code", "developer.training-module.form.error.duplicated");
    }
    if (!super.getBuffer().getErrors().hasErrors("project")) {
        Boolean isDraftMode = this.repository.ProjectIsDraftMode(object.getProject().getId());
        Assert.state(!isDraftMode, "Access is not authorised");
    }
    if (!super.getBuffer().getErrors().hasErrors("link") && !object.getLink().isEmpty()) {
        // Validate link length
        int linkLength = object.getLink().length();
        super.state(linkLength >= 7 && linkLength <= 255, "link", "developer.training-module.form.error.link.size");
    }
    super.validateSpam(object);
}
```

## Capturas de los authorise:



```

27
28 @Service
29 public class DeveloperTrainingModuleDeleteService extends AbstractService<Developer, TrainingModule> {
30
31     @Autowired
32     private DeveloperTrainingModuleRepository repository;
33
34
35     @Override
36     public void authorise() {
37         boolean status;
38         int moduleId;
39         TrainingModule module;
40         Developer developer;
41
42         moduleId = super.getRequest().getData("id", int.class);
43         module = this.repository.findOneTrainingById(moduleId);
44         developer = module == null ? null : module.getDeveloper();
45         status = module != null && module.getDraftMode() && super.getRequest().getPrincipal().hasRole(developer);
46
47         super.getResponse().setAuthorised(status);
48     }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130 @Service
131 public class DeveloperTrainingModulePublishService extends AbstractAntiSpamService<Developer, TrainingModule> {
132
133     @Autowired
134     private DeveloperTrainingModuleRepository repository;
135
136
137     @Override
138     public void authorise() {
139         boolean status;
140         int moduleId;
141         TrainingModule module;
142         Developer developer;
143
144         moduleId = super.getRequest().getData("id", int.class);
145         module = this.repository.findOneTrainingById(moduleId);
146         developer = module == null ? null : module.getDeveloper();
147         status = module != null && module.getDraftMode() && super.getRequest().getPrincipal().hasRole(developer);
148
149         super.getResponse().setAuthorised(status);
150     }
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231 @Service
232 public class DeveloperTrainingModuleUpdateService extends AbstractAntiSpamService<Developer, TrainingModule> {
233
234     @Autowired
235     private DeveloperTrainingModuleRepository repository;
236
237
238     @Override
239     public void authorise() {
240         boolean status;
241         int moduleId;
242         TrainingModule module;
243         Developer developer;
244
245         moduleId = super.getRequest().getData("id", int.class);
246         module = this.repository.findOneTrainingById(moduleId);
247         developer = module == null ? null : module.getDeveloper();
248         status = module != null && module.getDraftMode() && super.getRequest().getPrincipal().hasRole(developer);
249
250         super.getResponse().setAuthorised(status);
251     }
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

28 @Service
29 public class DeveloperTrainingModuleShowService extends AbstractService<Developer, TrainingModule> {
30
31     @Autowired
32     private DeveloperTrainingModuleRepository repository;
33
34
35     @Override
36     public void authorise() {
37         boolean status;
38         int moduleId;
39         TrainingModule module;
40         Developer developer;
41
42         moduleId = super.getRequest().getData("id", int.class);
43         module = this.repository.findOneTrainingById(moduleId);
44         developer = module == null ? null : module.getDeveloper();
45         status = module != null && super.getRequest().getPrincipal().hasRole(developer);
46
47         super.getResponse().setAuthorised(status);
48     }
49

```

## 1.2 Training Session

### Safe:

Se crea un nueva sesión de entrenamiento desde developer1. Cosas que se tuvieron en cuenta para realizar los test:

Momentos:

- Momento Mínimo: 2000/01/01 00:00
- Momento Actual: 2022/07/30 00:00
- Momento Máximo: 2200/12/31 23:59
- Momento Base: 2022/07/29 23:00
- Se emiten advertencias para momentos que están por debajo del momento mínimo o actual.

URLs:

- URL Base: <http://www.lorem-ipsum.org>
- Se emiten advertencias para URLs más cortas de 7 caracteres o más largas de 255 caracteres.

Para la entidad TrainingSession, tenemos los siguientes atributos:

- Momento Máximo: 2200/12/31 23:59

- código: Una cadena única que coincide con el patrón TS-[A-Z]{1,3}-[0-9]{3}
- timeStart: Una marca de tiempo que debe ser no nula, este debe de ser al menos de 7 días más que el momento de creación de un módulo de entrenamiento y posterior o igual a la fecha mínima y máxima.
- timeEnd: Una marca de tiempo que debe ser no nula, esta debe de ser al menos de 7 días más que el timeStart y posterior o igual a la fecha mínima y máxima.
- location: Una cadena no vacía con una longitud máxima de 75
- instructor: Una cadena no vacía con una longitud máxima de 75
- contactEmail: Una dirección de correo electrónico válida y no vacía con una longitud mínima de 6 y máxima de 254
- link: Una URL válida, no más cortas de 7 caracteres o no más largas de 255 caracteres.
- draftMode: Un booleano no nulo

## Hacking:

### Diferente Rol:

Eliminar Sesión de Entrenamiento:

1. Borrar una sesión existente publicada de otro usuario/rol.
2. Borrar una sesión existente no publicada de otro usuario/rol.
3. Borrar una sesión que no existe de otro usuario/rol.
4. Borrar una sesión existente con padre publicado de otro usuario/rol.

Listar Sesión de Entrenamiento:

5. Listar sesiones no publicadas de otro usuario/rol.
6. Listar sesiones que no existen de otro usuario/rol.
7. Listar Sesión de Entrenamiento por Módulo de Entrenamiento:
8. Listar sesiones de un módulo publicado (padre) de otro usuario/rol.
9. Listar sesiones de un módulo no publicado (padre) de otro usuario/rol.
10. Listar sesiones de un módulo que no existe (padre) de otro usuario/rol.

#### Mostrar Sesión de Entrenamiento:

11. Mostrar una sesión existente publicada de otro usuario/rol.
12. Mostrar una sesión existente no publicada de otro usuario/rol.
13. Mostrar una sesión que no existe de otro usuario/rol.

#### Crear Sesión de Entrenamiento:

14. Crear una sesión que existe para otro usuario/rol (con padre publicado).
15. Crear una sesión que existe para otro usuario/rol (con padre no publicado).
16. Crear una sesión que no existe para otro usuario/rol (con padre publicado).
17. Crear una sesión que no existe para otro usuario/rol (con padre no publicado).

#### Actualizar Sesión de Entrenamiento:

18. Actualizar una sesión existente publicada de otro usuario/rol (con padre publicado).
19. Actualizar una sesión existente publicada de otro usuario/rol (con padre no publicado).
20. Actualizar una sesión existente no publicada de otro usuario/rol (con padre no publicado).
21. Actualizar una sesión que no existe de otro usuario/rol.

#### Publicar Sesión de Entrenamiento:

22. Publicar una sesión existente ya publicada de otro usuario/rol (con padre publicado).
23. Publicar una sesión existente ya publicada de otro usuario/rol (con padre no publicado).
24. Publicar una sesión existente no publicada de otro usuario/rol (con padre no publicado).
25. Publicar una sesión que no existe de otro usuario/rol.

### **Mismo Rol, Diferente Usuario:**

#### Eliminar Sesión de Entrenamiento:

- 26. Borrar una sesión existente publicada de otro usuario.
- 27. Borrar una sesión existente no publicada de otro usuario.
- 28. Borrar una sesión que no existe de otro usuario.
- 29. Borrar una sesión existente con padre publicado de otro usuario.

Listar Sesión de Entrenamiento:

- 30. Listar sesiones no publicadas de otro usuario.
- 31. Listar sesiones que no existen de otro usuario.
- 32. Listar Sesión de Entrenamiento por Módulo de Entrenamiento:
- 33. Listar sesiones de un módulo publicado (padre) de otro usuario.
- 34. Listar sesiones de un módulo no publicado (padre) de otro usuario.
- 35. Listar sesiones de un módulo que no existe (padre) de otro usuario.

Mostrar Sesión de Entrenamiento:

- 36. Mostrar una sesión existente publicada de otro usuario.
- 37. Mostrar una sesión existente no publicada de otro usuario.
- 38. Mostrar una sesión que no existe de otro usuario.

Crear Sesión de Entrenamiento:

- 39. Crear una sesión que existe para otro usuario (con padre publicado).
- 40. Crear una sesión que existe para otro usuario (con padre no publicado).
- 41. Crear una sesión que no existe para otro usuario (con padre publicado).
- 42. Crear una sesión que no existe para otro usuario (con padre no publicado).

Actualizar Sesión de Entrenamiento:

- 43. Actualizar una sesión existente publicada de otro usuario (con padre publicado).
- 44. Actualizar una sesión existente publicada de otro usuario (con padre no publicado).
- 45. Actualizar una sesión existente no publicada de otro usuario (con padre no publicado).
- 46. Actualizar una sesión que no existe de otro usuario.

Publicar Sesión de Entrenamiento:

- 47. Publicar una sesión existente ya publicada de otro usuario (con padre publicado).
- 48. Publicar una sesión existente ya publicada de otro usuario (con padre no publicado).
- 49. Publicar una sesión existente no publicada de otro usuario (con padre no publicado).
- 50. Publicar una sesión que no existe de otro usuario.

## **Mismo Rol, Mismo Usuario, Cosas No Permitidas:**

### Eliminar Sesión de Entrenamiento:

- 51. Borrar una sesión existente publicada.
- 52. Borrar una sesión que no existe.
- 53. Borrar una sesión existente con padre publicado.

### Listar Sesión de Entrenamiento:

- 54. Listar sesiones que no existen.
- 55. Listar Sesión de Entrenamiento por Módulo de Entrenamiento:
- 56. Listar sesiones de un módulo que no existe (padre).

### Mostrar Sesión de Entrenamiento:

- 57. Mostrar una sesión que no existe.

### Crear Sesión de Entrenamiento:

- 58. Crear una sesión que ya existe (con padre publicado).
- 59. Crear una sesión que ya existe (con padre no publicado).
- 60. Crear una sesión que no existe (con padre publicado).

### Actualizar Sesión de Entrenamiento:

















- 61. Actualizar una sesión existente publicada (con padre publicado).
- 62. Actualizar una sesión existente publicada (con padre no publicado).
- 63. Actualizar una sesión que no existe.

### Publicar Sesión de Entrenamiento:

64. Publicar una sesión existente ya publicada (con padre publicado).
65. Publicar una sesión existente ya publicada (con padre no publicado).
66. Publicar una sesión que no existe.

## Coverage:

En el paquete `acme.features.developer.trainingSession` solo salen en amarillo los assert object `!= null`

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼  <code>acme.features.developer.trainingSession</code>	 95,0 %	1.441	76	1.517
>  <code>DeveloperTrainingSessionCreateService.java</code>	 96,4 %	429	16	445
>  <code>DeveloperTrainingSessionDeleteService.java</code>	 91,1 %	163	16	179
>  <code>DeveloperTrainingSessionPublishService.java</code>	 91,7 %	177	16	193
>  <code>DeveloperTrainingSessionUpdateService.java</code>	 96,2 %	407	16	423
>  <code>DeveloperTrainingSessionListService.java</code>	 93,3 %	112	8	120
>  <code>DeveloperTrainingSessionShowService.java</code>	 96,7 %	118	4	122
>  <code>DeveloperTrainingSessionController.java</code>	 100,0 %	35	0	35

## Capturas validate:

```
@Override
public void validate(final TrainingSession object) {
    assert object != null;
    Long moment = object.getTrainingModule().getCreationMoment().getTime();
    // Validate code
    if (!super.getBuffer().getErrors().hasErrors("code")) {
        TrainingSession existing;
        existing = this.repository.findOneTrainingSessionByCode(object.getCode());
        super.state(existing == null, "code", "developer.training-session.form.error.duplicated");
    }

    if (!super.getBuffer().getErrors().hasErrors("timeStart")) {
        Date minDate = new GregorianCalendar(2000, Calendar.JANUARY, 1, 0, 0, 0).getTime();
        Date maxDate = new GregorianCalendar(2200, Calendar.DECEMBER, 31, 23, 59, 59).getTime();
        Date timeStart = object.getTimeStart();

        super.state(MomentHelper.isAfterOrEqual(timeStart, minDate), "timeStart", "developer.training-session.form.error.timeStart.min");
        super.state(MomentHelper.isBeforeOrEqual(timeStart, maxDate), "timeStart", "developer.training-session.form.error.timeStart.max");

        long amount = 7;
        ChronoUnit unit = ChronoUnit.DAYS;
        long offset = amount * unit.getDuration().toMillis();
        Date oneWeekAhead = new Date(moment + offset);
        boolean isOneWeekAhead = MomentHelper.isAfterOrEqual(object.getTimeStart(), oneWeekAhead);
        super.state(isOneWeekAhead, "timeStart", "developer.training-session.form.error.notOneWeekAhead");
    }

    if (!super.getBuffer().getErrors().hasErrors("timeStart") && !super.getBuffer().getErrors().hasErrors("timeEnd")) {
        long amount = 7;
        ChronoUnit unit = ChronoUnit.DAYS;
        long offset = amount * unit.getDuration().toMillis();
        Date oneWeekPeriod = new Date(object.getTimeStart().getTime() + offset);
        boolean isOneWeekLong = MomentHelper.isAfterOrEqual(object.getTimeEnd(), oneWeekPeriod);
        super.state(isOneWeekLong, "timeEnd", "developer.training-session.form.error.notOneWeekLong");
    }

    if (!super.getBuffer().getErrors().hasErrors("timeEnd")) {
        // Validate timeEnd
        Date minDate = new GregorianCalendar(2000, Calendar.JANUARY, 1, 0, 0, 0).getTime();
        Date maxDate = new GregorianCalendar(2200, Calendar.DECEMBER, 31, 23, 59, 59).getTime();
        Date timeEnd = object.getTimeEnd();
        super.state(MomentHelper.isAfterOrEqual(timeEnd, minDate), "timeEnd", "developer.training-session.form.error.timeEnd.min");
        super.state(MomentHelper.isBeforeOrEqual(timeEnd, maxDate), "timeEnd", "developer.training-session.form.error.timeEnd.max");
    }

    if (!super.getBuffer().getErrors().hasErrors("link") && !object.getLink().isEmpty()) {
        // Validate link length
        int linkLength = object.getLink().length();
        super.state(linkLength >= 7 && linkLength <= 255, "link", "developer.training-session.form.error.link.size");
    }

    super.validateSpam(object);
}
```

## Capturas de los autorise:

```
~
30 @Service
31 public class DeveloperTrainingSessionCreateService extends AbstractAntiSpamService<Developer, TrainingSession> {
32
33     // Internal state -----
34
35     @Autowired
36     private DeveloperTrainingSessionRepository repository;
37
38     // AbstractService interface -----
39
40
41     @Override
42     public void authorise() {
43         boolean status;
44         int masterId;
45         TrainingModule trainingModule;
46
47         masterId = super.getRequest().getData("masterId", int.class);
48         trainingModule = this.repository.findOneTrainingModuleById(masterId);
49         status = trainingModule != null && trainingModule.getDraftMode() && super.getRequest().getPrincipal().hasRole(trainingM
50
51         super.getResponse().setAuthorised(status);
52     }
53 }
```



```

43
24@Service
25public class DeveloperTrainingSessionDeleteService extends AbstractService<Developer, TrainingSession> {
26
27    // Internal state -----
28
29    @Autowired
30    private DeveloperTrainingSessionRepository repository;
31
32    // AbstractService interface -----
33
34
35    @Override
36    public void authorise() {
37        boolean status;
38        int trainingSessionId;
39        TrainingModule trainingModule;
40
41        trainingSessionId = super.getRequest().getData("id", int.class);
42        trainingModule = this.repository.findOneTrainingModuleByTrainingSessionId(trainingSessionId);
43        status = trainingModule != null && trainingModule.getDraftMode() && super.getRequest().getPrincipal().hasRole(tra
44
45        super.getResponse().setAuthorised(status);
46    }
}

13@Service
14public class DeveloperTrainingSessionPublishService extends AbstractAntiSpamService<Developer, TrainingSession> {
15
16    // Internal state -----
17
18    @Autowired
19    private DeveloperTrainingSessionRepository repository;
20
21    // AbstractService interface -----
22
23
24    @Override
25    public void authorise() {
26        boolean status;
27        int trainingSessionId;
28        TrainingModule trainingModule;
29        Developer developer;
30
31        trainingSessionId = super.getRequest().getData("id", int.class);
32        trainingModule = this.repository.findOneTrainingModuleByTrainingSessionId(trainingSessionId);
33        developer = trainingModule == null ? null : trainingModule.getDeveloper();
34        status = trainingModule != null && trainingModule.getDraftMode() && super.getRequest().getPrincipal().hasRole(developer);
35
36        super.getResponse().setAuthorised(status);
37    }
}

30@Service
31public class DeveloperTrainingSessionUpdateService extends AbstractAntiSpamService<Developer, TrainingSession> {
32
33    // Internal state -----
34
35    @Autowired
36    private DeveloperTrainingSessionRepository repository;
37
38    // AbstractService interface -----
39
40
41    @Override
42    public void authorise() {
43        boolean status;
44        int trainingSessionId;
45        TrainingModule trainingModule;
46        Developer developer;
47
48        trainingSessionId = super.getRequest().getData("id", int.class);
49        trainingModule = this.repository.findOneTrainingModuleByTrainingSessionId(trainingSessionId);
50        developer = trainingModule == null ? null : trainingModule.getDeveloper();
51        status = trainingModule != null && trainingModule.getDraftMode() && super.getRequest().getPrincipal().hasRole(developer)
52
53        super.getResponse().setAuthorised(status);
54    }
}
55

```

```

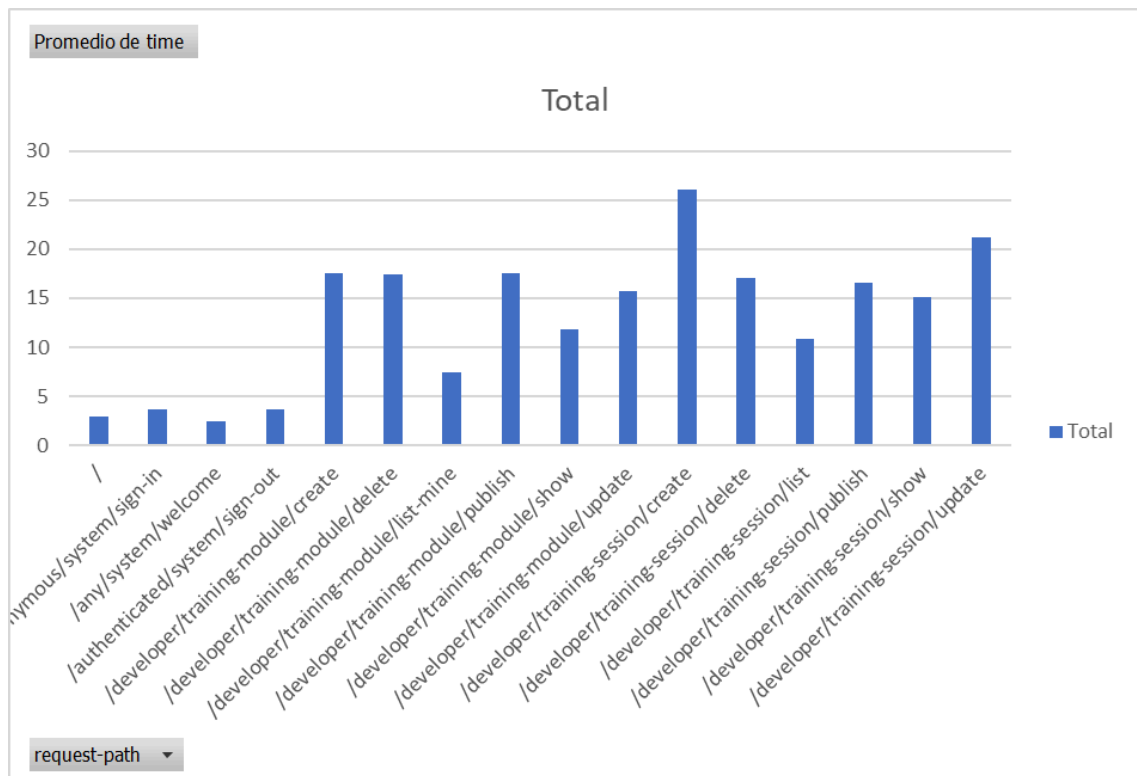
26 @Service
27 public class DeveloperTrainingSessionListService extends AbstractService<Developer, TrainingSession> {
28
29     // Internal state -----
30
31     @Autowired
32     private DeveloperTrainingSessionRepository repository;
33
34     // AbstractService interface -----
35
36     @Override
37     public void authorise() {
38         boolean status;
39         int masterId;
40         TrainingModule trainingModule;
41
42         masterId = super.getRequest().getData("masterId", int.class);
43         trainingModule = this.repository.findOneTrainingModuleById(masterId);
44         status = trainingModule != null && super.getRequest().getPrincipal().hasRole(trainingModule.getDeveloper());
45
46         super.getResponse().setAuthorised(status);
47     }
48
49
50
51
52
53
54 @Service
55 public class DeveloperTrainingSessionShowService extends AbstractService<Developer, TrainingSession> {
56
57     // Internal state -----
58
59     @Autowired
60     private DeveloperTrainingSessionRepository repository;
61
62     // AbstractService interface -----
63
64     @Override
65     public void authorise() {
66         boolean status;
67         int trainingSessionId;
68         TrainingModule trainingModule;
69
70         trainingSessionId = super.getRequest().getData("id", int.class);
71         trainingModule = this.repository.findOneTrainingModuleByTrainingSessionId(trainingSessionId);
72         status = trainingModule != null && super.getRequest().getPrincipal().hasRole(trainingModule.getDeveloper());
73
74         super.getResponse().setAuthorised(status);
75     }
76
77
78
79
80
81

```

## 2. Pruebas de rendimiento

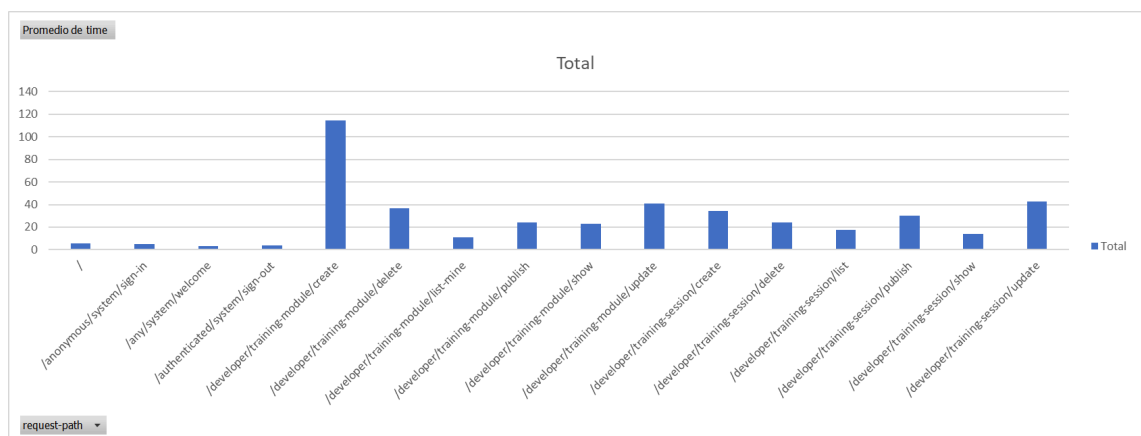
### Rendimiento Estudiante #3

Etiquetas de fila	Promedio de time
/	2,929761728
/anonymous/system/sign-in	3,617767257
/any/system/welcome	2,455397368
/authenticated/system/sign-out	3,697853125
/developer/training-module/create	17,58514545
/developer/training-module/delete	17,45760357
/developer/training-module/list-mine	7,466562903
/developer/training-module/publish	17,51477391
/developer/training-module/show	11,84338286
/developer/training-module/update	15,7039125
/developer/training-session/create	26,0973807
/developer/training-session/delete	17,07514737
/developer/training-session/list	10,88851765
/developer/training-session/publish	16,62526087
/developer/training-session/show	15,13344048
/developer/training-session/update	21,2090175
<b>Total general</b>	<b>10,92559792</b>



## Rendimiento Estudiante #1

Etiquetas de fila	Promedio de time
/	5,457660294
/anonymous/system/sign-in	5,248802247
/any/system/welcome	3,253193478
/authenticated/system/sign-out	4,017343478
/developer/training-module/create	114,574375
/developer/training-module/delete	36,73599167
/developer/training-module/list-mine	10,83494681
/developer/training-module/publish	24,11390435
/developer/training-module/show	22,91958814
/developer/training-module/update	40,71721667
/developer/training-session/create	34,26430556
/developer/training-session/delete	24,42800526
/developer/training-session/list	17,62288723
/developer/training-session/publish	29,88617391
/developer/training-session/show	13,92336957
/developer/training-session/update	42,79219375
<b>Total general</b>	<b>17,97684052</b>



## **Estadísticas descriptivas**

Estudiante 1						Estudiante 3					
Media	17,9768405					Media	10,9255979				
Error típico	0,88059549					Error típico	0,39991184				
Mediana	9,2082					Mediana	6,7546				
Moda	2,4602					Moda	6,9498				
Desviación es	22,4335914					Desviación es	11,7617599				
Varianza de la	503,266025					Varianza de la	138,338997				
Curtosis	78,3725746					Curtosis	20,7175988				
Coeficiente de	6,09458853					Coeficiente de	3,34008331				
Rango	352,0527					Rango	126,8288				
Mínimo	1,9123					Mínimo	1,0608				
Máximo	353,965					Máximo	127,8896				
Suma	11666,9695					Suma	9450,6422				
Cuenta	649					Cuenta	865				
Nivel de confi	1,72916516					Nivel de confi	0,78491234				
Interval(ms)	16,2476754	19,7060057				Interval(ms)	10,1406856	11,7105103			
Interval(s)	0,01624768	0,01970601				Interval(s)	0,01014069	0,01171051			

## **Comparación de P-Value**

Prueba z para medias de dos muestras		
	<b>61,0747</b>	<b>131,7603</b>
Media	10,867555	17,8012488
Varianza (conocida)	138,338997	503,266025
Observaciones	864	648
Diferencia hipotética de las medias	0	
z	-7,16391729	
P(Z<=z) una cola	3,9202E-13	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	7,8404E-13	
Valor crítico de z (dos colas)	1,95996398	

## **Análisis**

Es importante destacar que los resultados de los análisis no fueron favorables debido a modificaciones realizadas en las pruebas después de que mi compañero utilizara el tester#replayer. Aunque los resultados, medidos en segundos, parecen similares, se puede observar una diferencia de 7 ms entre las computadoras del estudiante #3 y #1 al examinar más detenidamente.

Los errores surgieron en la computadora del estudiante #1, lo que resultó en tiempos de ejecución más largos a pesar de tener un hardware superior. Sin embargo, en ambos casos, los intervalos cumplen con las expectativas, ya que no comienzan ni terminan después de 1 segundo.

Para confirmar estos hallazgos, realizamos una prueba Z. Los resultados indican que la computadora del estudiante #3 es superior, ya que el valor p es menor a 0.05 en el valor crítico de z (dos colas). En este caso, se identificó un cuello de botella durante la creación del módulo de entrenamiento.

Estos resultados se obtuvieron tras ejecutar todas las pruebas creadas para las entidades training-module y training-session.

También tengo el análisis a la hora de correr mis test por VisualVM:

