

**Edgar Miralles Sosa**

**NUDATO SPEAKERS:  
GESTIÓN INTEGRAL PARA  
ORADORES EN EL MARCO DE  
LA ORGANIZACIÓN DE  
EVENTOS**

TRABAJO DE FIN DE GRADO

*Supervisado por*

ENRIQUE MOLINA GIMÉNEZ

GRADO DE INGENIERÍA INFORMÁTICA



UNIVERSITAT ROVIRA I VIRGILI

Tarragona  
2024

## Summary

In the context of event organization, organizers have encountered problems managing participants in an agile and efficient manner. The need for a Customer Relationship Management (CRM) [1] to centrally store participant data, including their sessions, email campaigns, and accreditations, led to the development of **Nudato**. To solve specific communication and management issues, a speaker module and a participant panel have been implemented within Nudato. These tools aim to minimize contact through less efficient external channels such as email.

The speaker module and participant panel allow users to access a web portal where they can request data changes, respond to articles, and update their personal information. All these requests are sent directly to the main administration panel, where organizers can review and approve them. This integration facilitates a more efficient communication model, improving the experience for both parties.

During development, a rigorous methodology based on software engineering principles was followed, analyzing all possible user interactions with the application. The implementation was designed using the Model-View-Controller (Model-View-Controller (MVC)) design pattern, leveraging advanced technologies such as Laravel [2] and Filament [3] to create efficient and reusable user interfaces. A multitenancy system has been applied to ensure the confidentiality and security of each client's data.

An agile approach was adopted in the project's development, allowing for rapid iterations and constant client feedback. The result is a solid and scalable solution for managing speakers and participants that enhances organizational efficiency.

## Resum

En el marc de la realització d'esdeveniments, els organitzadors han descobert problemes a l'hora de gestionar els participants de manera àgil i eficient. La necessitat d'un CRM [1] per emmagatzemar de manera centralitzada les dades dels participants, incloent les seves sessions, campanyes de correu i acreditacions, va portar al desenvolupament de **Nudato**. Per resoldre problemes específics de comunicació i gestió, s'ha implementat un mòdul de ponents i un panell de participants dins de Nudato. Aquestes eines busquen minimitzar el contacte a través de vies exteriors menys eficients com el correu electrònic.

El mòdul de ponents i el panell de participants permeten a aquests últims accedir a un portal web des del qual poden fer sol·licituds de canvi de dades, respondre a articles i actualitzar la seva informació personal. Totes aquestes sol·licituds s'envien directament al panell principal d'administració, on els organitzadors poden revisar-les i aprovar-les. Aquesta integració implica un model de comunicació més eficient, millorant l'experiència d'ambdues parts.

Durant el desenvolupament, s'ha seguit una metodologia rigorosa basada en principis d'enginyeria de programari, analitzant totes les possibles interaccions dels usuaris amb l'aplicatiu. La implementació ha estat dissenyada utilitzant el patró de disseny Model-Vista-Controlador (MVC), aprofitant tecnologies avançades com Laravel [2] i Filament [3] per a la creació d'interfícies d'usuari eficients i reutilitzables. S'ha aplicat un sistema de multitenència per assegurar

la confidencialitat i seguretat de les dades de cada client.

S'ha adoptat un enfocament àgil en el desenvolupament del projecte, permetent iteracions ràpides i la incorporació de feedback constant del client. El resultat ha estat una solució sòlida i ampliable per a la gestió de ponents i participants que millora l'eficiència organitzativa.

## Resumen

En el marco de la realización de eventos, los organizadores han descubierto problemas a la hora de gestionar los participantes de manera ágil y eficiente. La necesidad de un CRM [1] para almacenar de forma centralizada los datos de los participantes, incluyendo sus sesiones, campañas de correo y acreditaciones, llevó al desarrollo de **Nudato**. Para resolver problemas específicos de comunicación y gestión, se ha implementado un módulo de oradores y un panel de participantes dentro de Nudato. Estas herramientas buscan minimizar el contacto a través de vías exteriores menos eficientes como el correo electrónico.

El módulo de oradores y el panel de participantes permiten a estos últimos acceder a un portal web desde el cual pueden realizar peticiones de cambio de datos, responder a artículos y actualizar su información personal. Todas estas peticiones se envían directamente al panel principal de administración, donde los organizadores pueden revisarlas y aprobarlas. Esta integración implica un modelo de comunicación más eficiente, mejorando la experiencia de ambas partes.

Durante el desarrollo se ha seguido una metodología rigurosa basada en principios de ingeniería de software, analizando todas las posibles interacciones de los usuarios con el aplicativo. La implementación ha sido diseñada utilizando el patrón de diseño Modelo-Vista-Controlador (MVC), aprovechando tecnologías avanzadas como Laravel [2] y Filament [3] para la creación de interfaces de usuario eficientes y reutilizables. Se ha aplicado un sistema de multitenencia para asegurar la confidencialidad y seguridad de los datos de cada cliente.

Se ha adoptado un enfoque ágil en el desarrollo del proyecto, permitiendo iteraciones rápidas y la incorporación de feedback constante del cliente. El resultado ha sido una solución sólida y ampliable para la gestión de oradores y participantes que mejora la eficiencia organizativa.

# Índice general

<b>1. Introducción y objetivos</b>	<b>2</b>
<b>2. Contextualización de la plataforma</b>	<b>4</b>
2.1. Antecedentes . . . . .	4
2.2. Situación actual de Nudato . . . . .	4
2.3. Desafíos futuros y proyecciones del Trabajo de Fin de Grado . . . . .	5
<b>3. Requisitos</b>	<b>6</b>
3.1. Requisitos funcionales . . . . .	6
3.1.1. Guion del organizador . . . . .	6
3.1.2. Guion del orador . . . . .	7
3.1.3. Guion del asistente . . . . .	7
3.1.4. Glosario de requisitos . . . . .	7
3.1.5. Reglas de negocio . . . . .	7
3.1.6. Casos de uso . . . . .	9
3.2. Requisitos no Funcionales . . . . .	32
<b>4. Análisis de requisitos</b>	<b>33</b>
4.1. Diagrama de clases de entidades sin operaciones . . . . .	33
4.2. Traducción de actividades . . . . .	34
4.2.1. Caso de uso <u>01. VerOradores</u> . . . . .	34
4.2.2. Caso de uso <u>02. CrearOrador</u> . . . . .	36
4.2.3. Caso de uso <u>03. ModificarOrador</u> . . . . .	38
4.2.4. Caso de uso <u>04. BorrarOrador</u> . . . . .	40
4.2.5. Caso de uso <u>05. VerDetallesOrador</u> . . . . .	42
4.2.6. Caso de uso <u>06. ValidarPeticones</u> . . . . .	46
4.2.7. Caso de uso <u>07. ModificarInformación</u> . . . . .	48
4.2.8. Caso de uso <u>08. AñadirContactoVinculado</u> . . . . .	50
4.2.9. Caso de uso <u>09. BorrarSesion</u> . . . . .	52
4.2.10. Caso de uso <u>10. AñadirSesion</u> . . . . .	54
4.2.11. Caso de uso <u>11. ModificarDocumentos</u> . . . . .	56
4.2.12. Caso de uso <u>12. VerArtículos</u> . . . . .	59
4.2.13. Caso de uso <u>13. CrearArtículo</u> . . . . .	61
4.2.14. Caso de uso <u>14. VerDetallesArtículo</u> . . . . .	63
4.2.15. Caso de uso <u>15. ValidarRespuesta</u> . . . . .	65
4.2.16. Caso de uso <u>16. ValidarPeticiónFecha</u> . . . . .	67
4.2.17. Caso de uso <u>17. BorrarArtículo</u> . . . . .	70
4.2.18. Caso de uso <u>18. ModificarPetición</u> . . . . .	72

4.2.19.	Caso de uso <u>19. EnviarCorreo</u>	74
4.2.20.	Caso de uso <u>20. Login</u>	76
4.2.21.	Caso de uso <u>21. SeleccionarOrganizador</u>	78
4.2.22.	Caso de uso <u>22. SeleccionarEvento</u>	80
4.2.23.	Caso de uso <u>23. VerDocumentos</u>	82
4.2.24.	Caso de uso <u>24. ModificarDocumentos</u>	84
4.2.25.	Caso de uso <u>25. VerParticipación</u>	86
4.2.26.	Caso de uso <u>26. RechazarActividad</u>	88
4.2.27.	Caso de uso <u>27. ModificarParticipación</u>	90
4.2.28.	Caso de uso <u>28. VerSesiones</u>	92
4.2.29.	Caso de uso <u>29. ModificarDocumentosSesión</u>	94
4.2.30.	Caso de uso <u>30. VerArtículosOrador</u>	96
4.2.31.	Caso de uso <u>31. ResponderArtículo</u>	98
4.2.32.	Caso de uso <u>32. RechazarArtículo</u>	100
4.2.33.	Caso de uso <u>33. PedirAmpliaciónFechaArtículo</u>	102
4.2.34.	Caso de uso <u>34. VerInformaciónPersonal</u>	104
4.2.35.	Caso de uso <u>35. ModificarInformaciónPersonal</u>	106
4.2.36.	Caso de uso <u>36. ModificarAutorizaciónAsistente</u>	108
4.2.37.	Caso de uso <u>37. SeleccionarOrador</u>	110
4.3.	Diagrama de clases de entidades con operaciones	112
4.4.	Especificación de las operaciones de las clases de entidad	113
<b>5.</b>	<b>Diseño</b>	<b>116</b>
5.1.	Diseño de la arquitectura	116
5.1.1.	Arquitectura básica	116
5.1.2.	Multitenencia	117
5.1.3.	Despliegue <i>serverless</i>	117
5.2.	Diseño de la interfaz	119
5.2.1.	Listado de oradores	119
5.2.2.	Detalles del orador	123
5.2.3.	Listado de artículos	132
5.2.4.	Detalles del artículo	135
5.2.5.	Panel de participantes	139
5.3.	Diseño de la persistencia	146
5.3.1.	Diagrama entidad relación	146
5.3.2.	Especificación de las tablas	146
<b>6.</b>	<b>Implementación</b>	<b>150</b>
6.1.	Elección de tecnologías	150
6.1.1.	Arquitectura básica	150
6.1.2.	Implementación de la multitenencia	152
6.1.3.	Despliegue <i>serverless</i>	154
6.2.	Desafíos en el desarrollo	156
6.2.1.	Metodologías ágiles y configuración del entorno	156
6.2.2.	Eficiencia de peticiones en múltiples bases de datos	159
6.2.3.	Gestión de los documentos de los oradores	159
<b>7.</b>	<b>Evaluación</b>	<b>160</b>
7.1.	Listado de oradores	160

7.2. Detalles del orador . . . . .	162
7.2.1. Ventana de información . . . . .	163
7.2.2. Ventana de documentos y sesiones . . . . .	164
7.3. Revista . . . . .	166
7.4. Listado de artículos . . . . .	166
7.5. Detalles artículo (Caso de uso 14) . . . . .	168
7.6. Portal de participante . . . . .	169
7.7. Conclusión de las pruebas . . . . .	176
<b>8. Planificación y gestión de costes</b>	<b>178</b>
8.1. Planificación temporal . . . . .	178
8.2. Evaluación de costes . . . . .	179
<b>9. Legislación y protección de datos</b>	<b>180</b>
9.1. Medidas de protección . . . . .	180
<b>10. Conclusiones</b>	<b>181</b>

## Índice de cuadros

7.1. Tests de renderizado sobre el componente de listado de oradores. . . . .	160
7.2. Tests del caso de uso 01. VerOradores. . . . .	161
7.3. Tests del caso de uso 02. CrearOrador. . . . .	161
7.4. Tests del caso de uso 03. ModificarOrador. . . . .	161
7.5. Tests del caso de uso 04. BorrarOrador. . . . .	162
7.6. Tests del caso de uso 19. BorrarOrador. . . . .	162
7.7. Tests de renderizado sobre la página de detalles del orador. . . . .	162
7.8. Tests del caso de uso 05. VerDetallesOrador. . . . .	163
7.9. Tests del caso de uso 06. ValidarPeticones. . . . .	163
7.10. Tests del caso de uso 07. ModificarInformación. . . . .	163
7.11. Tests del caso de uso 08. AñadirContactoVinculado. . . . .	164
7.12. Tests de renderizado sobre la página de detalles del orador, ventana de sesiones y documentos. . . . .	164
7.13. Tests del caso de uso 09. BorrarSesion. . . . .	164
7.14. Tests del caso de uso 10. AñadirSesion. . . . .	165
7.15. Tests del caso de uso 11. ModificarDocumentos. . . . .	165
7.16. Tests de la creación de la revista. . . . .	166
7.17. Tests de renderizado sobre el componente de listado de artículos. . . . .	166
7.18. Tests del caso de uso 12. VerArtículos. . . . .	167
7.19. Tests del caso de uso 18. ModificarPetición. . . . .	167
7.20. Tests del caso de uso 17. BorrarArtículo. . . . .	167
7.21. Tests del caso de uso 19. EnviarCoreo. . . . .	167
7.22. Tests de renderizado sobre el componente de la página de detalles del artículo. . . . .	168
7.23. Tests del caso de uso 14. VerDetallesArtículo. . . . .	168
7.24. Tests de renderizado sobre el componente de respuesta del artículo. . . . .	168
7.25. Tests del caso de uso 14. VerDetallesArtículo parte 2. . . . .	169
7.26. Tests del caso de uso 15. ValidarRespuesta. . . . .	169
7.27. Tests del caso de uso 16. ValidarPeticiónFecha. . . . .	169
7.28. Tests del caso de uso 20. Login. . . . .	169
7.29. Tests de renderizado de la configuración de entorno. Casos de uso 21 y 22. . . . .	170
7.30. Tests de configuración del entorno. Casos de uso 21 y 22. . . . .	171
7.31. Tests de configuración del entorno. Casos de uso 21 y 22. Parte 2. . . . .	172
7.32. Tests renderizado caso de uso 25 . . . . .	172
7.33. Tests caso de uso 25 . . . . .	172
7.34. Tests caso de uso 26. RechazarActividad . . . . .	172
7.35. Tests caso de uso 27. ModificarParticipación . . . . .	173
7.36. Tests renderizado del panel de información . . . . .	173
7.37. Tests caso de uso 34. VerInformaciónPersonal . . . . .	173

7.38. Tests caso de uso 35. ModificarInformaciónPersonal . . . . .	173
7.39. Tests caso de uso 36. ModificarAutorizaciónAsistente . . . . .	174
7.40. Tests renderizado del panel de sesiones . . . . .	174
7.41. Tests renderizado del caso de uso 29. ModificarDocumentosSesión . . . . .	174
7.42. Tests renderizado del panel de documentos . . . . .	174
7.43. Tests renderizado del caso de uso 24. ModificarDocumentos . . . . .	175
7.44. Tests renderizado del panel de artículos . . . . .	175
7.45. Tests del caso de uso 30. VerArtículosOrador . . . . .	175
7.46. Tests del caso de uso 31. ResponderArtículo . . . . .	176
7.47. Tests del caso de uso 32. RechazarArtículo . . . . .	176
7.48. Tests del caso de uso 33. PedirAmpliaciónFechaArtículo . . . . .	176

## Índice de figuras

3.1. Diagrama de casos de uso del panel para organizadores de Nudato. . . . .	9
3.2. Diagrama de casos de uso del panel de participantes de Nudato. . . . .	10
4.1. Diagrama de casos de entidades sin operaciones del módulo de oradores de Nudato. . . . .	33
4.2. Diagrama de actividades del caso de uso 01. . . . .	34
4.3. Diagrama de clases del caso de uso 01. . . . .	34
4.4. Diagrama de secuencias del caso de uso 01. . . . .	35
4.5. Diagrama de actividades del caso de uso 02. . . . .	36
4.6. Diagrama de clases del caso de uso 02. . . . .	37
4.7. Diagrama de secuencias del caso de uso 02. . . . .	37
4.8. Diagrama de actividades del caso de uso 03. . . . .	38
4.9. Diagrama de clases del caso de uso 03. . . . .	39
4.10. Diagrama de secuencias del caso de uso 03. . . . .	39
4.11. Diagrama de actividades del caso de uso 04. . . . .	40
4.12. Diagrama de clases del caso de uso 04. . . . .	40
4.13. Diagrama de secuencias del caso de uso 04. . . . .	41
4.14. Diagrama de actividades del caso de uso 05. . . . .	42
4.15. Diagrama de clases del caso de uso 05. . . . .	43
4.16. Diagrama de secuencias del caso de uso 05. . . . .	44
4.17. Diagrama de secuencias del caso de uso 05. . . . .	45
4.18. Diagrama de actividades del caso de uso 06. . . . .	46
4.19. Diagrama de clases del caso de uso 06. . . . .	46
4.20. Diagrama de secuencias del caso de uso 06. . . . .	47
4.21. Diagrama de actividades del caso de uso 07. . . . .	48
4.22. Diagrama de clases del caso de uso 07. . . . .	48
4.23. Diagrama de secuencias del caso de uso 07. . . . .	49
4.24. Diagrama de actividades del caso de uso 08. . . . .	50
4.25. Diagrama de clases del caso de uso 08. . . . .	50
4.26. Diagrama de secuencias del caso de uso 08. . . . .	51
4.27. Diagrama de actividades del caso de uso 09. . . . .	52
4.28. Diagrama de clases del caso de uso 09. . . . .	52
4.29. Diagrama de secuencias del caso de uso 09. . . . .	53
4.30. Diagrama de actividades del caso de uso 10. . . . .	54
4.31. Diagrama de clases del caso de uso 10. . . . .	54
4.32. Diagrama de secuencias del caso de uso 10. . . . .	55
4.33. Diagrama de actividades del caso de uso 11. . . . .	56
4.34. Diagrama de clases del caso de uso 11. . . . .	57

4.35. Diagrama de secuencias del caso de uso 11. . . . .	58
4.36. Diagrama de actividades del caso de uso 12. . . . .	59
4.37. Diagrama de clases del caso de uso 12. . . . .	59
4.38. Diagrama de secuencias del caso de uso 12. . . . .	60
4.39. Diagrama de actividades del caso de uso 13. . . . .	61
4.40. Diagrama de clases del caso de uso 13. . . . .	61
4.41. Diagrama de secuencias del caso de uso 13. . . . .	62
4.42. Diagrama de actividades del caso 14. . . . .	63
4.43. Diagrama de clases del caso 14. . . . .	63
4.44. Diagrama de secuencias del caso 14. . . . .	64
4.45. Diagrama de actividades del caso de uso 15. . . . .	65
4.46. Diagrama de clases del caso de uso 15. . . . .	65
4.47. Diagrama de secuencias del caso de uso 15. . . . .	66
4.48. Diagrama de actividades del caso de 16. . . . .	67
4.49. Diagrama de clases del caso de 16. . . . .	68
4.50. Diagrama de secuencias del caso de 16. . . . .	69
4.51. Diagrama de actividades del caso de uso 17. . . . .	70
4.52. Diagrama de clases del caso de uso 17. . . . .	70
4.53. Diagrama de secuencias del caso de uso 17. . . . .	71
4.54. Diagrama de actividades del caso de uso 18. . . . .	72
4.55. Diagrama de clases del caso de uso 18. . . . .	72
4.56. Diagrama de secuencias del caso de uso 18. . . . .	73
4.57. Diagrama de actividades del caso de uso 19. . . . .	74
4.58. Diagrama de clases del caso de uso 19. . . . .	74
4.59. Diagrama de secuencias del caso de uso 19. . . . .	75
4.60. Diagrama de actividades del caso de uso 20. . . . .	76
4.61. Diagrama de clases del caso de uso 20. . . . .	77
4.62. Diagrama de secuencias del caso de uso 20. . . . .	77
4.63. Diagrama de actividades del caso de uso 21. . . . .	78
4.64. Diagrama de clases del caso de uso 21. . . . .	78
4.65. Diagrama de secuencias del caso de uso 21. . . . .	79
4.66. Diagrama de actividades del caso de uso 22. . . . .	80
4.67. Diagrama de clases del caso de uso 22. . . . .	80
4.68. Diagrama de secuencias del caso de uso 22. . . . .	81
4.69. Diagrama de actividades del caso de uso 23. . . . .	82
4.70. Diagrama de clases del caso de uso 23. . . . .	82
4.71. Diagrama de secuencias del caso de uso 23. . . . .	83
4.72. Diagrama de actividades del caso de uso 24. . . . .	84
4.73. Diagrama de clases del caso de uso 24. . . . .	84
4.74. Diagrama de secuencias del caso de uso 24. . . . .	85
4.75. Diagrama de actividades del caso de uso 25. . . . .	86
4.76. Diagrama de clases del caso de uso 25. . . . .	86
4.77. Diagrama de secuencias del caso de uso 25. . . . .	87
4.78. Diagrama de actividades del caso de uso 26. . . . .	88
4.79. Diagrama de clases del caso de uso 26. . . . .	88
4.80. Diagrama de secuencias del caso de uso 26. . . . .	89
4.81. Diagrama de actividades del caso de uso 27. . . . .	90
4.82. Diagrama de clases del caso de uso 27. . . . .	90

4.83. Diagrama de secuencias del caso de uso 27. . . . .	91
4.84. Diagrama de actividades del caso de uso 28. . . . .	92
4.85. Diagrama de clases del caso de uso 28. . . . .	92
4.86. Diagrama de secuencias del caso de uso 28. . . . .	93
4.87. Diagrama de actividades del caso de uso 29. . . . .	94
4.88. Diagrama de clases del caso de uso 29. . . . .	94
4.89. Diagrama de secuencias del caso de uso 29. . . . .	95
4.90. Diagrama de actividades del caso de uso 30. . . . .	96
4.91. Diagrama de clases del caso de uso 30. . . . .	96
4.92. Diagrama de secuencias del caso de uso 30. . . . .	97
4.93. Diagrama de actividades del caso de uso 31. . . . .	98
4.94. Diagrama de clases del caso de uso 31. . . . .	98
4.95. Diagrama de secuencias del caso de uso 31. . . . .	99
4.96. Diagrama de actividades del caso de uso 32. . . . .	100
4.97. Diagrama de clases del caso de uso 32. . . . .	100
4.98. Diagrama de secuencias del caso de uso 32. . . . .	101
4.99. Diagrama de actividades del caso de uso 33. . . . .	102
4.100 Diagrama de clases del caso de uso 33. . . . .	102
4.101 Diagrama de secuencias del caso de uso 33. . . . .	103
4.102 Diagrama de actividades del caso de uso 34. . . . .	104
4.103 Diagrama de clases del caso de uso 34. . . . .	104
4.104 Diagrama de secuencias del caso de uso 34. . . . .	105
4.105 Diagrama de actividades del caso de uso 35. . . . .	106
4.106 Diagrama de clases del caso de uso 35. . . . .	106
4.107 Diagrama de secuencias del caso de uso 35. . . . .	107
4.108 Diagrama de actividades del caso de uso 36. . . . .	108
4.109 Diagrama de clases del caso de uso 36. . . . .	108
4.110 Diagrama de secuencias del caso de uso 36. . . . .	109
4.111 Diagrama de actividades del caso de uso 37. . . . .	110
4.112 Diagrama de clases del caso de uso 37. . . . .	110
4.113 Diagrama de secuencias del caso de uso 37. . . . .	111
4.114 Diagrama de casos de entidades sin operaciones del módulo de oradores de Nudato. . . . .	112
5.1. Interfaz del listado de oradores. . . . .	119
5.2. Acciones en masa disponibles al seleccionar uno o varios oradores. . . . .	119
5.3. Listado de Oradores con peticiones pendientes de revisión. . . . .	120
5.4. Filtros para oradores. . . . .	120
5.5. Modal de edición del orador del Caso de uso 03. ModificarOrador. . . . .	121
5.6. Modal de confirmación de borrado. . . . .	122
5.7. Modal de confirmación de borrado. . . . .	122
5.8. Ventana de información de los detalles del orador. . . . .	123
5.9. Vista de validación de los datos de contacto del participante. . . . .	124
5.10. Vista de validación de los documentos de una sesión del orador. . . . .	124
5.11. Carta de información de contacto antes y después de iniciar el caso de uso 07. ModificarInformación. . . . .	125
5.12. Carta de información de participación antes y después de iniciar el caso de uso 07. ModificarInformación. . . . .	126

5.13. Carta de contactos vinculados antes y después de iniciar el caso de uso 08. AñadirContactoVinculado. . . . .	126
5.14. Ventana de sesiones y documentos del orador. . . . .	127
5.15. Modal de confirmación de borrado de sesión. . . . .	128
5.16. Modal de creación de sesión. . . . .	129
5.17. Modal de añadir sesión existente. . . . .	129
5.18. Modal de añadir sesión existente. . . . .	130
5.19. Modal con los documentos disponibles para ser añadidos a la sesión. . . . .	130
5.20. Documento en estado de <i>hover</i> , mostrando la basura para ser eliminado de la sesión. . . . .	131
5.21. Interfaz de creación de una sección. . . . .	132
5.22. Modal de creación de un artículo. . . . .	133
5.23. Interfaz del listado de artículos. . . . .	133
5.24. Acciones en masa disponibles al seleccionar uno o varios artículos. . . . .	134
5.25. Listado de artículos con nuevas versiones y rechazos. . . . .	134
5.26. Filtros para artículos. . . . .	135
5.27. Ventana de configuración de la petición del artículo. . . . .	136
5.28. Ventana de respuesta de un artículo. . . . .	137
5.29. Correo de retroalimentación de validación de respuesta. . . . .	137
5.30. Petición de cambio de fecha de un artículo. . . . .	138
5.31. Modal de validación de la fecha al denegar una petición y tras activar la acción de dar nueva fecha. . . . .	138
5.32. Correo de retroalimentación por respuesta a petición de cambio de fecha. . . . .	138
5.33. Ventana de login. . . . .	139
5.34. Selección de evento desde el panel de participación. . . . .	139
5.35. Configuración de entorno desde el menú desplegable. . . . .	139
5.36. Panel de participación. . . . .	140
5.37. Panel de información. . . . .	141
5.38. Modal de cambio de información en un campo. . . . .	141
5.39. Panel de sesiones. . . . .	142
5.40. Panel de documentos. . . . .	142
5.41. Panel de artículos. . . . .	143
5.42. Formulario de respuesta de un artículo. . . . .	143
5.43. Correo de confirmación del envío de un artículo. . . . .	144
5.44. Modal de confirmación para el rechazo de un artículo. . . . .	144
5.45. Modal de cambio de fecha de entrega de un artículo. . . . .	145
5.46. Diagrama entidad relación del proyecto. . . . .	146
6.1. Diagrama de flujo de la configuración de entorno con los posibles errores repre- sentados. . . . .	158
8.1. Diagrama de Gantt del proyecto. . . . .	178

## **Siglas**

**AWS** Amazon Web Services. 154, 156

**CRM** Customer Relationship Management. 1, 2

**CRUD** Create, Read, Update, Delete. 116, 150, 179

**CSS** Cascading Style Sheets. 116, 152

**HTML** Hypertext Markup Language. 116, 152

**HTTP** Hypertext Transfer Protocol. 117, 154

**MVC** Model-View-Controller. 1, 2, 116, 118, 150, 152

**ORM** Object-Relational Mapping. 150, 152

**PHP** Hypertext Preprocessor. 150

**SQL** Structured Query Language. 146

# 1 Introducción y objetivos

La organización de eventos presenta una gran cantidad de desafíos relacionados con la gestión eficiente de los registros, coordinándolos con todas las actividades disponibles (reserva de hoteles, debates, conferencias, incluso la conexión a la red inalámbrica del evento debe ser tenida en cuenta). Los organizadores necesitan disponer de información sobre los asistentes de manera centralizada para poder resolver problemas, gestionar las acreditaciones, ejecutar campañas de comunicación, etc. En este contexto, Nudato surge como una solución integral, diseñada para optimizar todos estos críticos mediante una misma plataforma.

Nudato ofrece herramientas para la gestión de registros, impresión de acreditaciones, creación y procesamiento de formularios, análisis de estadísticas en tiempo real y la ejecución de campañas de correo electrónico. Esta solución todo-en-uno busca minimizar la dependencia de herramientas desconectadas y procesos manuales, proporcionando una mayor eficiencia y precisión en la organización de eventos.

Dentro del marco de Nudato, este proyecto se centra en el desarrollo de un módulo específico para satisfacer las demandas de los oradores y automatizar las tareas administrativas que enfrentan los organizadores. Los oradores, presentes en la mayoría de eventos, son responsables de presentaciones, discursos o debates. Por otro lado, los organizadores son los encargados de la planificación y coordinación general del evento, y requieren herramientas que simplifiquen la gestión de la información y mejoren la comunicación con los participantes.

El propósito del proyecto ha sido la integración de un nuevo módulo dentro de la aplicación web principal, contribuyendo así en la transformación digital en el sector de la organización de eventos. Este objetivo ha requerido un análisis exhaustivo de las necesidades de todos los usuarios implicados, tanto oradores como organizadores. Estos objetivos han requerido un análisis exhaustivo de las necesidades de todos los usuarios implicados, tanto oradores como organizadores.

Para los oradores, se ha desarrollado un espacio privado donde pueden gestionar su participación: modificar datos personales, preparar sesiones o debates, y gestionar artículos. Para los organizadores, se han priorizado herramientas que permiten la validación automática de datos y minimizan la necesidad de comunicación directa, sustituyendo métodos convencionales como llamadas telefónicas o correos electrónicos.

La implementación de este módulo ha implicado todas las etapas necesarias del desarrollo de software, desde la planificación temporal y de costes, hasta el diseño, implementación de la lógica de negocio y pruebas. Todo ello siguiendo metodologías ágiles para favorecer la evolución continua del producto mediante revisiones regulares con el cliente.

Desde una perspectiva formativa, este proyecto ha sido una excelente oportunidad para aplicar y profundizar en los conocimientos adquiridos a lo largo del Grado en Ingeniería Informática. Se

ha puesto especial énfasis en aspectos fundamentales como la arquitectura de datos, la comunicación entre cliente y servidor, el uso de patrones de diseño, la implementación de algoritmos eficientes, el manejo de estructuras de datos, el diseño de interfaces y la planificación de pruebas. Este enfoque integral ha permitido no solo consolidar el aprendizaje teórico, sino también expandirlo mediante la aplicación práctica en un proyecto real.

Para llevar a cabo el desarrollo del proyecto, se ha seguido la siguiente estructura:

- **Sección 1:** Introducción y objetivos del proyecto.
- **Sección 2:** Contextualización de la plataforma, donde se detalla el origen y el estado actual y futuro de Nudato.
- **Sección 3:** Requisitos funcionales y no funcionales del módulo desarrollado.
- **Sección 4:** Análisis de requisitos, incluyendo diagramas de clases y casos de uso.
- **Sección 5:** Diseño de la arquitectura, interfaz y persistencia.
- **Sección 6:** Implementación, desafíos enfrentados y tecnologías utilizadas.
- **Sección 7:** Evaluación del módulo desarrollado.
- **Sección 8:** Evaluación de costes.
- **Sección 9:** Legislación y protección de datos.
- **Sección 10:** Conclusiones y futuras líneas de trabajo.

## **2 Contextualización de la plataforma**

### **2.1. Antecedentes**

La organización de eventos siempre ha sido una tarea compleja que ha requerido de la gestión de múltiples procesos críticos. Entre otros, la gestión de los registros y sus estados, la producción de acreditaciones personalizadas, la ejecución de campañas de correo y la recolección continua de datos se destacan como procesos fundamentales pero complicados. Antes de la era de la transformación digital, la gestión del evento dependía en gran medida de procesos manuales y sistemas de papel. Esto resultaba una tarea tediosa y que requería de una gran cantidad de horas, limitando la escalabilidad de los eventos. Con la digitalización, surgieron diversas soluciones en forma de software que buscaban aliviar estos problemas. Aun así, estas soluciones solo se centraban en aspectos individuales, como la distribución de correos. Este hecho conlleva la necesidad de usar múltiples herramientas para tratar los mismos datos, aumentando en gran medida la probabilidad de error y de incoherencia entre las fuentes.

### **2.2. Situación actual de Nudato**

Nudato representa una solución integral a todas estas necesidades que antes se enfrentaban mediante herramientas desconectadas. Esta solución todo-en-uno ofrece desde la gestión de contactos y registros para múltiples eventos, hasta la creación de plantillas de correo y acreditaciones, facilitando la comunicación efectiva y personalizada con los asistentes. Además, su capacidad para ejecutar campañas de correo masivo y procesar datos de formularios en tiempo real asegura una recopilación de información fluida y precisa, que se traduce en registros actualizados en la base de datos del evento.

Con Nudato, los organizadores tienen a su disposición herramientas avanzadas para la creación de estadísticas personalizadas, permitiendo un análisis detallado y en tiempo real del progreso y alcance de sus eventos. La plataforma simplifica aspectos logísticos como las reservas de hoteles y el almacenamiento de documentos importantes, garantizando que la información esencial esté accesible y bien organizada. Asimismo, la gestión de actividades o sesiones y el uso de un calendario interactivo contribuyen a una planificación eficaz y a una experiencia del usuario excepcional.

## **2.3. Desafíos futuros y proyecciones del Trabajo de Fin de Grado**

A pesar de las amplias funcionalidades que Nudato ofrece para la gestión integral de eventos, se ha identificado un área de oportunidad significativa en la comunicación y gestión de información de los participantes, especialmente de los oradores. Tradicionalmente, la actualización de datos personales, la organización de sesiones y la preparación de materiales requerían que los oradores contactaran directamente con los organizadores del evento mediante correo electrónico o llamadas telefónicas. Este proceso no solo resultaba tedioso y costoso a nivel temporal para ambas partes, sino que también era propenso a errores debido a la necesidad de introducir los cambios manualmente en el sistema. Además, la falta de un mecanismo de confirmación inmediata dejaba a los oradores sin la seguridad de que sus solicitudes habían sido atendidas correctamente.

La integración del módulo introducido en este proyecto busca seguir la línea planteada en el resto de la aplicación, ampliando el alcance de esta más allá de la visión del panel administrativo. Esta expansión añade una nueva visión dirigida a los participantes del evento. Por ejemplo, los oradores ahora tendrán la capacidad de generar solicitudes directamente desde su espacio privado en Nudato, permitiéndoles solicitar cambios en sus datos personales, enviar propuestas para sesiones, y cargar materiales relevantes como presentaciones o artículos. Estas solicitudes serán automáticamente dirigidas al panel administrativo, donde los organizadores podrán procesarlas de manera eficiente sin necesidad de comunicación directa adicional. Esta funcionalidad no solo simplificará el proceso para los oradores, sino que también mejorará la precisión y la rapidez en la gestión de sus solicitudes.

Además, esta integración proporcionará un mecanismo de confirmación inmediata para los oradores. Una vez que envíen sus solicitudes a través de Nudato, recibirán notificaciones automáticas sobre el estado de sus solicitudes, brindándoles la tranquilidad de que sus necesidades están siendo atendidas de manera oportuna y precisa.

Con estas mejoras, Nudato transformará la experiencia de participación de los oradores al proporcionarles herramientas intuitivas y eficientes para gestionar su participación en eventos, al tiempo que optimiza la comunicación y colaboración entre oradores y organizadores.

## **3 Requisitos**

En este capítulo se describirán los requisitos funcionales y no funcionales que el sistema debe cumplir. Para facilitar la comprensión de los requisitos, se incluirá un glosario de los términos usados...

### **3.1. Requisitos funcionales**

Los requisitos funcionales describen el comportamiento y las funcionalidades específicas que el sistema debe tener. A continuación, se detallan los guiones de los principales actores del sistema: el organizador, el orador y el asistente. A partir de estos guiones, se extraerán las funcionalidades que debe tener la aplicación.

#### **3.1.1. Guion del organizador**

Administra registros al evento con el rol de orador (crear, borrar y modificar), vinculándolo a un contacto existente en la base de datos y añadiéndole como mínimo un estado (invitado, registrado, confirmado, rechazado, etc.). De forma opcional, debe poder rellenar otros datos referentes al registro como el hotel, el número de noches, la asistencia, actividades del evento, información de los vuelos y tantos requisitos especiales como se necesiten. Una vez se ha completado el registro, el organizador accede a una página de detalles donde debe poder visualizar y modificar los datos personales y sobre la participación del orador. Además, si el orador ha hecho alguna petición de cambios, se mostrará una interfaz de validación al organizador. Desde la página de detalles, debe poder cambiar la ficha de contacto y su foto de perfil; Puede añadir contactos relacionados en forma de asistente y copy to; Puede modificar los datos de participación (hotel, vuelos, requisitos especiales, actividades, estado, etc.); Puede visualizar las sesiones del orador y todos los otros oradores que participan. Además, debe poder crear o añadir sesiones, modificar el rol de los oradores existentes y añadir nuevos. Finalmente, en la misma página, el administrador puede ver y modificar los documentos del orador y asignarlos a las sesiones de este; Adicionalmente, el organizador puede administrar la revista del evento, que consta de secciones, subsecciones y artículos que deben poder tener un orden asignable. El organizador puede hacer peticiones de envío de artículos al orador. Los artículos tienen principalmente un tema, una fecha de entrega, una longitud recomendada, un estado y un número entero que representa su orden. Debe poder visualizar una tabla ordenada según el orden de las secciones, subsecciones y artículos. En la tabla se utilizará una de las columnas para mostrar notificaciones según el estado del artículo: nueva versión, petición de cambio de fecha...

### 3.1.2. Guion del orador

Accede con su correo de registro al panel de participantes. Una vez dentro, elige una de las compañías organizadoras y el evento al que asiste. Si solo hay una empresa, se selecciona por defecto. Si solo hay un evento, se selecciona también por defecto. Una vez hecha la selección, podrá ver diferentes secciones y visualizar sus datos y realizar los cambios pertinentes. Podrá acceder a una sección inicial que resuma la participación al evento. En esta podrá cancelar sus actividades, añadir requisitos especiales o cambiar sus datos de vuelos; Podrá acceder a un panel desde el que ver todas las sesiones en las que participa y añadir sus documentos; Podrá acceder a un panel desde el cual gestionar todos sus documentos; También podrá acceder a una sección que le permite realizar una petición de modificación de sus datos personales; Finalmente podrá acceder a un panel de artículos donde podrá visualizar y editar sus artículos para el evento, añadiendo un título, la respuesta, una frase clave, una imagen y rellenar un campo en el que especificar correos a los que también les llegará la confirmación del artículo. En cuanto a esta confirmación, se mandará automáticamente a los contactos que tenga vinculados el orador y se requerirá una validación por parte de la organización en el caso de los correos no vinculados.

### 3.1.3. Guion del asistente

Si el orador ha dado acceso a su asistente, este puede acceder al panel y ver toda su información y realizar las mismas acciones que realizaría el propio orador.

### 3.1.4. Glosario de requisitos

**Administrador de Documentos:** plug-in de Filament [3] que proporciona un gestor de documentos que permite subir, modificar, visualizar y seleccionar archivos. [4]

**Artículo:** documentos escritos por un orador hablando de algún tema solicitado por el personal organizativo.

**Asistente y Copia:** los contactos en Nudato pueden tener relaciones con otros contactos. Esta relación puede ser de Asistente/Jefe o Copia

**Portal:** espacio privado al que accede el orador con su correo de registro y en el que puede visualizar y modificar sus datos de participación al evento.

**Requisitos Especiales:** comentarios introducidos por parte de un orador para comunicar a la organización del evento alguna situación excepcional.

**Sesión:** segmento de tiempo en la agenda de un evento, dedicado a tratar temas específicos o a la presentación de ideas por parte de uno o varios oradores expertos en el campo.

### 3.1.5. Reglas de negocio

Se han separado las reglas de negocio en dos grupos correspondientes a la aplicación para oradores y al panel administrativo de la organización: **Grupo 1. Oradores**

1. Un orador solo puede acceder al **portal** si está registrado al menos a un evento de la compañía organizadora.

2. Un asistente solo puede acceder al **portal** si recibe la autorización expresa del orador relacionado.
3. Un orador solo puede acceder a información sobre eventos a los que está registrado.
4. Todo cambio en la información personal del orador debe ser validado por un miembro del personal organizativo.
5. Un orador puede añadir tantos **requisitos especiales** como considere, pero nunca vacíos.
6. Un orador no puede ver los documentos subidos por los otros oradores, aunque el **administrador de documentos** utilizado sea global.
7. Los documentos añadidos a una sesión deben ser validados por un organizador del evento.
8. Los oradores solo pueden rechazar un **artículo** si aún no han escrito nada.
9. Los oradores solo pueden solicitar una ampliación de fecha a dos días de la entrega del **artículo**.
10. Si un orador pone en **copia** a un correo que no tenga vinculado en su ficha de contacto, deberá ser revisado por un miembro del personal administrativo.
11. Una vez rechazado, un orador no puede realizar ninguna acción sobre un **artículo**.

## **Grupo 2. Personal administrativo**

1. Solo se puede crear un **orador** a partir de un **contacto** existente de Nudato.
2. Solo se pueden añadir a **asistentes** o **copia** a un contacto ya registrado en Nudato.
3. No se pueden añadir **requisitos especiales** vacíos.
4. No se puede añadir a un orador a una **sesión**
  - La fecha de inicio cae entre la fecha de inicio o fin de otra **sesión** del orador.
  - La fecha de fin cae entre la fecha de inicio o fin de otra **sesión** del orador.
5. No se puede establecer una fecha de sesión fuera de la fecha que comprende el evento.



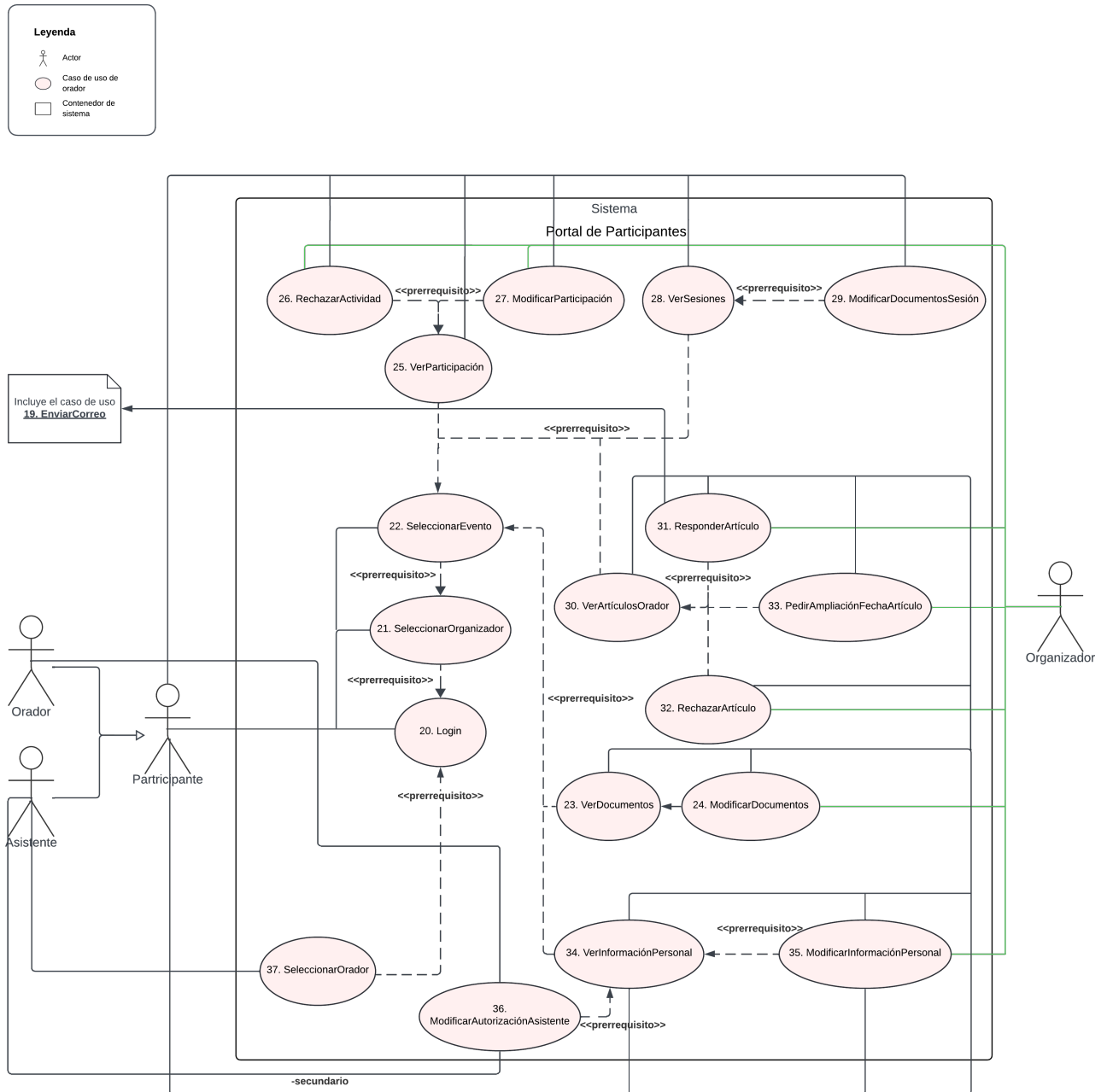


Figura 3.2: Diagrama de casos de uso del panel de participantes de Nudato.

## Especificación textual de casos de uso

### Caso de uso 01. VerOradores

*Resumen de la funcionalidad:* presenta los oradores del evento, deja seleccionarlos para ejecutar las acciones de borrar, modificar y ver detalles y permite aplicar filtros.

*Parámetros de entrada:* filtros de búsqueda.

*Parámetros de salida:* lista de oradores.

*Actores:* **Organizador.**

*Precondición:* el módulo de oradores está activo en el sistema.

*Postcondición:* se muestra una lista de oradores sobre los que realizar las operaciones de gestión.

*Proceso normal principal:*

1. El sistema lee todos los oradores del evento seleccionado.
2. El sistema presenta el resultado en una tabla.

*Alternativas de proceso y excepciones:*

- 1a. El **Organizador** ha aplicado algún filtro de búsqueda.
  - 1a1. El sistema lee todos los oradores del evento seleccionado aplicando los filtros de búsqueda proporcionados por el **Organizador**.
  - 1a2. El sistema presenta el resultado en una lista.
- 2a. El **Organizador** selecciona uno o más oradores y ejecuta la acción de borrar.
  - 2a1. El sistema ejecuta el caso de uso 04. BorrarOrador con el/los oradores seleccionados.
- 2b. El **Organizador** selecciona algún orador y ejecuta la acción de modificar.
  - 2b1. El sistema ejecuta el caso de uso 03. ModificarOrador con el orador seleccionado.
- 2c. El **Organizador** ejecuta la acción de crear orador.
  - 2c1. El sistema ejecuta el caso de uso 02. CrearOrador con el orador seleccionado.
- 2d. El **Organizador** selecciona algún orador y ejecuta la acción de ver detalles.
  - 2d1. El sistema ejecuta el caso de uso 05. VerDetallesOrador con el orador seleccionado.
- 2e. El **Organizador** selecciona varios oradores y ejecuta la acción de enviar correo al **Orador**.
  - 2e1. El sistema ejecuta el caso de uso 19. EnviarCorreo.

*Caso de uso 02. CrearOrador*

*Resumen de la funcionalidad:* añade un registro con el rol de orador a la base de datos con los datos introducidos por el **Organizador**.

*Parámetros de entrada:* ninguno.

*Parámetros de salida:* el orador creado.

*Actores:* **Organizador**.

*Precondición:* ninguna.

*Postcondición:* un orador ha sido añadido a la base de datos.

*Proceso normal principal:*

1. El sistema muestra una ventana pidiendo los datos de la *Lista 02.1*, cumpliendo con las reglas de negocio 01, 03 y 04.
2. El **Organizador** rellena los datos.
3. El sistema guarda el nuevo orador en la base de datos.
4. El sistema vuelve al caso de uso 01. VerOradores

*Alternativas de proceso y excepciones:*

- 2a. El **Organizador** selecciona la opción de añadir sesiones
  - 2a1. El sistema ejecuta el caso de uso 10. AñadirSesión sin permitir crear una nueva sesión.
- 2b. El **Organizador** selecciona la opción de añadir documentos
  - 2b1. El sistema ejecuta el caso de uso 11. ModificarDocumentos.

### *Lista 02.1*

- Contacto de Nudato
- Fecha de confirmación
- Fecha de acreditación
- Registro 'on site'
- Visible en web
- Estado del registro
- Hotel
- Número de Noches
- Actividades
- Información de vuelo
- Plantilla de acreditación
- Sesiones en las que participa
- Requisitos especiales
- Documentos del orador

### *Caso de uso 03. ModificarOrador*

*Resumen de la funcionalidad:* modifica un orador existente en la base de datos con los datos introducidos por el **Organizador**.

*Parámetros de entrada:* el orador seleccionado.

*Parámetros de salida:* el orador modificado.

*Actores:* **Organizador**.

*Precondición:* existe el orador en la base de datos.

*Postcondición:* el orador se ha modificado en la base de datos

*Proceso normal principal:*

1. El sistema muestra una ventana con campos según la *Lista 02.1*, cumpliendo con las reglas de negocio 01, 03 y 04 y rellenado con los datos del orador existente.
2. El **Organizador** modifica los datos que considere.
3. El sistema modifica el orador en la base de datos.
4. El sistema vuelve al caso de uso 01. VerOradores
- 2a. El **Organizador** selecciona la opción de añadir sesiones
  - 2a1. El sistema ejecuta el caso de uso 10. AñadirSesión sin permitir crear una nueva sesión.
  - 2b. El **Organizador** selecciona la opción de añadir documentos
    - 2b1. El sistema ejecuta el caso de uso 11. ModificarDocumentos.

### *Caso de uso 04. BorrarOrador*

*Resumen de la funcionalidad:* elimina uno o varios oradores existentes en la base de datos.

*Parámetros de entrada:* los oradores seleccionados.

*Parámetros de salida:* ninguno.

*Actores:* **Organizador**

*Precondición:* existen los oradores en la base de datos.

*Postcondición:* los oradores se han eliminado de la base de datos.

*Proceso normal principal:*

1. El sistema muestra una ventana de confirmación de la acción.
  2. El **Organizador** confirma la acción.
  3. El sistema elimina los oradores de la base de datos.
  4. El sistema vuelve al caso de uso 01. VerOradores.
- 2a. El **Organizador** cancela la acción.
- 2a1. El sistema vuelve al caso de uso 01. VerOradores.

*Caso de uso 05. VerDetallesOrador*

*Resumen de la funcionalidad:* presenta una ficha interactiva del orador que permite cambiar entre las vistas descritas en la *Lista 05.1*. Deja interactuar sobre todos los componentes de información para realizar acciones de modificación.

*Parámetros de entrada:* el orador seleccionado.

*Parámetros de salida:* ninguno.

*Actores:* **Organizador**.

*Precondición:* el orador existe en la base de datos.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recolecta toda la información personal y de participación (primera vista).
2. El sistema muestra la vista de información personal y participación junto a un componente para cambiar entre las ventanas de la *Lista 05.1*

*Alternativas de proceso y excepciones:*

- 1a. Hay peticiones pendientes de validar.
  - 1a1. El sistema ejecuta el caso de uso 06. ValidarPeticiones.
- 2a. El **Organizador** selecciona la opción de editar información personal.
  - 2a1. El sistema ejecuta el caso de uso 07. ModificarInformación para modificar la información personal.
- 2b. El **Organizador** selecciona la opción de vincular un nuevo contacto.
  - 2b1. El sistema ejecuta el caso de uso 08. AñadirContactoVinculado.
- 2c. El **Organizador** selecciona la ventana de sesiones y documentos.
  - 2c1. El sistema carga los datos de sesiones y documentos.
  - 2c2. El sistema carga presenta la vista de sesiones y documentos.
  - 2c3. El **Organizador** selecciona la opción de añadir sesión.
    - 2c3.1. El sistema ejecuta el caso de uso 10. AñadirSesion.
  - 2c4. El **Organizador** selecciona la opción de borrar sesión.
    - 2c4.1. El sistema ejecuta el caso de uso 09. BorrarSesion.
  - 2c5. El **Organizador** prueba a hacer cualquier modificación en sus documentos
    - 2c5.1. El sistema ejecuta el caso de uso 11. ModificarDocumentos.

- 2d. El **Organizador** selecciona la ventana de artículos.
  - 2d1. El sistema carga los datos de sesiones y documentos.
  - 2d2. El sistema carga presenta la vista de sesiones y documentos.
  - 2d3. El sistema ejecuta el caso de uso 12. VerArtículos, pasando el orador como parámetro.
- 2e. El **Organizador** selecciona la opción de enviar correo al **Orador**.
  - 2e1. El sistema ejecuta el caso de uso 19. EnviarCorreo.

#### *Lista 02.1*

- Información personal y de participación.
- Sesiones y documentos.
- Artículos.

#### Caso de uso 06. ValidarPeticiones

*Resumen de la funcionalidad:* compara los datos actuales del orador con los de la petición hecha por el propio participante en su panel. Permite aceptar o rechazar los cambios.

*Parámetros de entrada:* las peticiones del participante y el propio participante.

*Parámetros de salida:* el orador modificado.

*Actores:* **Organizador**

*Precondición:* hay una petición del orador.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recoge la petición.
2. El sistema presenta la comparación entre la petición y los datos actuales.
3. El **Organizador** valida los datos.
4. El **Organizador** guarda los cambios.
5. El sistema borra la petición de la base de datos.
6. El sistema actualiza los datos del participante en la base de datos.
7. El sistema vuelve al caso de uso 05. VerDetallesOrador.

*Alternativas de proceso y excepciones:*

- 3a. El **Organizador** esconde la ventana de validación.
  - 3a1. El sistema vuelve al caso de uso 05. VerDetallesOrador.

#### Caso de uso 07. ModificarInformación

*Resumen de la funcionalidad:* modifica los datos personales y de participación del orador en el componente interactivo.

*Parámetros de entrada:* el orador seleccionado.

*Parámetros de salida:* el orador modificado.

*Actores:* **Organizador**

*Precondición:* ninguna.

*Postcondición:* el orador se ha modificado según los nuevos datos introducidos.

*Proceso normal principal:*

1. El sistema cambia la carta de información para mostrar campos editables.

2. El **Organizador** modifica los datos pertinentes.
3. El **Organizador** guarda los datos.
4. El sistema valida los datos introducidos por el **Organizador**.
5. El sistema modifica los datos del orador en la base de datos.
6. El sistema vuelve al caso de uso 05. VerDetallesOrador.

*Alternativas de proceso y excepciones:*

- 5a. El sistema detecta que uno o más campos son incorrectos.
  - 5a1. El sistema muestra un mensaje de error por cada campo.
  - 5a2. El sistema vuelve al paso 1.

Caso de uso 08. AñadirContactoVinculado

*Resumen de la funcionalidad:* crea una relación entre el orador seleccionado y otro contacto existente en la base de datos.

*Parámetros de entrada:* el orador seleccionado.

*Parámetros de salida:* los contactos relacionados.

*Actores:* **Organizador**

*Precondición:* ninguna.

*Postcondición:* se ha establecido la relación entre contactos en la base de datos.

*Proceso normal principal:*

1. El sistema busca todos los contactos existentes en la base de datos.
2. El **Organizador** elige un contacto.
3. El **Organizador** elige un rol para el contacto.
4. El **Organizador** guarda la relación.
5. El sistema establece la relación en la base de datos.
6. El sistema vuelve al caso de uso 05. VerDetallesOrador.

Caso de uso 09. BorrarSesion

*Resumen de la funcionalidad:* elimina a un orador de una sesión.

*Parámetros de entrada:* el orador seleccionado y la sesión.

*Parámetros de salida:* la sesión sin el orador.

*Actores:* **Organizador**.

*Precondición:* el orador seleccionado se encuentra en la sesión seleccionada.

*Postcondición:* la sesión no incluye al orador seleccionado en su lista de oradores.

*Proceso normal principal:*

1. El sistema muestra un mensaje de alerta pidiendo una confirmación.
2. El **Organizador** confirma la acción.
3. El sistema desvincula al orador de la sesión en la base de datos.
4. El sistema vuelve al caso de uso 05. VerDetallesOrador.

Caso de uso 10. AñadirSesion

*Resumen de la funcionalidad:* se añade al orador seleccionado a la lista de oradores de una sesión o se crea una nueva sesión con el orador.

*Parámetros de entrada:* el orador seleccionado

*Parámetros de salida:* la sesión con el orador seleccionado.

*Actores:* **Organizador**

*Precondición:* ninguna.

*Postcondición:* la sesión incluye al orador seleccionado en su lista de oradores.

*Proceso normal principal:*

1. El sistema muestra una ventana con los campos necesarios para crear una sesión, descritos en la *Lista 10.1*.
2. El **Organizador** rellena los campos del formulario.
3. El sistema valida los campos del formulario.
4. El sistema crea la sesión con el orador en ella.
5. El sistema vuelve al caso de uso 05. VerDetallesOrador.

*Alternativas de proceso y excepciones:*

- 2a. El **Organizador** elige la opción de añadir a sesión ya existente.
  - 3a1. El sistema muestra un campo estilo desplegable de todas las sesiones del evento en las que el orador no está.
  - 3a2. El **Organizador** selecciona una opción.
  - 3a3. El **Organizador** confirma la sesión seleccionada.
  - 3a4. El sistema crea la relación entre el orador y la sesión.
  - 3a5. El sistema vuelve al caso de uso 05. VerDetallesOrador.
- 3a. El sistema detecta algún campo erróneo.
  - 3a1. El sistema muestra un error en cada campo detectado como incorrecto.

*Lista 10.1*

- Nombre
- Nombre Corto
- Ubicación
- Categoría
- Inicio
- Fin
- Descripción
- Tipo
- Ajustes Agenda
- Oradores

*Caso de uso 11. Modificar Documentos*

*Resumen de la funcionalidad:* añade o elimina documentos a la carpeta personal del orador o a una de sus sesiones.

*Parámetros de entrada:* sesión seleccionada y orador seleccionado.

*Parámetros de salida:* sesión con el documento incluido

*Actores:* **Organizador**

*Precondición:* ninguna.

*Postcondición:* el orador y sus sesiones con sus documentos actualizados.

*Proceso normal principal:*

1. El sistema muestra todos los documentos de la sesión.
2. El **Organizador** selecciona la opción de añadir documentos.
3. El sistema muestra todos los documentos del orador.
4. El **Organizador** selecciona uno o varios documentos a añadir documento.
5. El **Organizador** confirma la selección.
6. El sistema guarda la relación de la sesión con los documentos en la base de datos.
7. El sistema vuelve al caso de uso 05. VerDetallesOrador.

*Alternativas de proceso y excepciones:*

- 2a. El **Organizador** elimina un documento de la sesión mediante un botón en cada documento de sesión.
  - 2a1. El sistema muestra un mensaje de confirmación.
  - 2a2. El **Organizador** confirma la acción.
  - 2a3. El sistema elimina la relación entre la sesión y el documento.
  - 2a4. El sistema vuelve al caso de uso 05. VerDetallesOrador.
- 4a. El **Organizador** selecciona subir nuevos documentos.
  - 3a1. El sistema muestra un área para arrastrar o subir nuevos documentos.
  - 3a2. El **Organizador** selecciona los documentos a subir de su ordenador.
  - 3a3. El **Organizador** confirma los documentos subidos.
  - 3a4. El sistema vuelve a ejecutar el caso de uso 11. ModificarDocumentos.

*Caso de uso 12. VerArtículos*

*Resumen de la funcionalidad:* presenta los artículos de un evento u orador, deja seleccionarlos para ejecutar las acciones de borrar, modificar y ver detalles. Además, permite crear nuevos y aplicar filtros.

*Parámetros de entrada:* filtros de búsqueda, orador o evento.

*Parámetros de salida:* lista de artículos.

*Actores:* **Organizador**

*Precondición:* el módulo oradores está activo en el sistema.

*Postcondición:* se muestra una lista de artículos sobre los que realizar operaciones de gestión.

*Proceso normal principal:*

1. El sistema lee todos los artículos del evento u orador seleccionado.
2. El sistema presenta el resultado en una tabla.

*Alternativas de proceso y excepciones:*

- 1a. El **Organizador** ha aplicado algún filtro de búsqueda.
  - 1a1. El sistema lee todos los artículos del evento u orador seleccionado aplicando los

filtros de búsqueda proporcionados por el **Organizador**.

1a2. El sistema presenta el resultado en una tabla.

2a. El **Organizador** selecciona uno o más artículos y ejecuta la acción de borrar.

2a1. El sistema ejecuta el caso de uso 17. BorrarArtículo con el/los artículos seleccionados.

2b. El **Organizador** selecciona algún artículo y ejecuta la acción de modificar.

2b1. El sistema ejecuta el caso de uso 18. ModificarPetición con el artículo seleccionado.

2c. El **Organizador** ejecuta la acción de crear artículo.

2c1. El sistema ejecuta el caso de uso 13. CrearArtículo con el artículo seleccionado.

2d. El **Organizador** selecciona algún artículo y ejecuta la acción de ver detalles.

2d1. El sistema ejecuta el caso de uso 14. VerDetallesArtículo con el artículo seleccionado.

2e. El **Organizador** selecciona varios artículos y realiza la acción de enviar correo al **Orador**.

2e1. El sistema ejecuta el caso de uso 19. EnviarCorreo.

#### Caso de uso 13. CrearArtículo

*Resumen de la funcionalidad:* añade un artículo a la base de datos con los datos introducidos por el **Organizador**.

*Parámetros de entrada:* ninguno.

*Parámetros de salida:* el artículo creado.

*Actores:* **Organizador**.

*Precondición:* ninguna.

*Postcondición:* un artículo ha sido añadido a la base de datos.

*Proceso normal principal:*

1. El sistema muestra una ventana pidiendo los datos de la *Lista 13.1*.
2. El **Organizador** rellena los datos de la petición y asigna un orador destinatario.
3. El sistema guarda la petición en la base de datos.
4. El sistema vuelve al caso de uso 12. VerArtículos.

#### *Lista 13.1*

- Orador
- Longitud Artículo
- Sección
- Subsección
- Tema
- Fecha Límite
- Longitud mínima del título
- Longitud máxima del título
- Longitud mínima de la frase clave

- Longitud máxima de la frase clave

#### Caso de uso 14. VerDetallesArtículo

*Resumen de la funcionalidad:* presenta un panel de detalles del artículo que permite visualizar y modificar todos los campos del artículo, tanto de la petición como de la respuesta. El panel permite iniciar todas las acciones de validación del artículo que origina el **Orador**.

*Parámetros de entrada:* el artículo seleccionado.

*Parámetros de salida:* panel de control del artículo.

*Actores:* **Organizador**.

*Precondición:* existe el artículo seleccionado en la base de datos.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recoge la información del artículo y la respuesta de la base de datos.
2. El sistema presenta la información del artículo y la respuesta.

*Alternativas de proceso y excepciones:*

- 2a. El **Organizador** valida la respuesta del **Orador**.
  - 2a1. El sistema ejecuta el caso de uso 15. ValidarRespuesta con el artículo seleccionado.
- 2b. El **Organizador** valida la fecha proporcionada por el **Orador**.
  - 2b1. El sistema ejecuta el caso de uso 16. ValidarPeticiónFecha con el artículo seleccionado.
- 2c. El **Organizador** selecciona la opción de enviar correo al **Orador**.
  - 2c1. El sistema ejecuta el caso de uso 19. EnviarCorreo.

#### Caso de uso 15. ValidarRespuesta

*Resumen de la funcionalidad:* permite visualizar una lista de estados del artículo de los que el orador escoge uno. Una vez validado, envía un correo al orador informándole del nuevo estado.

*Parámetros de entrada:* el artículo seleccionado.

*Parámetros de salida:* el artículo con el estado validado.

*Actores:* **Organizador**.

*Precondición:* El **Orador** ha respondido el artículo.

*Postcondición:* Se ha actualizado el estado del artículo.

*Proceso normal principal:*

1. El **Organizador** revisa el contenido y valida el artículo.
2. El **Organizador** hace la acción de validar escogiendo un estado de la *Lista 15.1*.
3. El sistema marca la versión como validada en la base de datos.
4. El sistema cambia el estado del artículo en la base de datos.
5. El sistema inicia el caso de uso 19. EnviarCorreo.

#### *Lista 15.1*

- Nuevo
- Nueva Versión
- Validado

- Publicado
- Aceptado por el participante
- Rechazado

#### Caso de uso 16. ValidarPeticiónFecha

*Resumen de la funcionalidad:* muestra la fecha límite actual del artículo y la fecha pedida por el orador. Se permite aceptar la petición o rechazar con o sin establecimiento de una nueva fecha. Una vez validado, se envía una notificación al orador por correo.

*Parámetros de entrada:* el artículo seleccionado.

*Parámetros de salida:* el artículo con la fecha revisada.

*Actores:* **Organizador.**

*Precondición:* El **Orador** ha realizado una petición de modificación de fecha de entrega.

*Postcondición:* El **Orador** ha recibido el correo de confirmación y se ha revisado la fecha del artículo.

*Proceso normal principal:*

1. El sistema muestra la notificación de validación pendiente.
2. El **Organizador** revisa el contenido y acepta la petición del **Orador**.
3. El sistema cambia la fecha de entrega del artículo en la base de datos.
4. El sistema borra la petición del **Orador** de la base de datos.
5. El sistema inicia el caso de uso 19. EnviarCorreo.

*Alternativas de proceso y excepciones:*

- 1a. El **Organizador** rechaza la petición del **Orador**.
  - 1a1. El sistema muestra un modal que pida la confirmación explícita del **Organizador**.
  - 1a2. El **Organizador** confirma la acción.
  - 1a3. El sistema borra la petición **Organizador**.
  - 1a4. El sistema inicia el caso de uso 19. EnviarCorreo.
- 1a2i. El **Organizador** introduce una nueva fecha.
  - 1a2i1. El sistema cambia la fecha de entrega del artículo en la base de datos.
  - 1a2i2. El sistema borra la petición **Organizador**.
  - 1a2i3. El sistema inicia el caso de uso 19. EnviarCorreo.

#### Caso de uso 17. BorrarArtículo

*Resumen de la funcionalidad:* elimina un artículo existente en la base de datos.

*Parámetros de entrada:* el artículo seleccionado.

*Parámetros de salida:* ninguno.

*Actores:* **Organizador.**

*Precondición:* existe el artículo en la base de datos.

*Postcondición:* el artículo se ha eliminado de la base de datos.

*Proceso normal principal:*

1. El sistema muestra una ventana de confirmación de la acción.
2. El **Organizador** confirma la acción.
3. El sistema elimina el artículo de la base de datos.

4. El sistema vuelve al caso de uso 12. VerArtículos.

Caso de uso 18. ModificarPetición

*Resumen de la funcionalidad:* permite cambiar los datos de gestión del artículo: tema, fecha de entrega, limitaciones de palabras, estado...

*Parámetros de entrada:* el artículo seleccionado.

*Parámetros de salida:* el artículo modificado en base de datos.

*Actores:* **Organizador**.

*Precondición:* ninguna.

*Postcondición:* el artículo se ha modificado en la base de datos.

*Proceso normal principal:*

1. El sistema muestra un formulario con los campos de administración del artículo.
2. El **Organizador** rellena los campos que necesite de la *Lista 13.1*.
3. El **Organizador** hace la acción de guardar.
4. El sistema modifica el artículo en la base de datos.
5. El sistema refresca la información en pantalla.
6. El sistema muestra una notificación.

Caso de uso 19. EnviarCorreo

*Resumen de la funcionalidad:* Envía un correo al orador. Puede ser ejecutado por el **Organizador**, quien puede añadirle una plantilla de correo o crear el correo desde cero, o de forma automática como resultado de alguna acción realizada por el **Organizador** o el **Orador**.

*Parámetros de entrada:* tipo de correo (automático o manual) y oradores destinatarios.

*Parámetros de salida:* correo enviado.

*Actores:* **Organizador** y **Orador**

*Precondición:* se ha iniciado manualmente la acción de envío de correo o se ha realizado un caso de uso que lo incluye.

*Postcondición:* se ha enviado el correo al orador.

*Proceso normal principal:*

1. El sistema recibe una petición de enviar correo manual.
2. El sistema muestra un formulario con los campos de la *Lista 19.1*.
3. El **Organizador** rellena los campos del correo con la opción de usar plantilla.
4. El **Organizador** selecciona una plantilla de las disponibles para el evento.
5. El **Organizador** confirma la acción.
6. El sistema sustituye las etiquetas de la plantilla por la información del orador destinatario.
7. El sistema envía el correo a los oradores recibidos por parámetro.

*Alternativas de proceso y excepciones:*

- 1a. El sistema recibe una petición de correo automático.
  - 1a1. El sistema comprueba el tipo de correo a enviar (confirmación de artículo, de cambio de fecha de entrega...) y selecciona la vista de correo adecuada.

1a2. El sistema envía el correo.

3a. El **Organizador** rellena los campos del correo con la opción de no usar plantilla.

3a1. El **Organizador** rellena el campo de cuerpo del correo.

3a2. El **Organizador** confirma la acción.

3a3. El sistema envía el correo a los oradores recibidos por parámetro.

#### *Lista 19.1*

- Asunto.
- BCC.
- CC.
- Usar plantilla.
- Cuerpo del correo.
- Plantilla.

#### *Caso de uso 20. Login*

*Resumen de la funcionalidad:* permite introducir un correo. Si el correo introducido pertenece a un orador o a un asistente validado, envía un correo con un código de un solo uso al participante. Si se introduce el código adecuado, el usuario queda autenticado.

*Parámetros de entrada:* ninguno.

*Parámetros de salida:* el usuario autenticado.

*Actores:* **Participante**

*Precondición:* ninguna.

*Postcondición:* el participante queda autenticado y tiene acceso a todas las funcionalidades del portal.

*Proceso normal principal:*

1. El sistema muestra un formulario con un campo de correo electrónico.
2. El **Participante** introduce su correo electrónico.
3. El sistema valida que el correo sea de un orador o de un asistente validado de algún evento.
4. El sistema envía un correo al **Participante** y muestra un nuevo campo para el código de un solo uso.
5. El **Participante** introduce el código recibido.
6. El sistema valida el código introducido.
7. El sistema autentifica al **Participante**.

*Alternativas de proceso y excepciones:*

4a. Si el correo no es válido, muestra un mensaje de error indicando el problema.

7a. Si el código introducido no es válido, vuelve al paso 1.

### Caso de uso 21. SeleccionarOrganizador

*Resumen de la funcionalidad:* muestra las organizaciones que tienen al participante en su lista de registros y permite escoger una para conectarse a su entorno.

*Parámetros de entrada:* **Participante** autenticado.

*Parámetros de salida:* conexión al entorno del organizador.

*Actores:* **Participante**.

*Precondición:* el participante está autenticado.

*Postcondición:* el sistema se ha conectado al entorno del organizador.

*Proceso normal principal:*

1. El sistema muestra todos los organizadores que tienen al participante registrado en sus eventos.
2. El **Participante** selecciona uno de los organizadores.
3. El sistema se conecta al entorno de ese organizador.
4. El sistema ejecuta el caso de uso 22. SeleccionarEvento.

*Alternativas de proceso y excepciones:*

- 1a. Si el **Participante** solo ha sido registrado por un organizador, el sistema lo selecciona automáticamente.

### Caso de uso 22. SeleccionarEvento

*Resumen de la funcionalidad:* muestra los eventos de un organizador a los que está registrado un orador y permite escoger uno.

*Parámetros de entrada:* **Participante** autenticado y organizador seleccionado.

*Parámetros de salida:* evento seleccionado.

*Actores:* **Participante**

*Precondición:* el **Participante** está autenticado y ha escogido un Organizador

*Postcondición:* el sistema ha guardado el evento seleccionado.

1. El sistema muestra todos los eventos a los que está registrado el participante.
2. El **Participante** selecciona uno de los eventos.
3. El sistema guarda el evento seleccionado.
4. El sistema ejecuta el caso de uso 25. VerParticipación.

*Alternativas de proceso y excepciones:*

- 1a. Si el **Participante** solo está registrado a un evento, el sistema lo selecciona automáticamente y no muestra ninguna lista de eventos.

### Caso de uso 23. VerDocumentos

*Resumen de la funcionalidad:* muestra un panel con todos los documentos subidos por el orador.

*Parámetros de entrada:* evento seleccionado y **Participante** autenticado.

*Parámetros de salida:* ninguno.

*Actores:* **Participante**

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recoge de la base de datos todos los documentos relacionados con el registro del **Participante**.
2. El sistema muestra todos los documentos.

*Alternativas de proceso y excepciones:*

- 3a. El **Participante** selecciona la opción de modificar documentos  
3a1. El sistema ejecuta el caso de uso 24. ModificarDocumentos.

#### Caso de uso 24. ModificarDocumentos

*Resumen de la funcionalidad:* permite al orador subir o eliminar documentos de su registro.

*Parámetros de entrada:* el **Participante** registrado y el evento.

*Parámetros de salida:* ninguno.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* se han modificado los documentos del participante en la base de datos.

*Proceso normal principal:*

1. El sistema muestra un menú desde el que eliminar documentos existentes o añadir nuevos.
2. El **Participante** realiza la acción de subir o borrar un documento.
3. El **Participante** acepta los cambios.
4. El sistema modifica la base de datos en consecuencia a la acción del **Participante**.
5. El sistema ejecuta el caso de uso 23. VerDocumentos.

#### Caso de uso 25. VerParticipación

*Resumen de la funcionalidad:* muestra los datos de participación de un orador al evento: estado, actividades en las que participa, vuelos, información del hotel, sesiones a las que atiende y requisitos especiales.

*Parámetros de entrada:* evento seleccionado y orador autenticado.

*Parámetros de salida:* ninguno.

*Actores:* **Participante**

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recoge de la base de datos los datos para el orador recibido en el evento recibido.
2. El sistema muestra los datos recogidos de la base de datos.

*Alternativas de proceso y excepciones:*

- 3a. El **Participante** selecciona la acción de modificar participación.  
3a1. El sistema ejecuta el caso de uso 27. ModificarParticipación.
- 3b. El **Participante** selecciona la acción de eliminar alguna de sus actividades.  
3b1. El sistema ejecuta el caso de uso 26. RechazarActividad

#### Caso de uso 26. RechazarActividad

*Resumen de la funcionalidad:* elimina la participación de un **Orador** a una actividad y notifica al **Organizador**.

*Parámetros de entrada:* actividad seleccionada y **Participante** autenticado.

*Parámetros de salida:* ninguno.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* el registro del participante en la base de datos ha borrado su relación con la actividad.

*Proceso normal principal:*

1. El sistema muestra un modal de confirmación de la acción.
2. El **Participante** confirma la acción.
3. El sistema elimina la relación del **Orador** de la base de datos.
4. El sistema vuelve a ejecutar el caso de uso 25. VerParticipación.

#### Caso de uso 27. ModificarParticipación

*Resumen de la funcionalidad:* permite modificar algunos datos de la participación como los vuelos y los requisitos especiales.

*Parámetros de entrada:* el evento seleccionado y el **Participante** autenticado.

*Parámetros de salida:* la participación modificada.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* se ha modificado la participación del **Orador** en la base de datos.

*Proceso normal principal:*

1. El sistema muestra un formulario con los datos modificables, descritos en la *Lista 27.1*
2. El **Participante** cambia la información que requiera.
3. El **Participante** guarda los cambios.
4. El sistema guarda los datos en la base de datos
5. El sistema vuelve a ejecutar el caso de uso 25. VerParticipación.

#### *Lista 27.1*

- Estado mesa de gala.
- Estado actividad social.
- Vuelo de ida.
- Vuelo de vuelta.
- Requisitos especiales.

#### Caso de uso 28. VerSesiones

*Resumen de la funcionalidad:* muestra todas las sesiones de un orador: sus documentos adjuntos y la información de cada orador de la sesión.

*Parámetros de entrada:* el evento seleccionado y el **Participante** autenticado.

*Parámetros de salida:* ninguno.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento.

*Postcondición:* ninguna.

*Proceso normal principal:*

1. El sistema recoge todas las sesiones y documentos del **Orador** en el evento seleccionado.
2. El sistema monta un panel que lista las sesiones, los oradores y los documentos.

*Alternativas de proceso y excepciones:*

- 3a. El **Participante** selecciona la acción de añadir documentos a una de las sesiones.
  - 3a1. El sistema ejecuta el caso de uso 29. ModificarDocumentosSesión.

Caso de uso 29. ModificarDocumentosSesión

*Resumen de la funcionalidad:* permite añadir nuevos documentos a una sesión a partir de los documentos subidos previamente por el **Participante** a su carpeta.

*Parámetros de entrada:* sesión seleccionada y **Participante**.

*Parámetros de salida:* sesión los documentos modificados.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento y una sesión

*Postcondición:* se han modificado los documentos de la sesión seleccionada.

*Proceso normal principal:*

1. El sistema recoge los documentos subidos por el orador de la base de datos.
2. El sistema recoge los documentos relacionados con la sesión.
3. El sistema muestra los documentos aún no relacionados.
4. El **Participante** selecciona los documentos que necesite.
5. El sistema crea una petición de cambio de documentos de sesión con los nuevos documentos seleccionados.
6. El sistema muestra una notificación para dar retroalimentación al usuario.

Caso de uso 30. VerArtículosOrador

*Resumen de la funcionalidad:* lista todos los artículos del orador autenticado agrupados según sus estados y permite iniciar acciones sobre este.

*Parámetros de entrada:* el evento seleccionado y el **Participante** autenticado.

*Parámetros de salida:* lista de los artículos del **Participante** agrupados por estado.

*Actores:* **Participante**

*Precondición:* el **Participante** ha seleccionado un evento

*Postcondición:* se muestran todos los artículos de un **Participante** para el evento seleccionado.

*Proceso normal principal:*

1. El sistema recoge todos los artículos del **Participante** para el evento seleccionado.
2. El sistema agrupa los artículos por estado.
3. El sistema muestra los artículos.

*Alternativas de proceso y excepciones:*

- 4a. El **Participante** selecciona la opción de responder un artículo
  - 4a1. El sistema ejecuta el caso de uso 31. ResponderArtículo.

- 4b. El **Participante** selecciona la opción de rechazar un artículo 4b1. El sistema ejecuta el caso de uso 32. RechazarArtículo.
- 4c. El **Participante** selecciona la opción de ampliar la fecha de entrega de un artículo 4c1. El sistema ejecuta el caso de uso 33. PedirAmpliaciónFechaArtículo.

#### Caso de uso 31. ResponderArtículo

*Resumen de la funcionalidad:* permite rellenar todos los campos referentes a la respuesta de un artículo: título, contenido, frase clave y fotografía. Los campos validan la longitud del contenido y muestran la retroalimentación necesaria.

*Parámetros de entrada:* el artículo seleccionado y el **Participante** autenticado.

*Parámetros de salida:* el artículo respondido.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado un evento y tiene alguna petición de artículo que no ha sido rechazada.

*Postcondición:* el artículo ha sido respondido y se ve reflejado en el modelo de la base de datos.

*Proceso normal principal:*

1. El sistema muestra un formulario con todos los campos del artículo.
2. El **Participante** rellena los campos del artículo correspondientes a la *Lista 31.1*.
3. El **Participante** selecciona la acción de enviar el artículo.
4. El sistema valida la longitud de las respuestas.
5. El sistema actualiza el artículo en la base de datos, cambiando el estado a "Nueva Versión".
6. El sistema ejecuta el caso de uso 19. EnviarCorreo.

*Alternativas de proceso y excepciones:*

- 3a. El **Participante** selecciona la acción de guardar artículo. 3a1. El sistema guarda los datos del artículo sin cambiar el estado de este.
- 5a. Si los campos no cumplen con la validación de longitud, se muestra un mensaje de error en cada campo que no lo cumpla.

#### *Lista 31.1*

- Título.
- Contenido.
- Frase clave.
- Imagen.
- Requisitos especiales.
- Correos copia.

#### Caso de uso 32. RechazarArtículo

*Resumen de la funcionalidad:* cambia el estado de un artículo a rechazado y no permite al **Participante** realizar modificaciones sobre este.

*Parámetros de entrada:* el artículo seleccionado y el **Orador** autenticado.

*Parámetros de salida:* el artículo con el estado cambiado.

**Actores: Participante.**

*Precondición:* el **Participante** tiene una petición de artículo no rechazada y ha seleccionado un evento.

*Postcondición:* el artículo ha quedado como rechazado, se ha persistido en la base de datos y se ha refrescado la información en pantalla.

*Proceso normal principal:*

1. El sistema muestra un modal de confirmación y un campo de Razón".
2. El **Orador** escribe la razón.
3. El **Orador** confirma la acción.
4. El sistema actualiza el artículo en la base de datos, cambiando el estado a Rechazado".
5. El sistema vuelve al caso de uso 30. VerArtículosOrador refrescando la información que se muestra.

*Alternativas de proceso y excepciones:*

- 3a. El **Orador** selecciona la acción de cancelar. 3a1. El sistema vuelve al caso de uso 30. VerArtículosOrador.

Caso de uso 33. PedirAmpliaciónFechaArtículo

*Resumen de la funcionalidad:* genera una petición de cambio de la fecha de límite de un artículo.

*Parámetros de entrada:* el artículo seleccionado y el **Participante** autenticado.

*Parámetros de salida:* la petición de cambio de fecha.

**Actores: Participante**

*Precondición:* el **Participante** tiene una petición de artículo no rechazada y ha seleccionado un evento.

*Postcondición:* se ha creado la petición de cambio de fecha de entrega.

*Proceso normal principal:*

1. El sistema muestra un modal con un campo para especificar la nueva fecha.
2. El **Participante** introduce una fecha.
3. El **Participante** confirma la acción.
4. El sistema actualiza el artículo en la base de datos, marcando que tiene una petición de cambio de fecha pendiente.
5. El sistema refresca la información que está mostrando.

*Alternativas de proceso y excepciones:*

- 3a. El **Orador** selecciona la acción de cancelar. 3a1. El sistema vuelve al caso de uso 30. VerArtículosOrador.

Caso de uso 34. VerInformaciónPersonal

*Resumen de la funcionalidad:* permite al participante visualizar su información personal, que es independiente de los eventos a los que se encuentra registrado, y iniciar cambios en esta.

*Parámetros de entrada:* el **Participante** autenticado.

*Parámetros de salida:* ninguno.

**Actores: Participante.**

*Precondición:* el **Participante** se ha autenticado en la base de datos y ha seleccionado el panel

de datos personales.

*Postcondición:* se muestra la información personal del **Participante**.

*Proceso normal principal:*

1. El sistema recoge todos los datos del modelo de contacto del **Participante**, listados en la *Lista 34. 1*.
2. El sistema muestra la información distribuida por secciones.
3. El **Participante** selecciona algún campo para editar.
4. El sistema ejecuta el caso de uso 35. ModificarInformaciónPersonal.

*Alternativas de proceso y excepciones:*

- 3a. El **Participante** selecciona la opción de autorizar/desautorizar a uno de sus **Asistentes**.
  - 3a1. El sistema inicia el caso de uso 36. ModificarAutorizaciónAsistente.

*Lista 34.1*

- Título
- Nombre
- Apellido
- Imagen
- Biografía
- Email
- Email2
- Teléfono
- Móvil
- Organización
- Función
- Departamento
- Contactos vinculados

*Caso de uso 35. ModificarInformaciónPersonal*

*Resumen de la funcionalidad:* permite realizar una petición a los organizadores del evento que pide el cambio de alguno de los campos de información personal del **Participante**.

*Parámetros de entrada:* el campo seleccionado y el **Participante** autenticado

*Parámetros de salida:* la petición de cambio de información personal.

*Actores:* **Participante**.

*Precondición:* el **Participante** ha seleccionado algún campo de su información personal.

*Postcondición:* se ha realizado la petición de cambio de información personal a los **Organizadores**.

*Proceso normal principal:*

1. El sistema muestra un modal con un campo que representa aquel que ha escogido el **Participante**.

2. El **Participante** modifica la información como considere.
3. El **Participante** guarda la información.
4. El sistema crea o modifica la petición de cambios en la base de datos en caso de ya existir una.
5. El sistema muestra una notificación para dar retroalimentación.
6. El sistema vuelve al caso de uso 34. VerInformaciónPersonal.

*Alternativas de proceso y excepciones:*

- 1a. El **Participante** ha iniciado el caso de uso mediante el campo de Contactos Vinculados.
  - 1a1. El recoge todos los contactos que no están vinculados al **Participante**.
  - 1a2. El sistema presenta un campo de búsqueda de los contactos recogidos.
  - 1a3. El **Participante** selecciona un Contacto.
  - 1a4. El sistema crea la relación entre el contacto del **Participante** y el seleccionado.
  - 1a5. El sistema vuelve al caso de uso 34. VerInformaciónPersonal.
- 3a. El **Participante** selecciona la acción de cancelar. 3a1. El sistema vuelve al caso de uso 34. VerInformaciónPersonal.

*Caso de uso 36. ModificarAutorizaciónAsistente Resumen de la funcionalidad:* permite a un **Orador** modificar la autorización de uno de sus asistentes para modificar su información personal

*Parámetros de entrada:* el **Orador** autenticado y el **Asistente** seleccionado.

*Parámetros de salida:* ninguno.

*Actores:* **Orador**

*Precondición:* el **Orador** se ha autenticado y ha seleccionado a un asistente con la acción de modificar la autorización.

*Postcondición:* se ha guardado en base de datos el nuevo estado.

*Proceso normal principal:*

1. El sistema muestra un modal de confirmación detallando las consecuencias de la acción.
2. El **Orador** confirma la acción.
3. El sistema modifica la relación entre el **Asistente** y el **Orador** en la base de datos.
4. El sistema muestra una notificación para dar retroalimentación.
5. El sistema vuelve al caso de uso 34. VerInformaciónPersonal.

*Alternativas de proceso y excepciones:*

- 2a. El **Orador** cancela la acción. 2a1. El sistema vuelve al caso de uso 34. VerInformaciónPersonal.

*Caso de uso 37. SeleccionarOrador*

*Resumen de la funcionalidad:* muestra los oradores a los que esta vinculado un asistente y permite escoger uno.

*Parámetros de entrada:* **Asistente** autenticado.

*Parámetros de salida:* evento seleccionado.

*Actores:* **Asistente**

*Precondición:* el **Asistente** está autenticado y ha escogido un Organizador y un evento.

*Postcondición:* se guarda el **Orador** seleccionado.

1. El sistema muestra oradores del Asistente.
2. El **Asistente** selecciona uno de los Oradores.
3. El sistema guarda el evento seleccionado.
4. El sistema ejecuta el caso de uso 25. VerParticipación.

*Alternativas de proceso y excepciones:*

- 1a. Si el **Asistente** solo está registrado a un evento, el sistema lo selecciona automáticamente y no muestra ninguna lista de oradores.

## 3.2. Requisitos no Funcionales

### Requisitos de seguridad

- **Privacidad de los datos:** se deberá proporcionar un sistema que asegure las políticas y legislación sobre privacidad de los datos. Para cumplir con esto, no solo se deberá cumplir con las medidas impuestas por la Ley Integral de Protección de Datos, sino que al trabajar con múltiples organizaciones, cada una deberá disponer de su base de datos propia para evitar el filtrado de datos a causa de consultas mal ejecutadas.

### Requisitos de rendimiento

- **Tiempo de respuesta:** se deberá garantizar una carga rápida de las páginas, especialmente en el panel del participante. Para ello, se deberá cargar siempre solo lo necesario y ejecutar las consultas de la manera más eficiente posible para garantizar un alto grado de usabilidad, implementar caches de datos donde se pueda...
- **Escalabilidad:** el factor de escalabilidad juega un papel muy importante debido a que se deben consultar los registros de múltiples organizaciones, pudiendo generar tiempos de espera largos. Por esto, se deberá construir un sistema de indexación que permita hacer este tipo de consultas de forma centralizada y rápida. Además, se implementará la arquitectura sobre una base de datos distribuida usando los servicios de Amazon Web Services.

### Requisitos de usabilidad

- **Interfaz intuitiva:** se deberá desarrollar una interfaz intuitiva, siguiendo los principios de diseño de Gestalt [5] para asegurar una experiencia homogénea y coherente: contraste adecuado, agrupación de contenidos similares...
- **Interfaz coherente:** al ser un módulo añadido a una aplicación existente, se deberá seguir el diseño general de esta: paleta de colores, tipografía, espaciados...

Mantenibilidad y Gestión del Código Al tratarse de un proyecto integrado a una aplicación en la que trabaja un equipo de múltiples personas, estos requisitos cobran especial importancia:

- **Claridad del Código:** se deberá mantener una base de código limpia y siguiendo siempre buenas prácticas de programación para asegurar la mantenibilidad del código: nombres de variables expresivos, separación de funciones, modularidad, funciones cortas...
- **Tests Automatizados:** se deberán crear tests automatizados que faciliten el despliegue del código a producción a través de CI/CD y que aseguren el funcionamiento del módulo en futuras actualizaciones.
- **Integración con metodologías ágiles:** se deberá seguir una metodología de trabajo ágil, ya que es la que se utiliza en el equipo de desarrollo. Se deberán planificar reuniones regulares con el cliente y sprints de desarrollo para las diferentes funcionalidades.

## 4 Análisis de requisitos

A continuación, se analizarán de forma individual los casos de uso especificados durante la descripción de requisitos.

### 4.1. Diagrama de clases de entidades sin operaciones

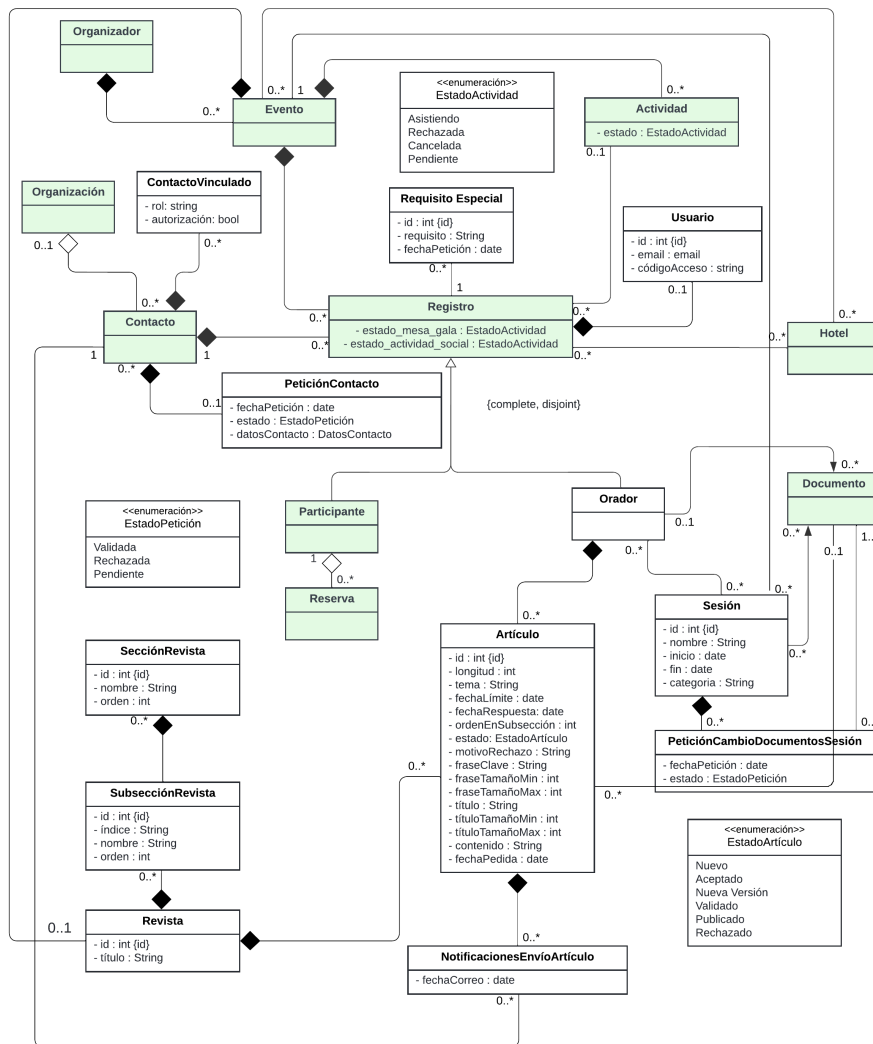


Figura 4.1: Diagrama de casos de entidades sin operaciones del módulo de oradores de Nudato.

## 4.2. Traducción de actividades

### 4.2.1. Caso de uso 01. VerOradores

#### Diagrama de actividades

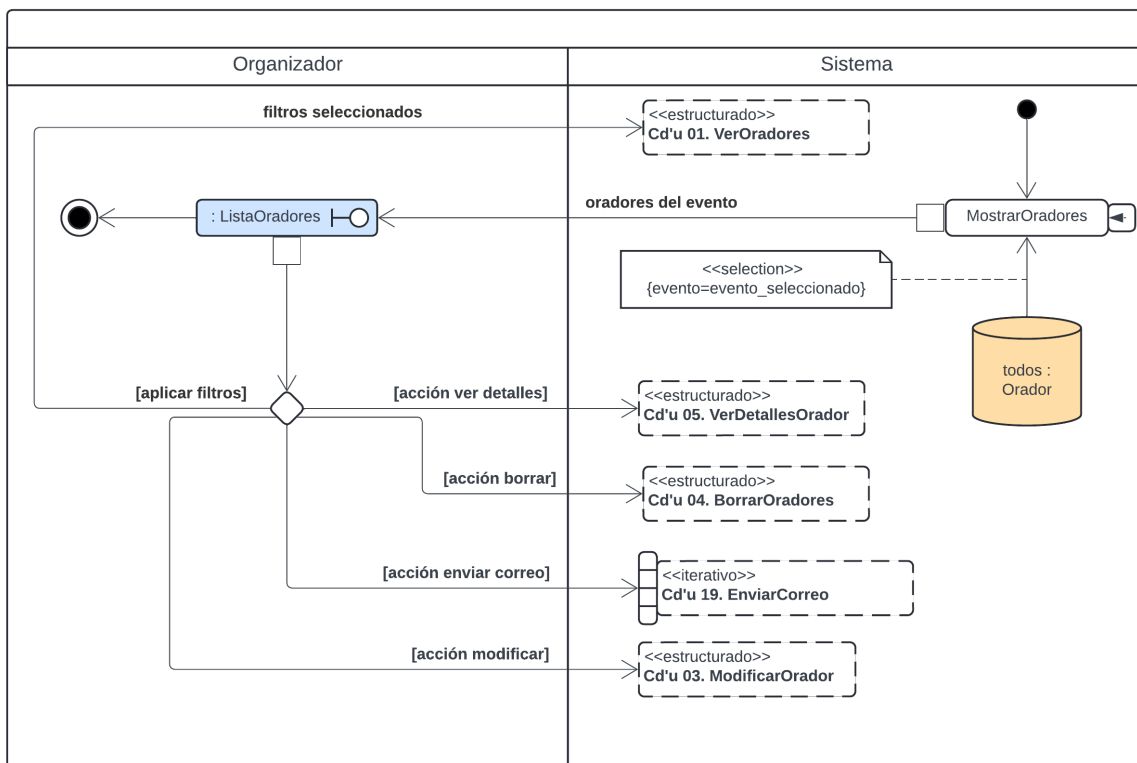


Figura 4.2: Diagrama de actividades del caso de uso 01.

#### Diagrama de clases

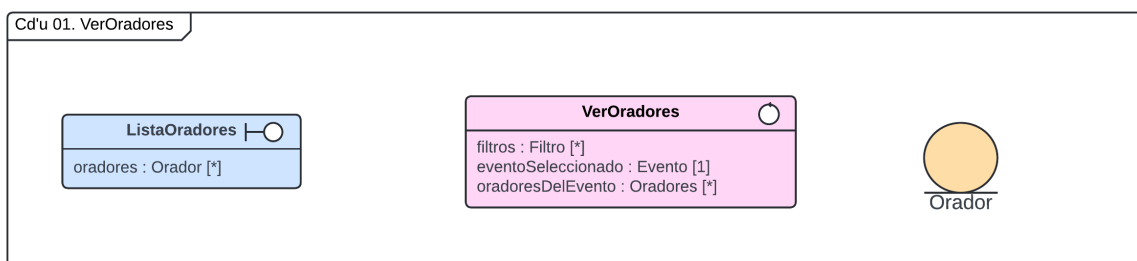


Figura 4.3: Diagrama de clases del caso de uso 01.

# Diagrama de secuencias

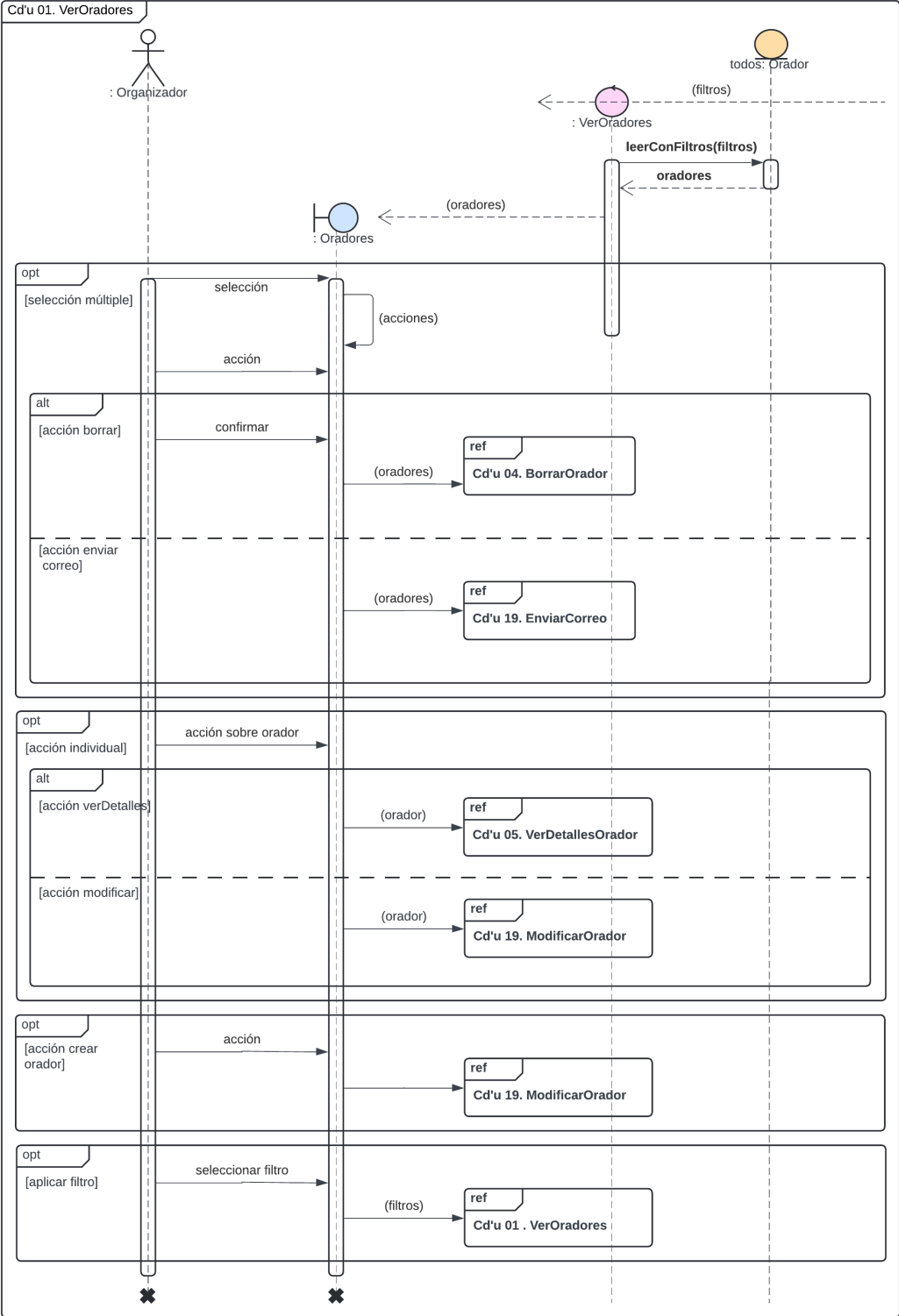


Figura 4.4: Diagrama de secuencias del caso de uso 01.

## 4.2.2. Caso de uso 02. CrearOrador

### Diagrama de actividades

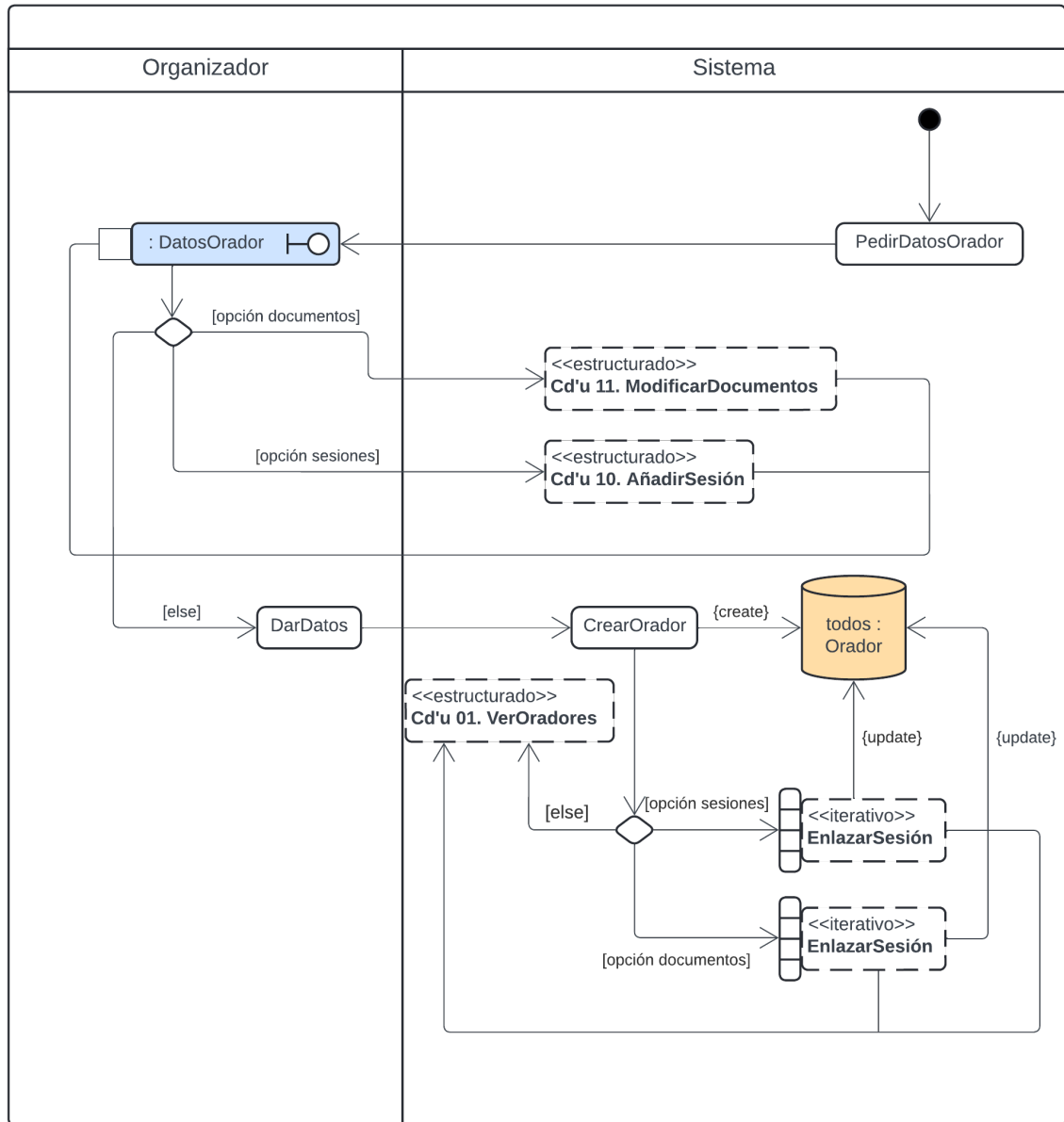


Figura 4.5: Diagrama de actividades del caso de uso 02.

## Diagrama de clases

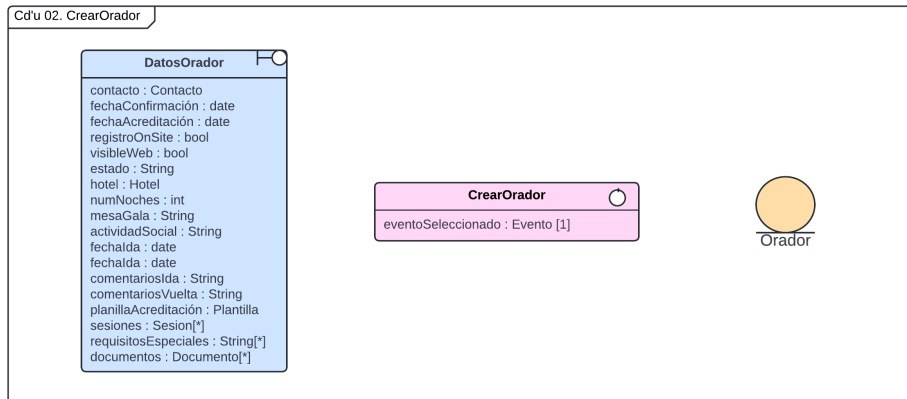


Figura 4.6: Diagrama de clases del caso de uso 02.

## Diagrama de secuencias

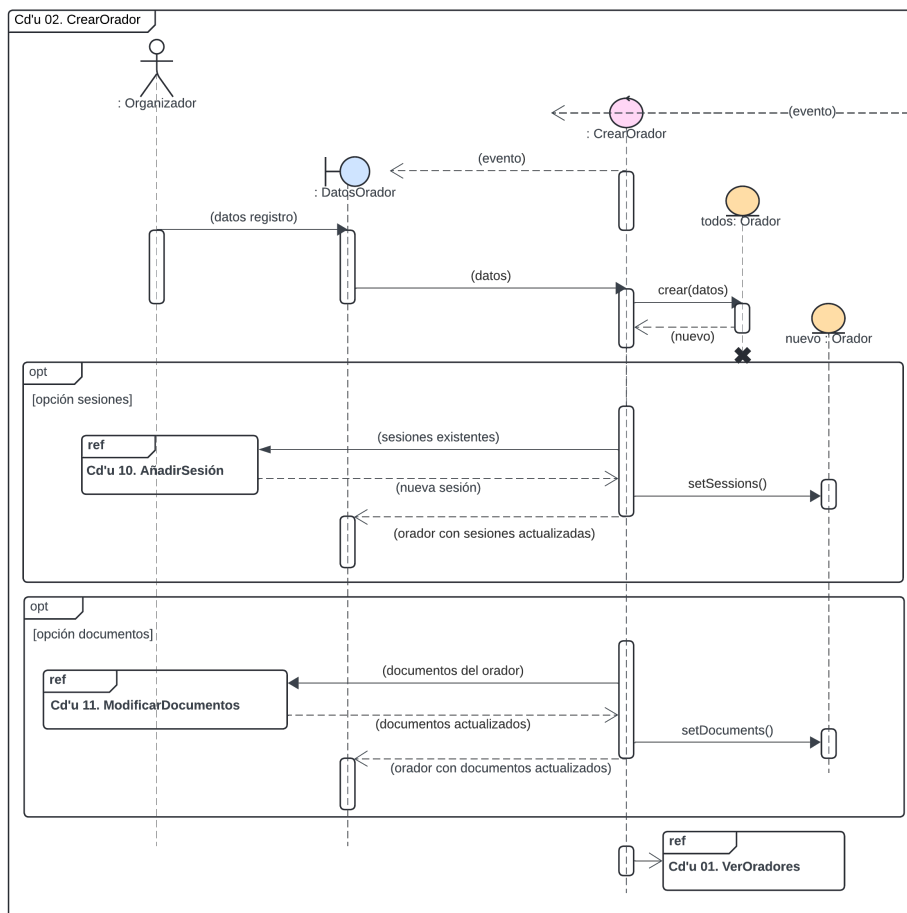


Figura 4.7: Diagrama de secuencias del caso de uso 02.

### 4.2.3. Caso de uso 03. ModificarOrador

#### Diagrama de actividades

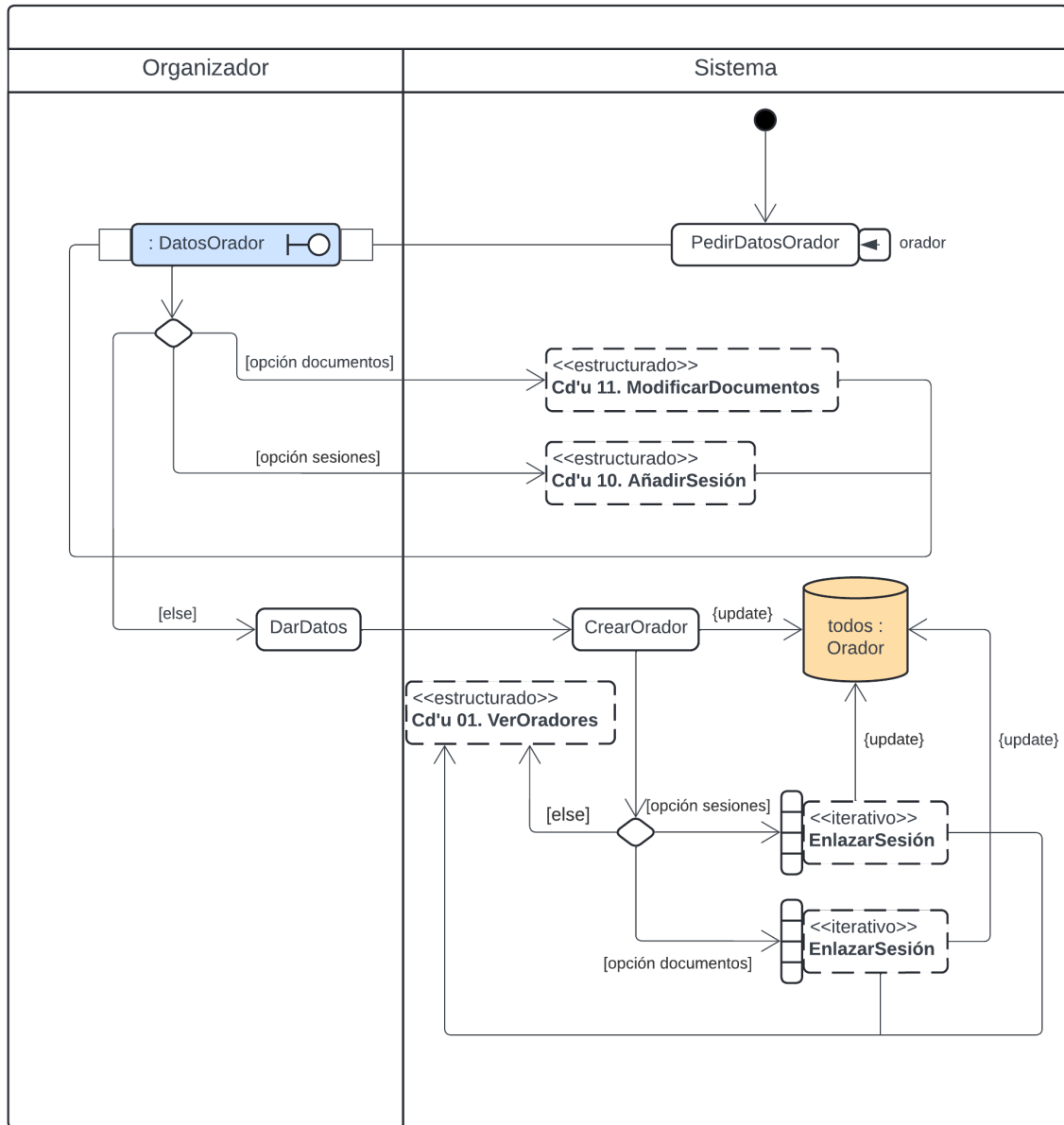


Figura 4.8: Diagrama de actividades del caso de uso 03.

## Diagrama de clases

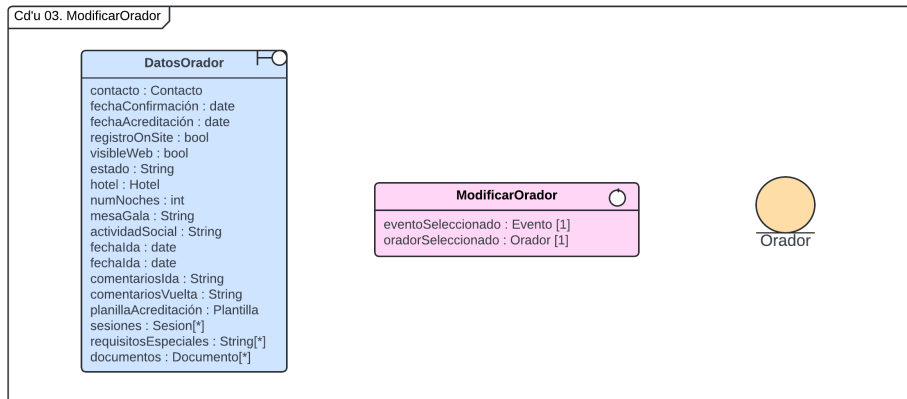


Figura 4.9: Diagrama de clases del caso de uso 03.

## Diagrama de secuencias

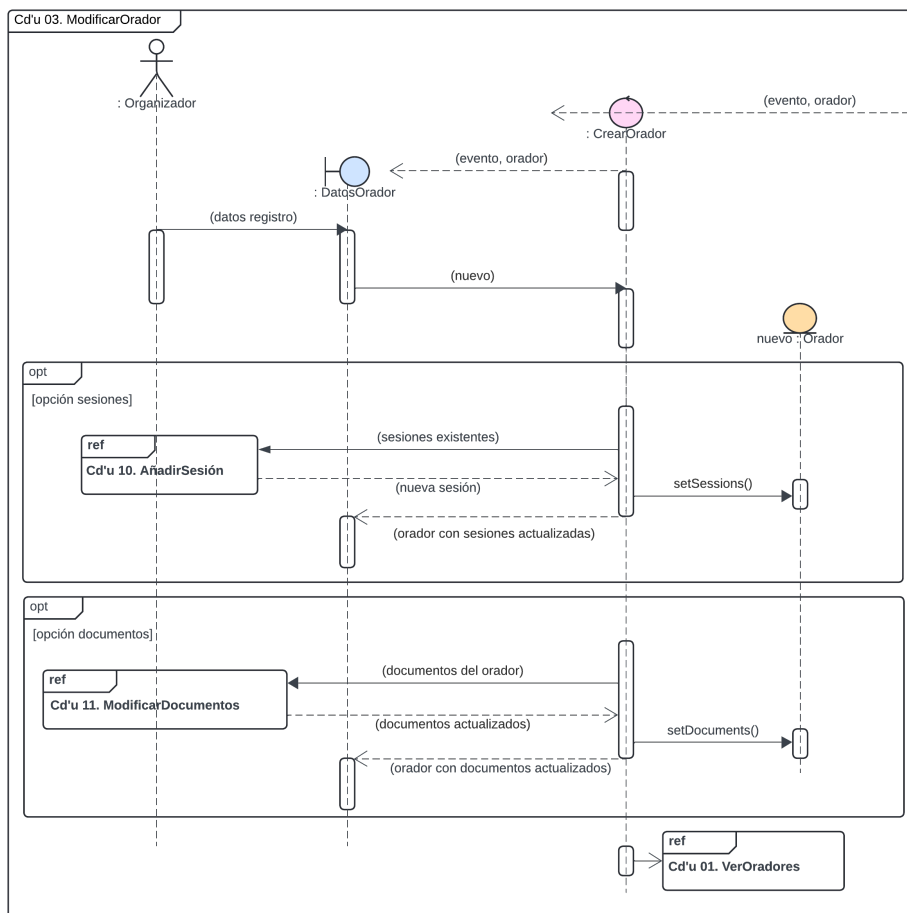


Figura 4.10: Diagrama de secuencias del caso de uso 03.

#### 4.2.4. Caso de uso 04. BorrarOrador

##### Diagrama de actividades

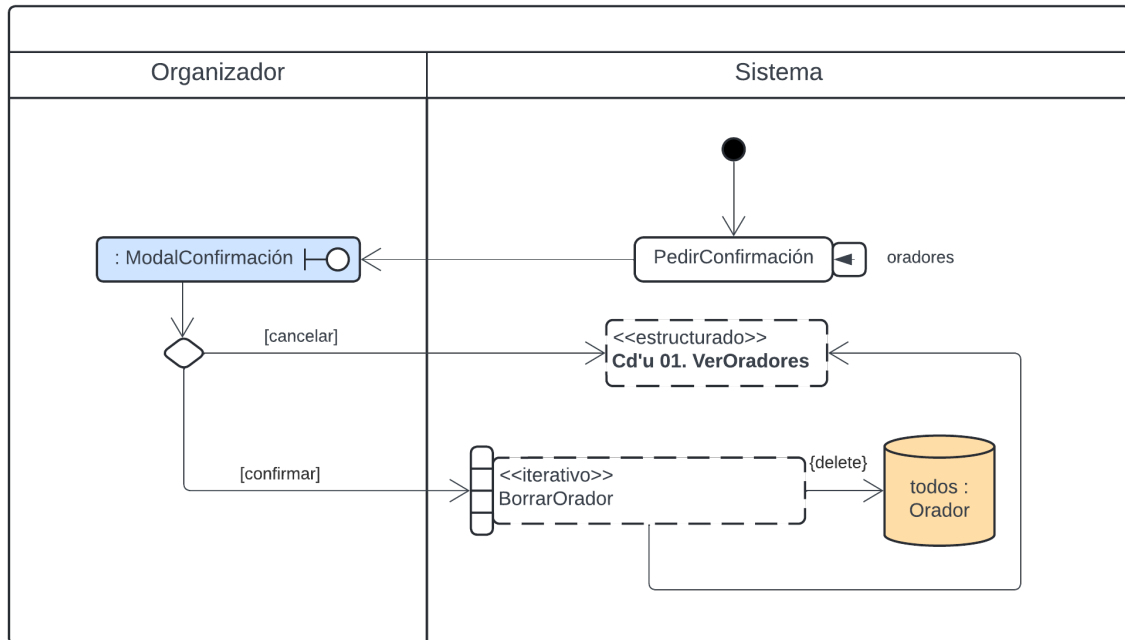


Figura 4.11: Diagrama de actividades del caso de uso 04.

##### Diagrama de clases

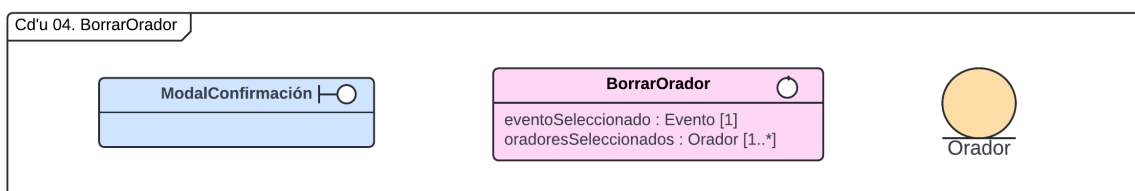


Figura 4.12: Diagrama de clases del caso de uso 04.

Diagrama de secuencias

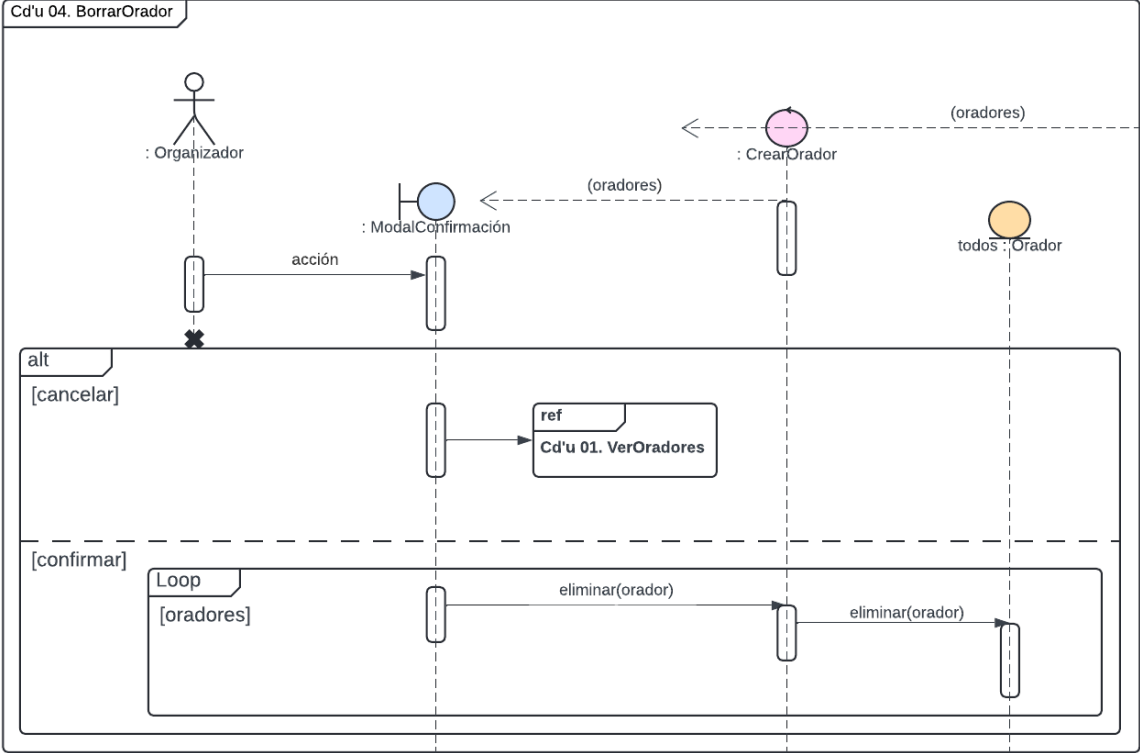


Figura 4.13: Diagrama de secuencias del caso de uso 04.



## Diagrama de clases

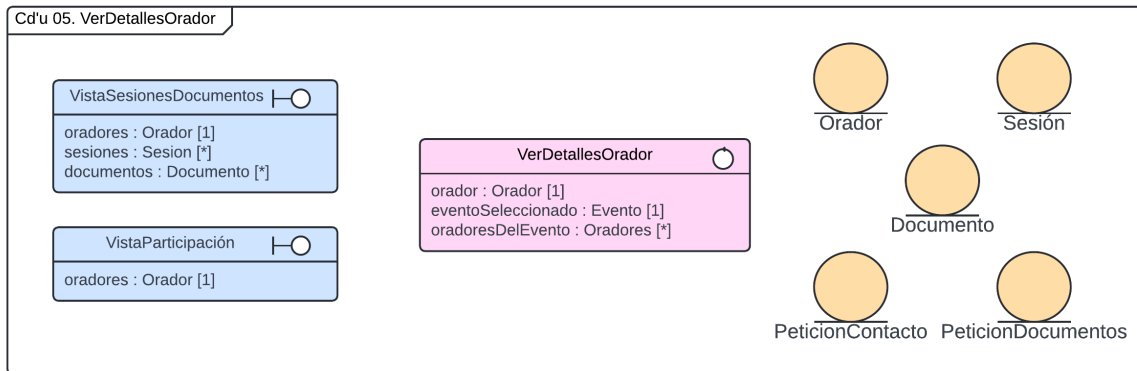


Figura 4.15: Diagrama de clases del caso de uso 05.

## Diagrama de secuencias

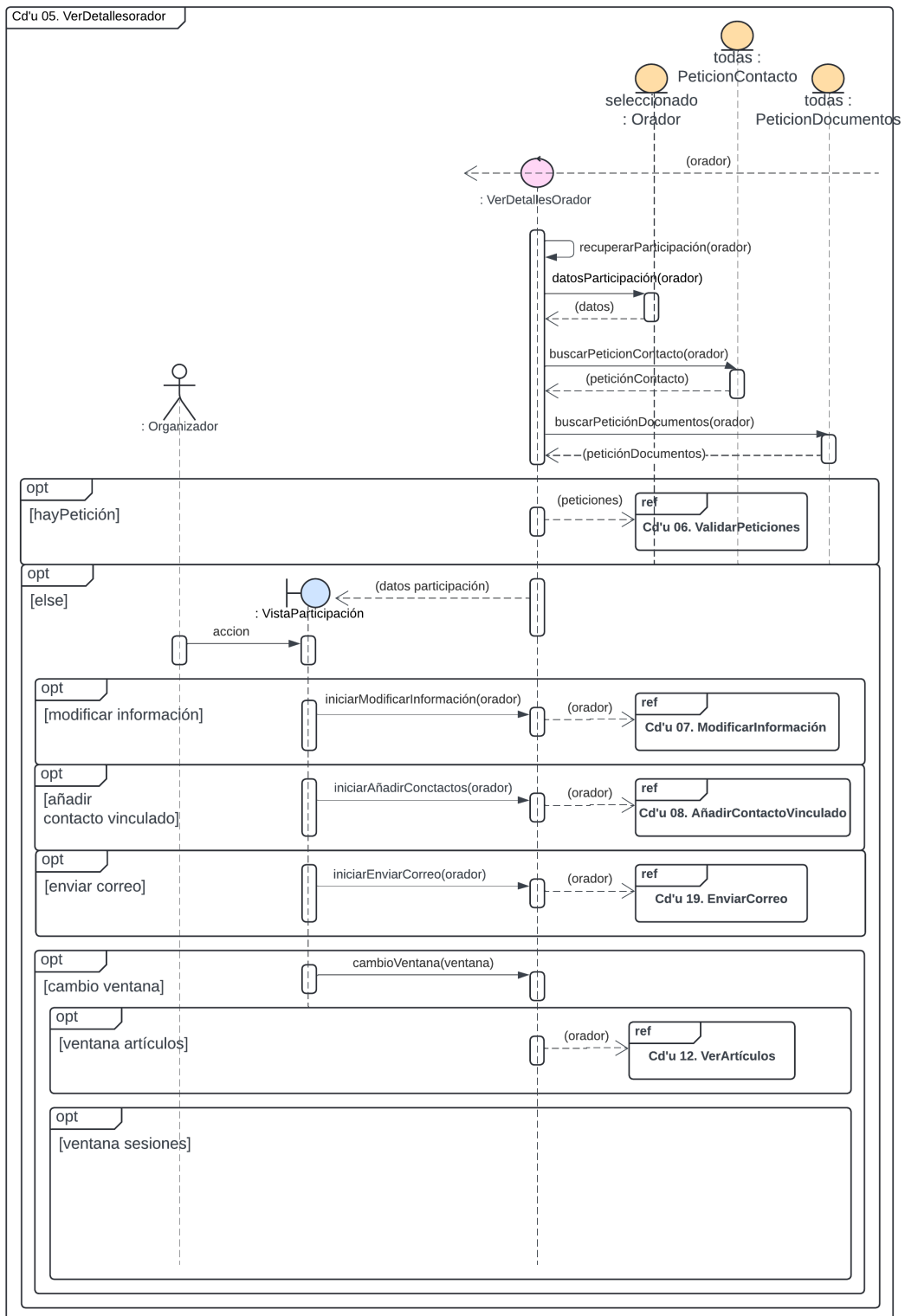


Figura 4.16: Diagrama de secuencias del caso de uso 05.

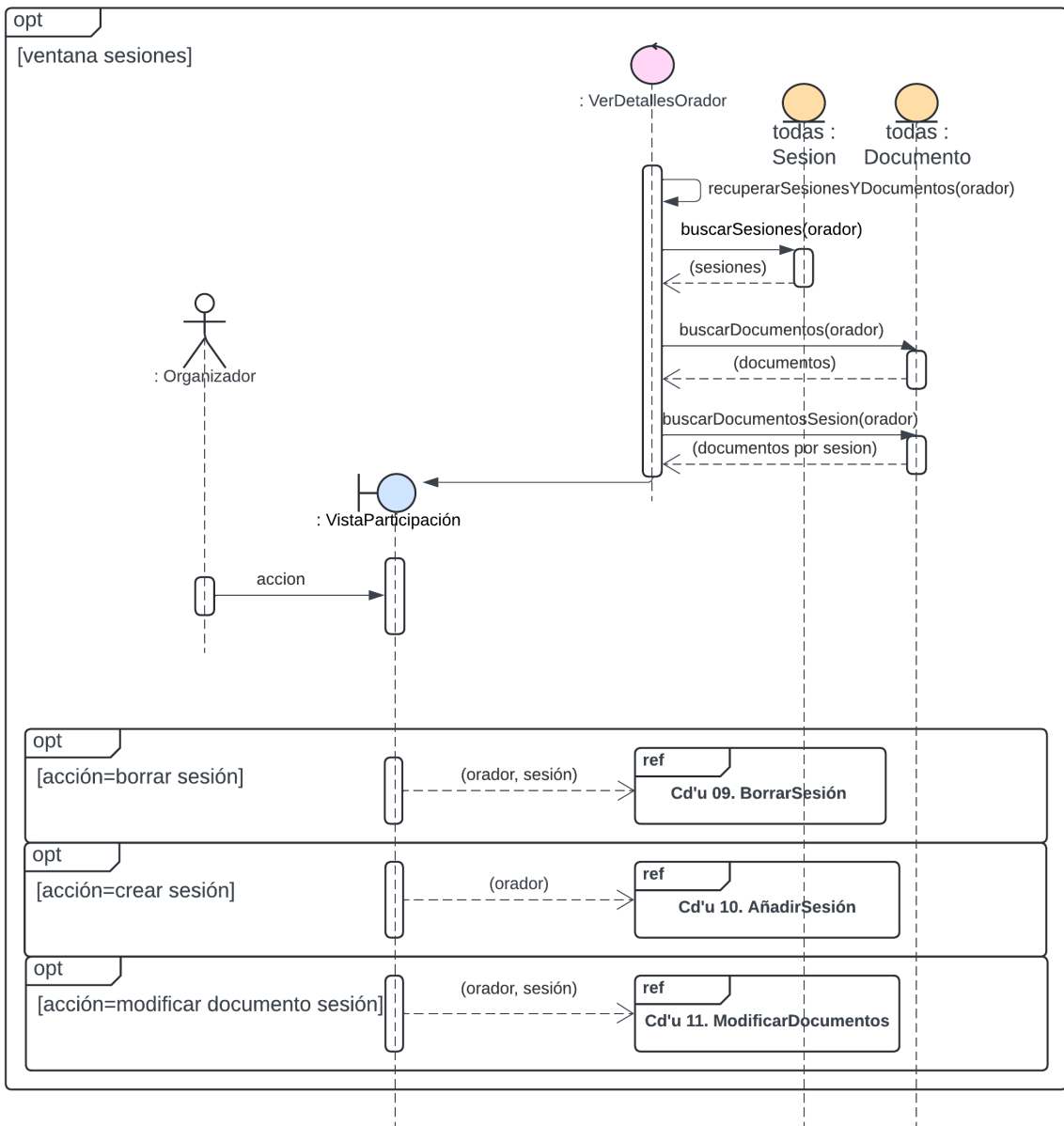


Figura 4.17: Diagrama de secuencias del caso de uso 05.

## 4.2.6. Caso de uso 06. ValidarPeticones

### Diagrama de actividades

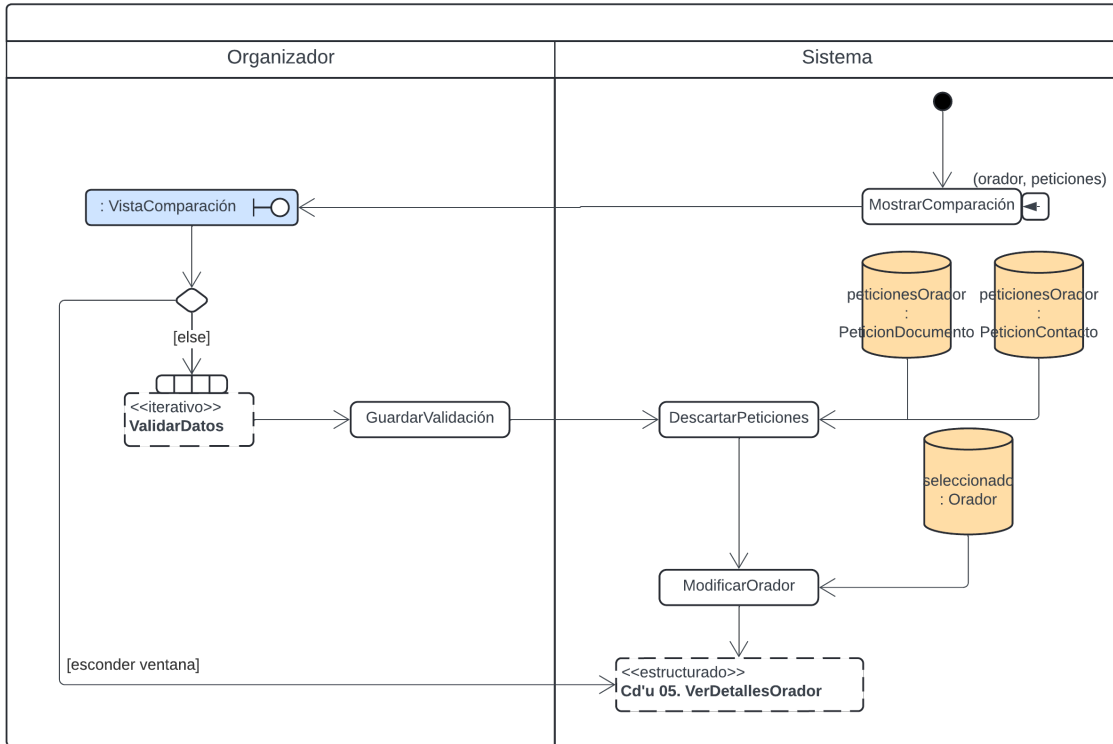


Figura 4.18: Diagrama de actividades del caso de uso 06.

### Diagrama de clases

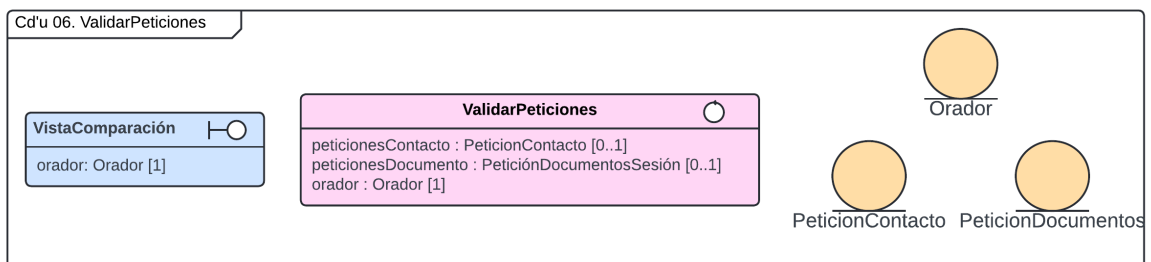


Figura 4.19: Diagrama de clases del caso de uso 06.

# Diagrama de secuencias

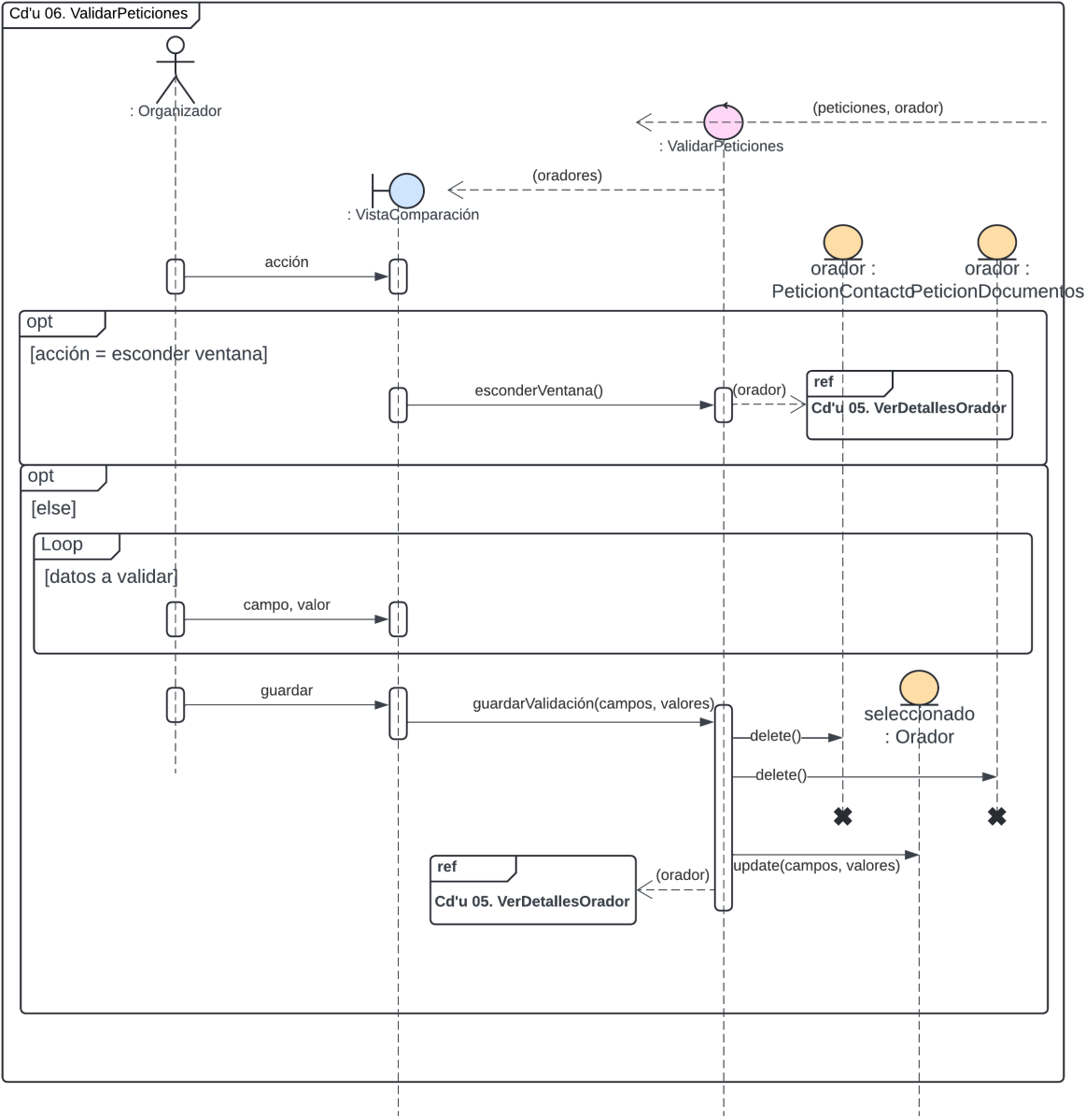


Figura 4.20: Diagrama de secuencias del caso de uso 06.

## 4.2.7. Caso de uso 07. Modificar Información

### Diagrama de actividades

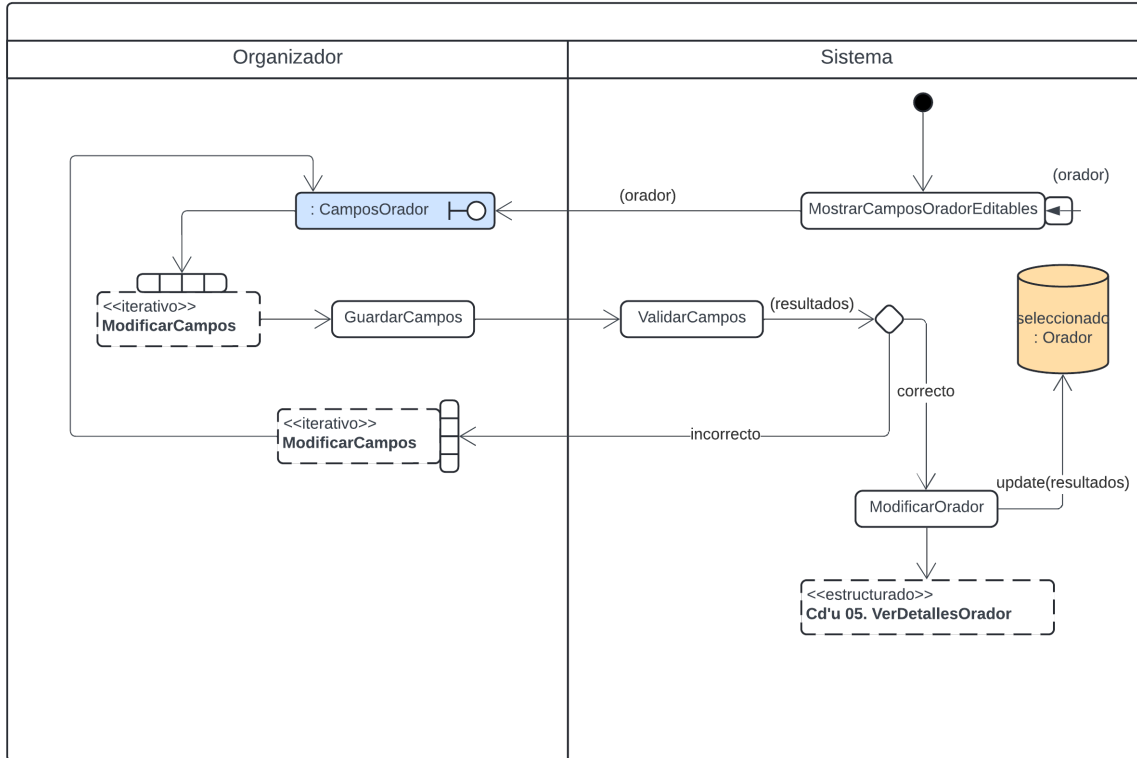


Figura 4.21: Diagrama de actividades del caso de uso 07.

### Diagrama de clases

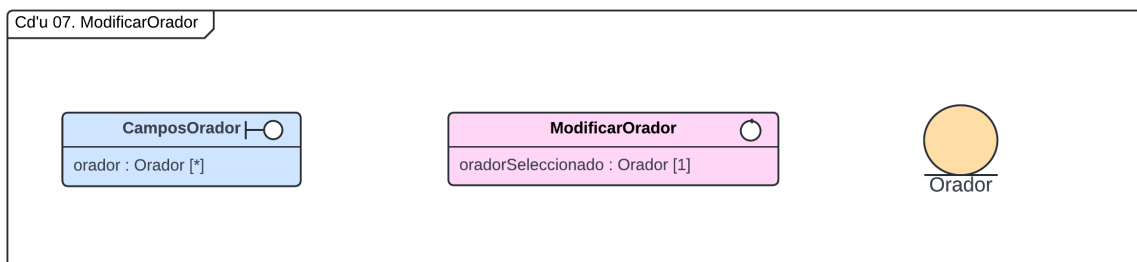


Figura 4.22: Diagrama de clases del caso de uso 07.

# Diagrama de secuencias

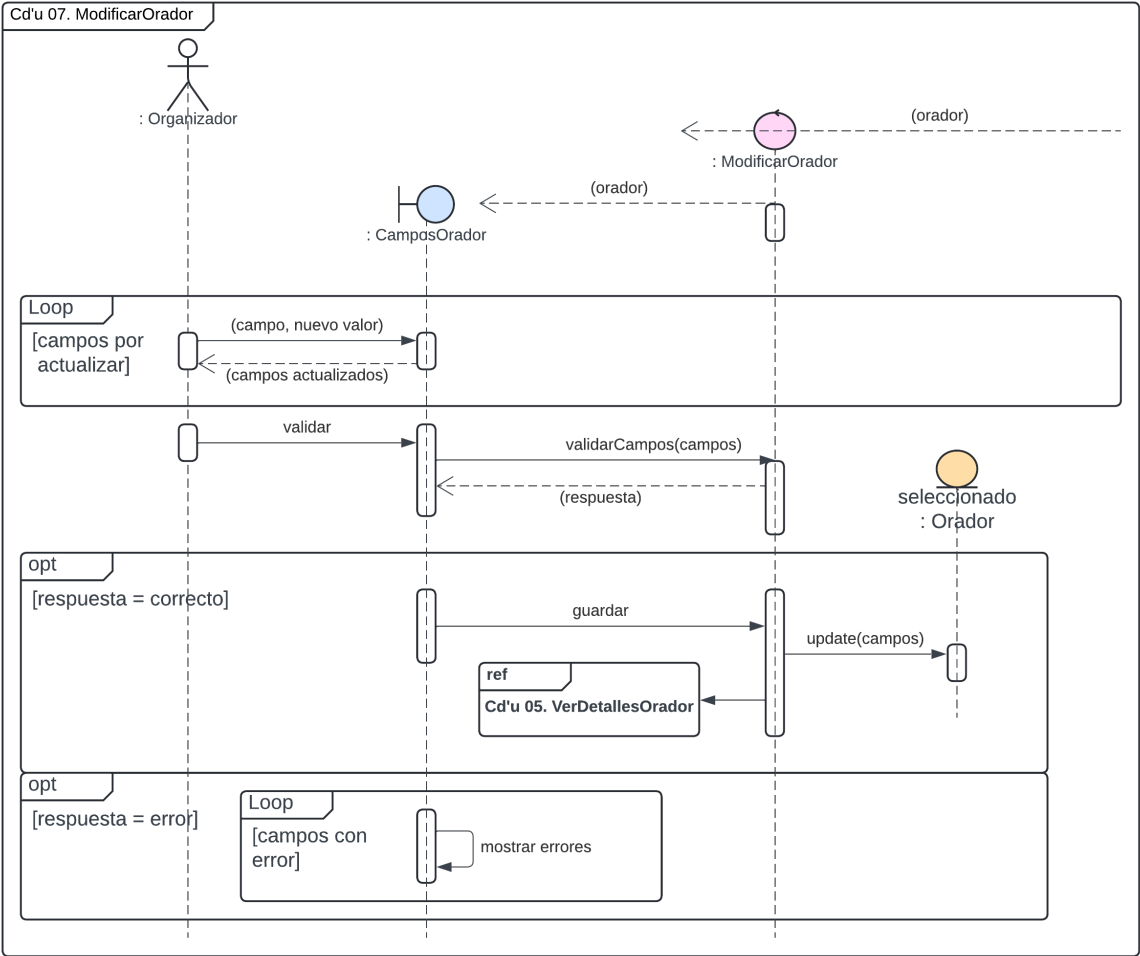


Figura 4.23: Diagrama de secuencias del caso de uso 07.

## 4.2.8. Caso de uso 08. AñadirContactoVinculado

### Diagrama de actividades

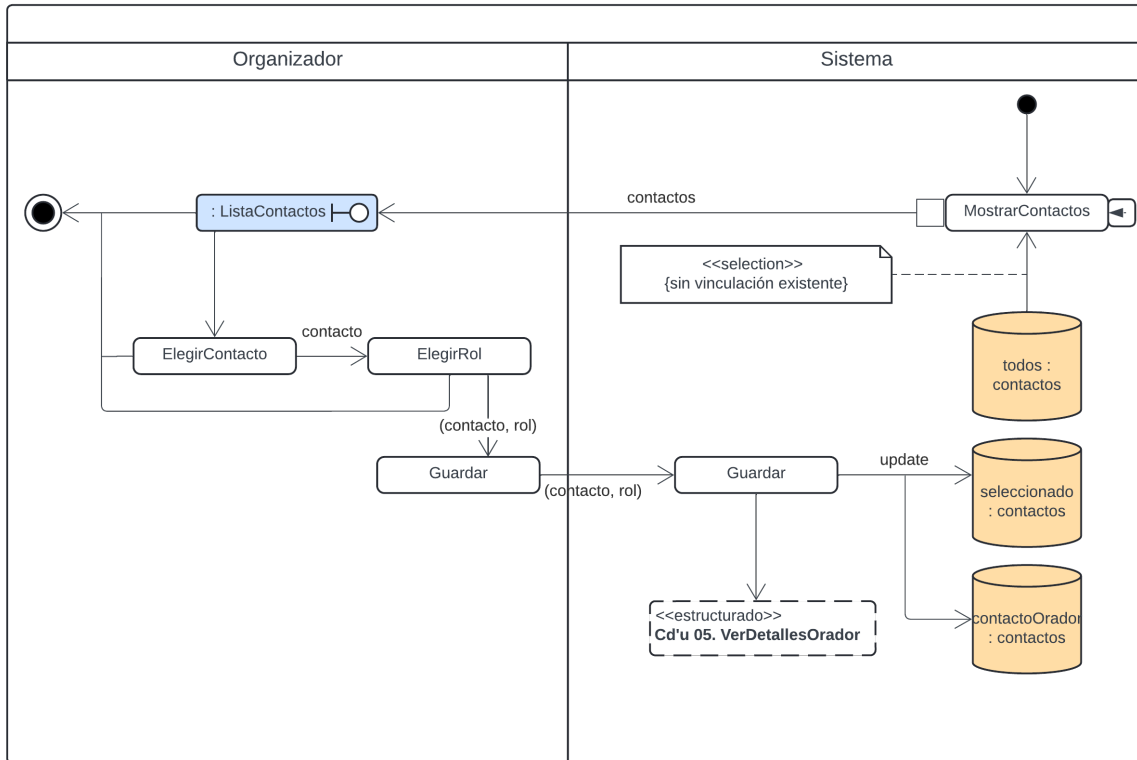


Figura 4.24: Diagrama de actividades del caso de uso 08.

### Diagrama de clases

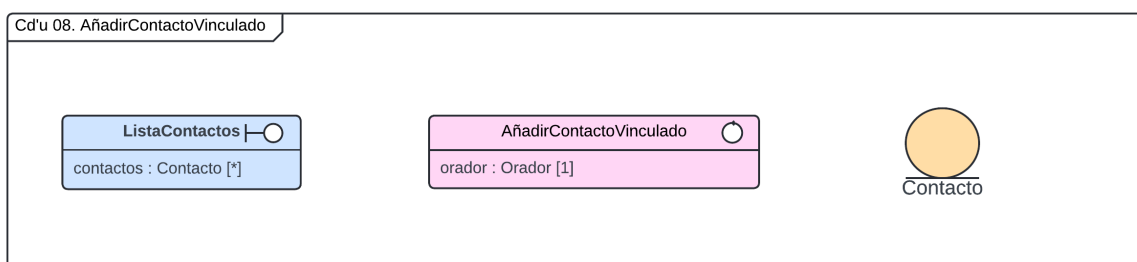


Figura 4.25: Diagrama de clases del caso de uso 08.

# Diagrama de secuencias

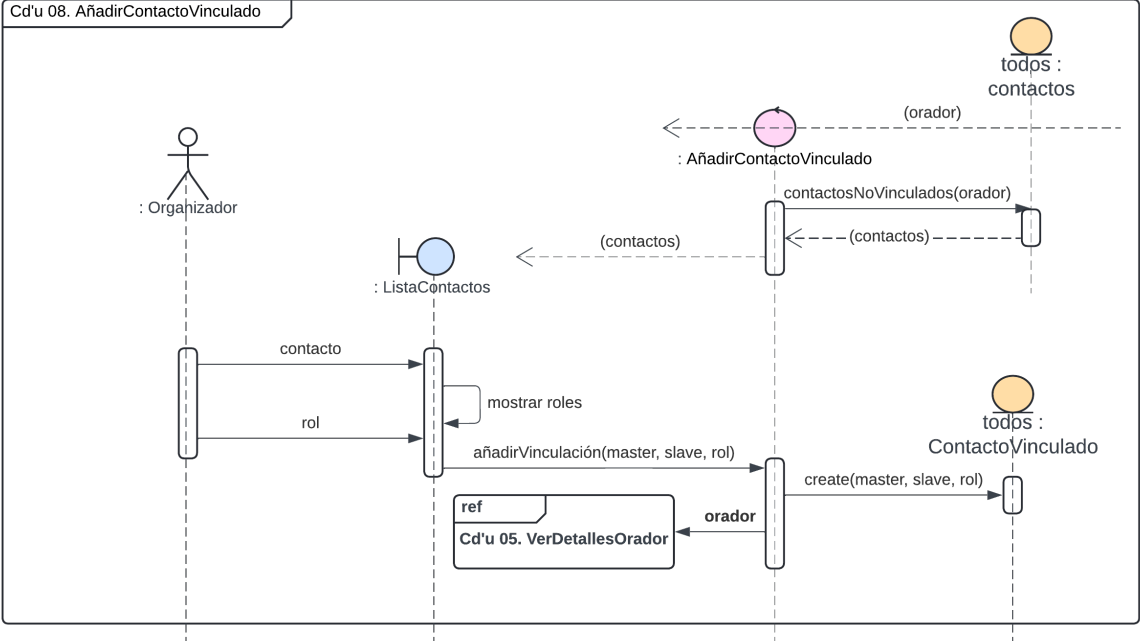


Figura 4.26: Diagrama de secuencias del caso de uso 08.

## 4.2.9. Caso de uso 09. BorrarSesion

### Diagrama de actividades

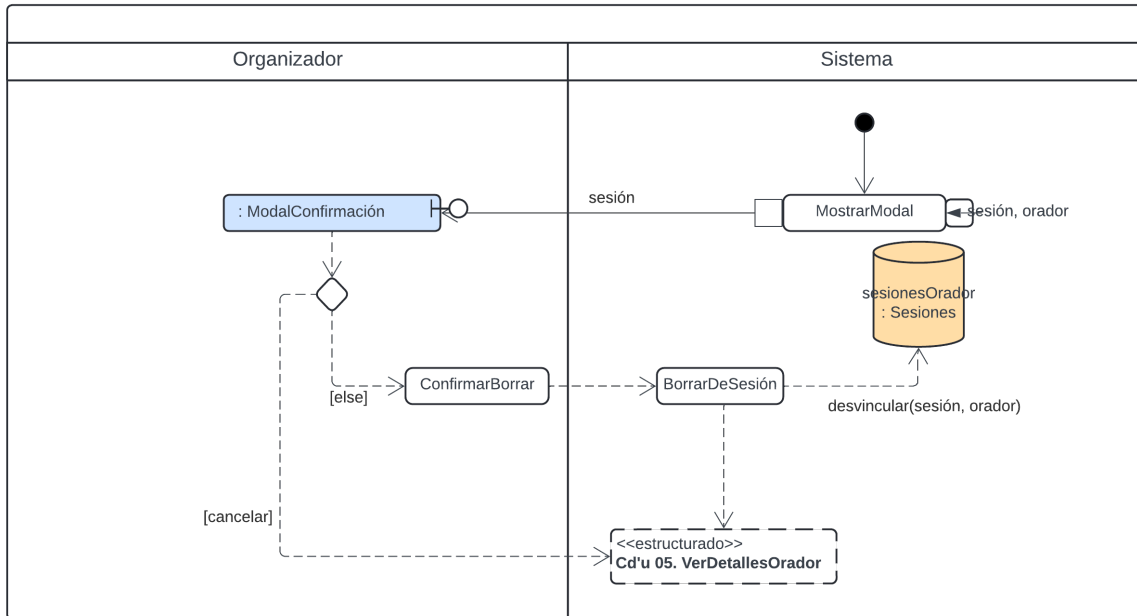


Figura 4.27: Diagrama de actividades del caso de uso 09.

### Diagrama de clases

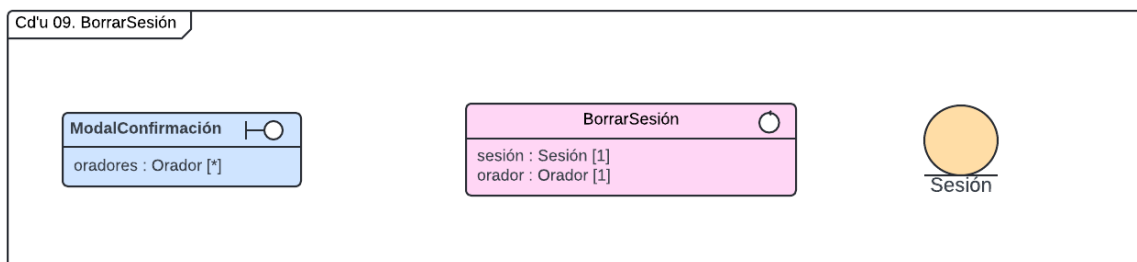


Figura 4.28: Diagrama de clases del caso de uso 09.

# Diagrama de secuencias

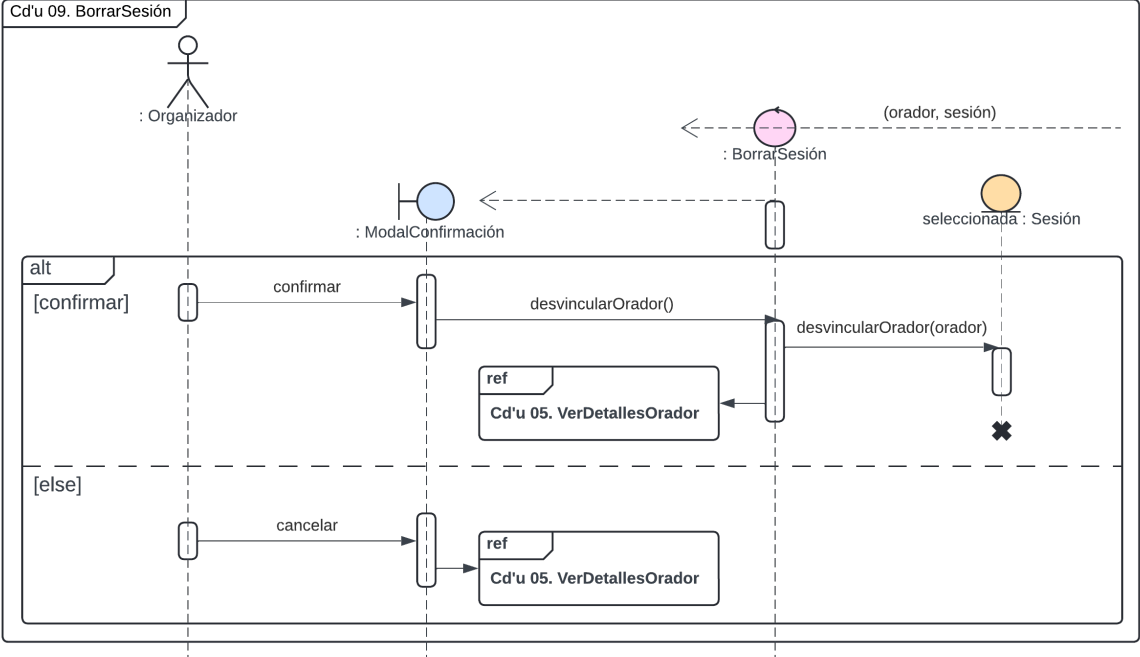


Figura 4.29: Diagrama de secuencias del caso de uso 09.

## 4.2.10. Caso de uso 10. AñadirSesion

### Diagrama de actividades

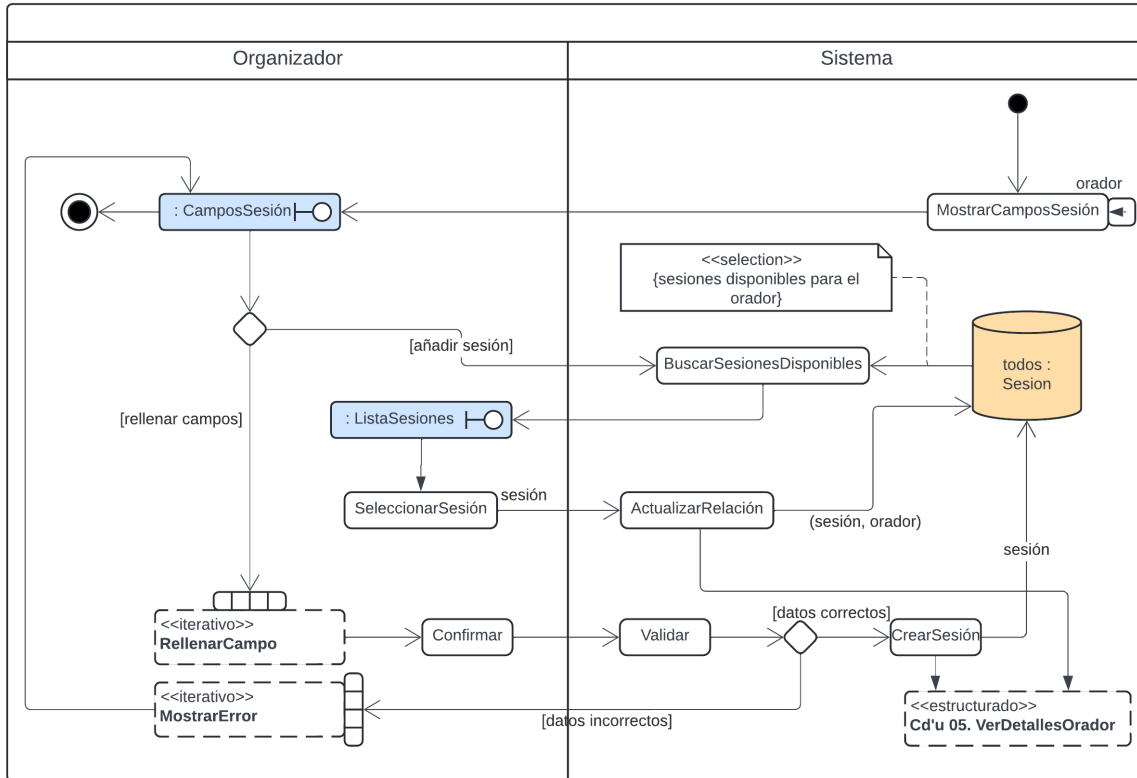


Figura 4.30: Diagrama de actividades del caso de uso 10.

### Diagrama de clases

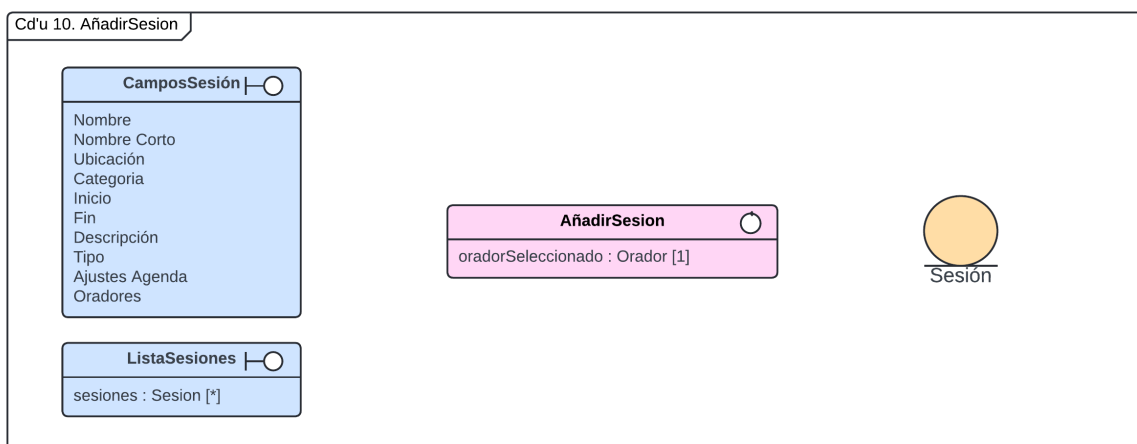


Figura 4.31: Diagrama de clases del caso de uso 10.

## Diagrama de secuencias

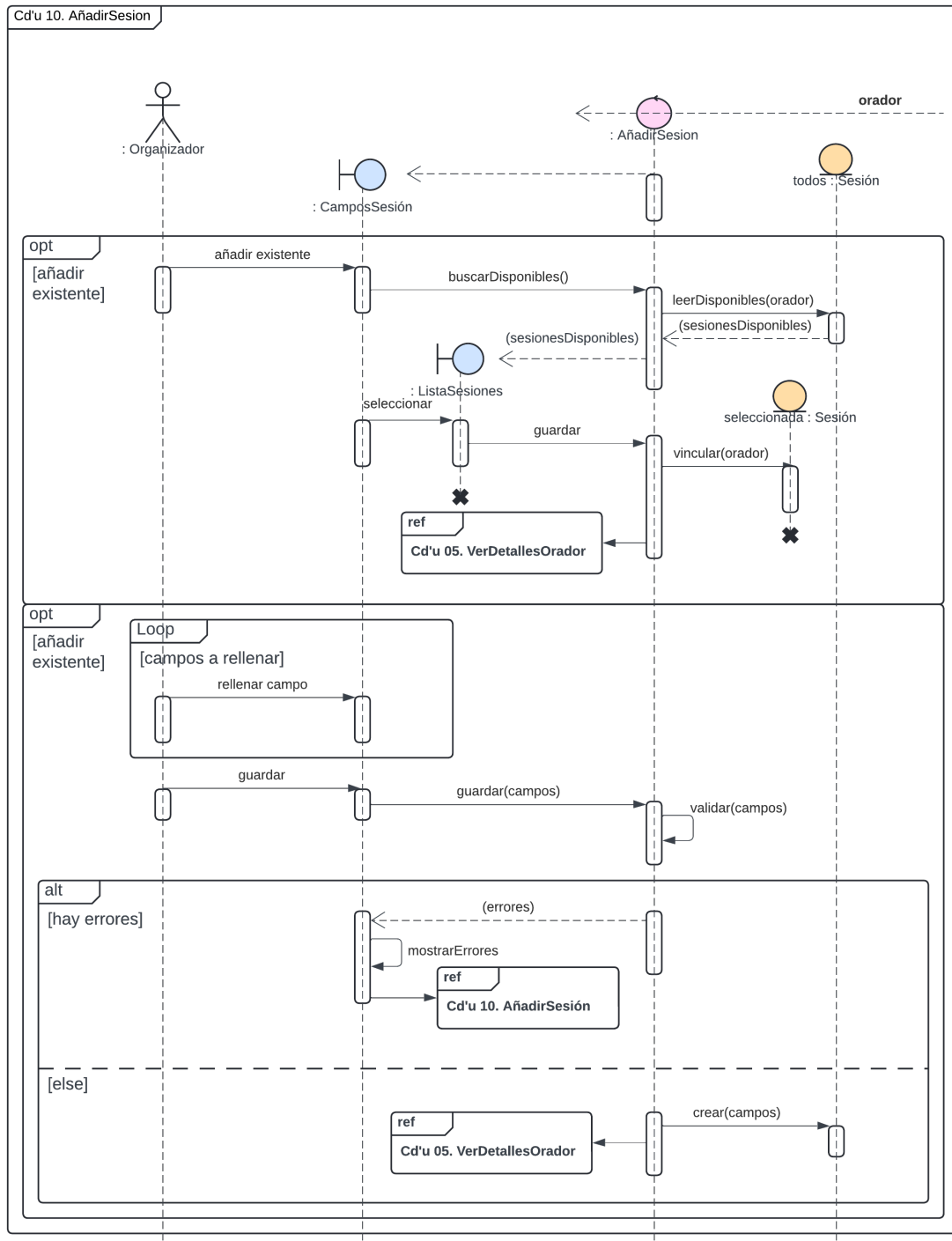


Figura 4.32: Diagrama de secuencias del caso de uso 10.

## 4.2.11. Caso de uso 11. Modificar Documentos

### Diagrama de actividades

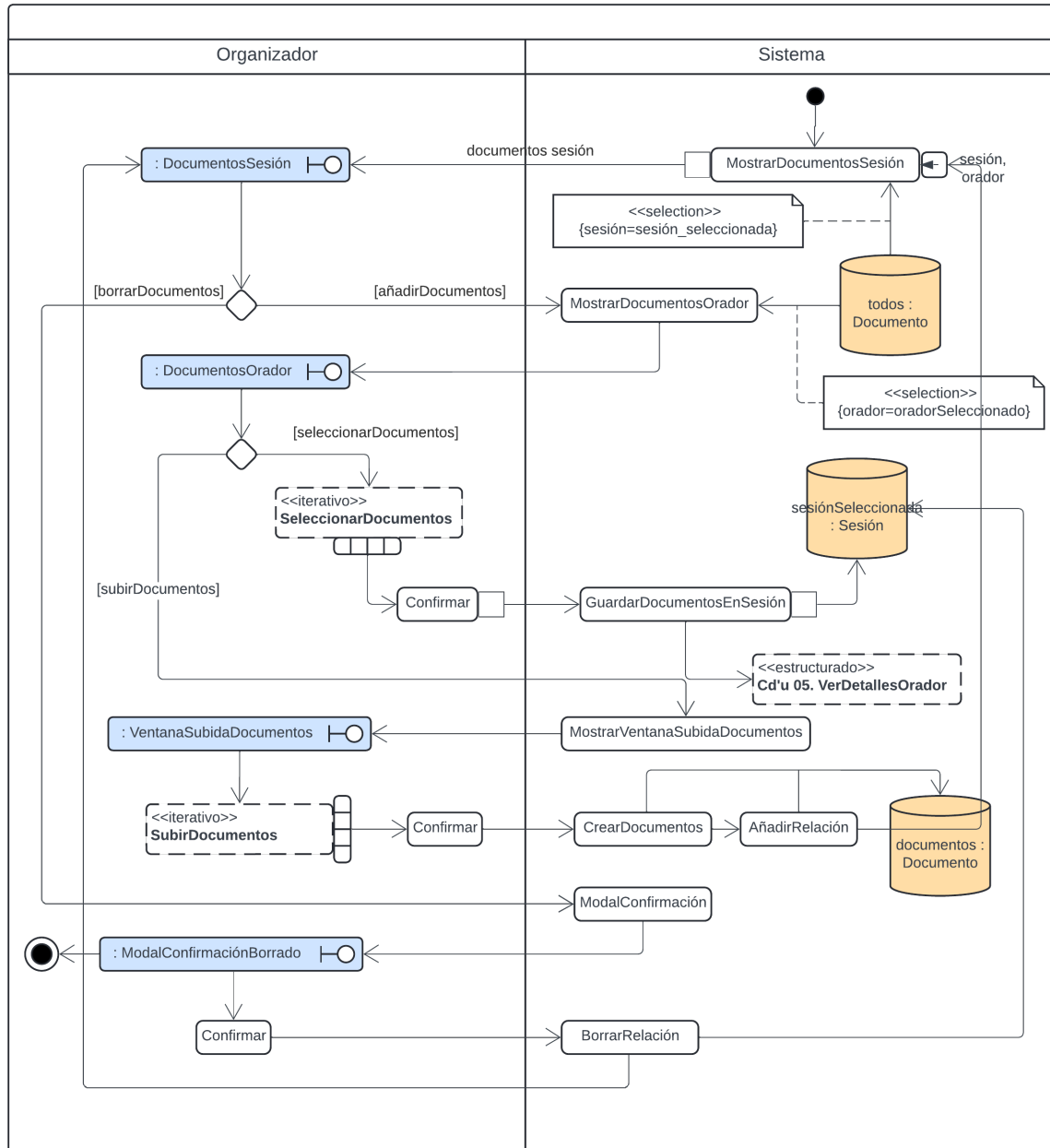


Figura 4.33: Diagrama de actividades del caso de uso 11.

# Diagrama de clases

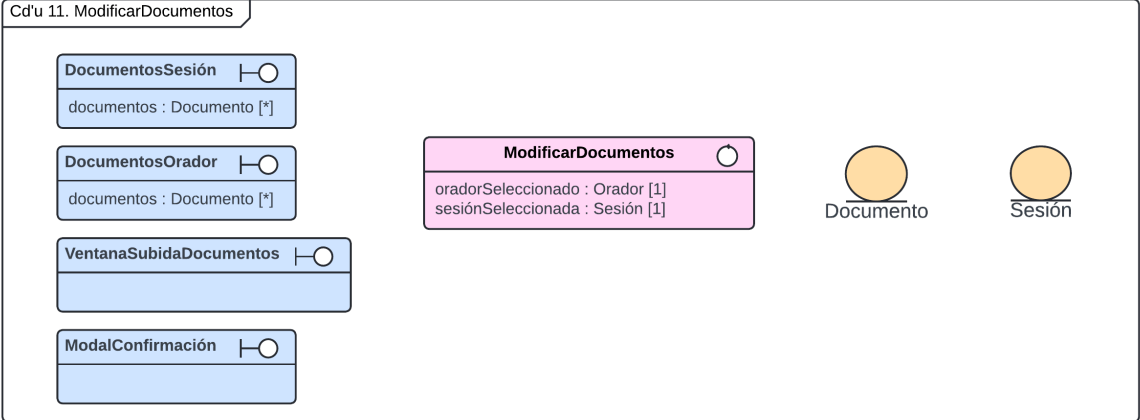


Figura 4.34: Diagrama de clases del caso de uso 11.

## Diagrama de secuencias

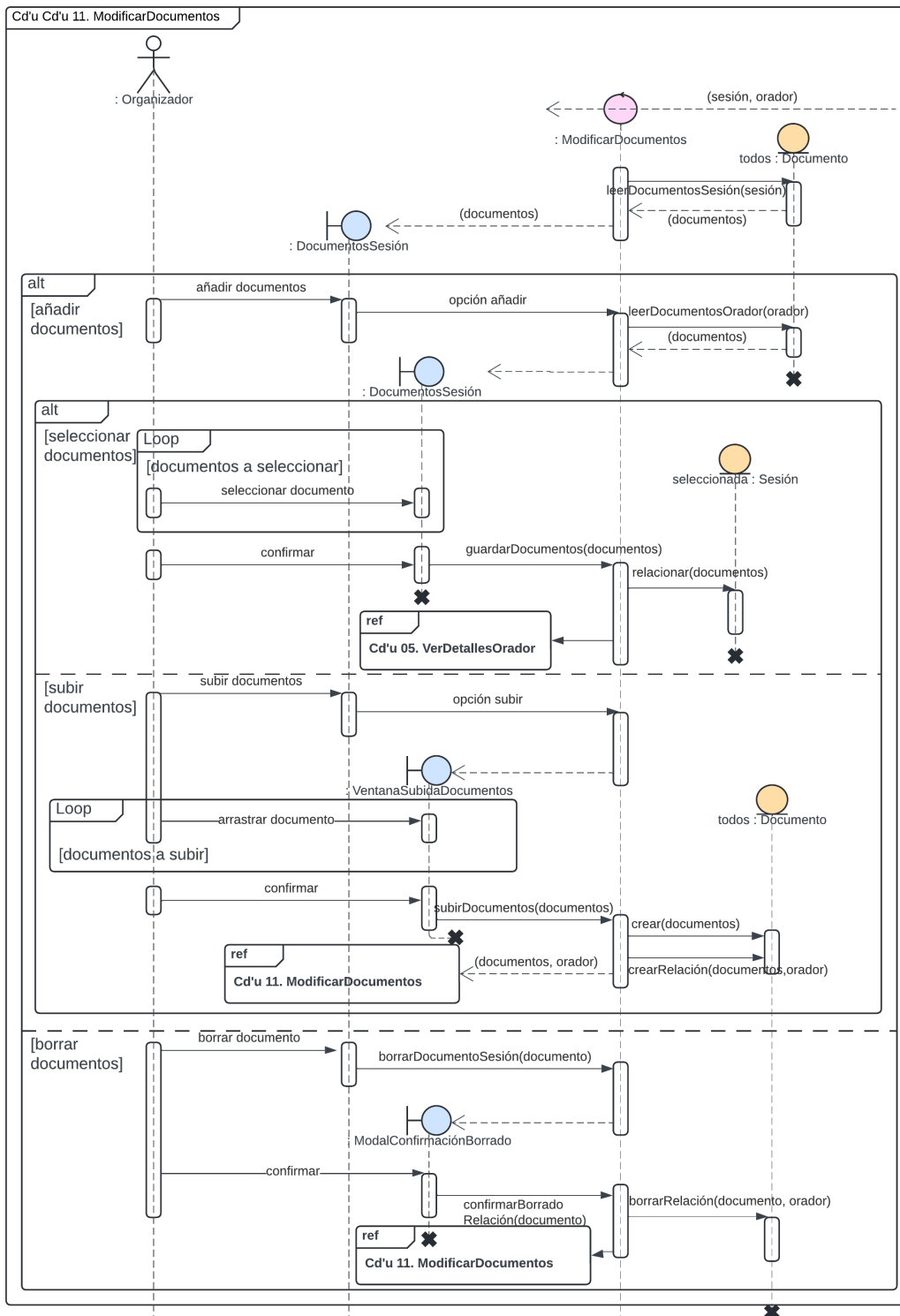


Figura 4.35: Diagrama de secuencias del caso de uso 11.

## 4.2.12. Caso de uso 12. VerArtículos

### Diagrama de actividades

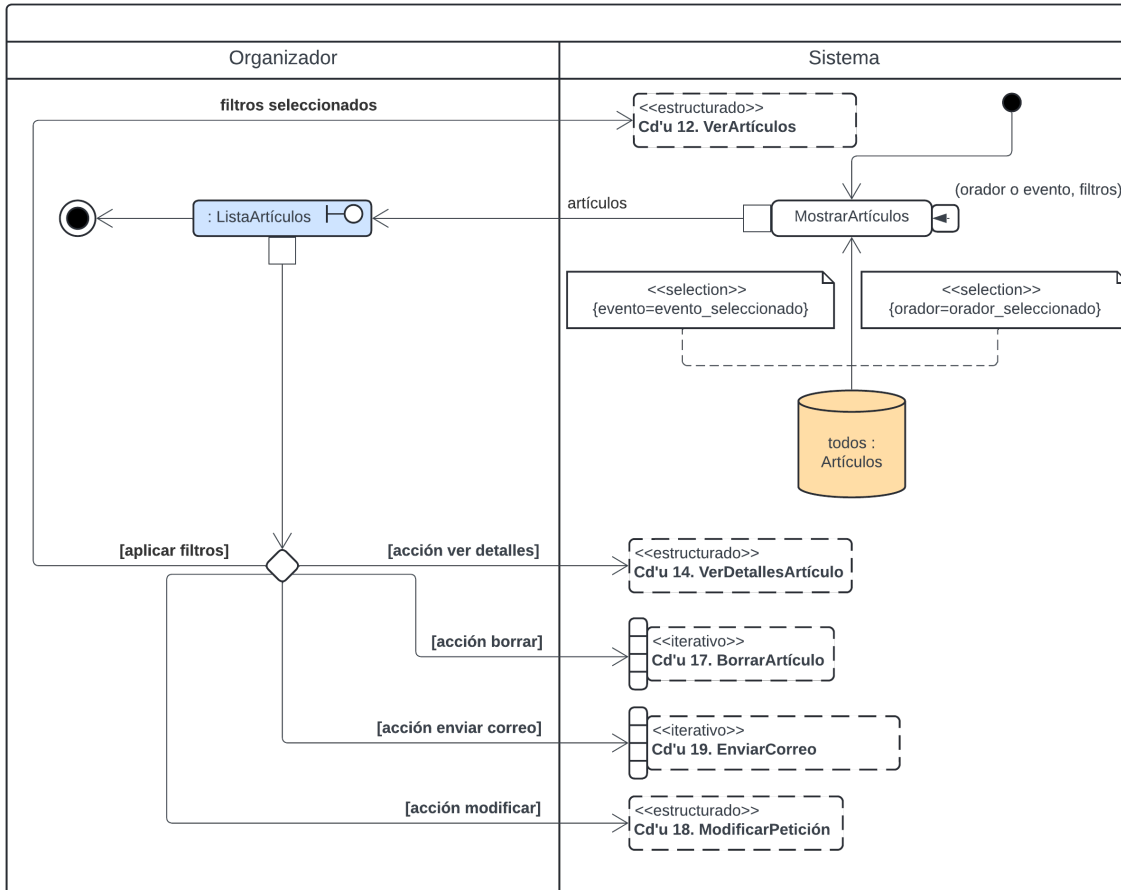


Figura 4.36: Diagrama de actividades del caso de uso 12.

### Diagrama de clases

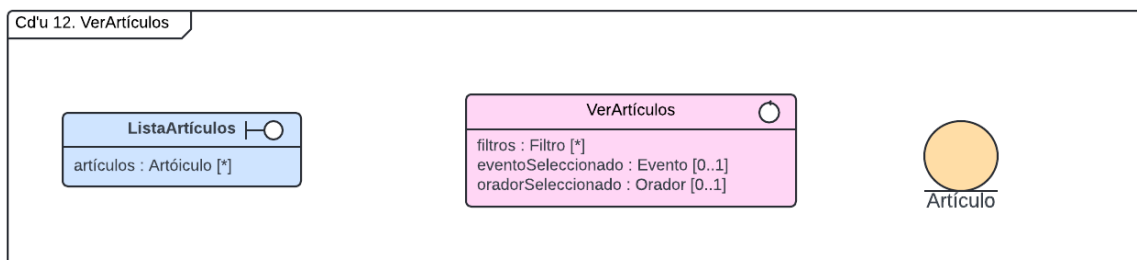


Figura 4.37: Diagrama de clases del caso de uso 12.

## Diagrama de secuencias

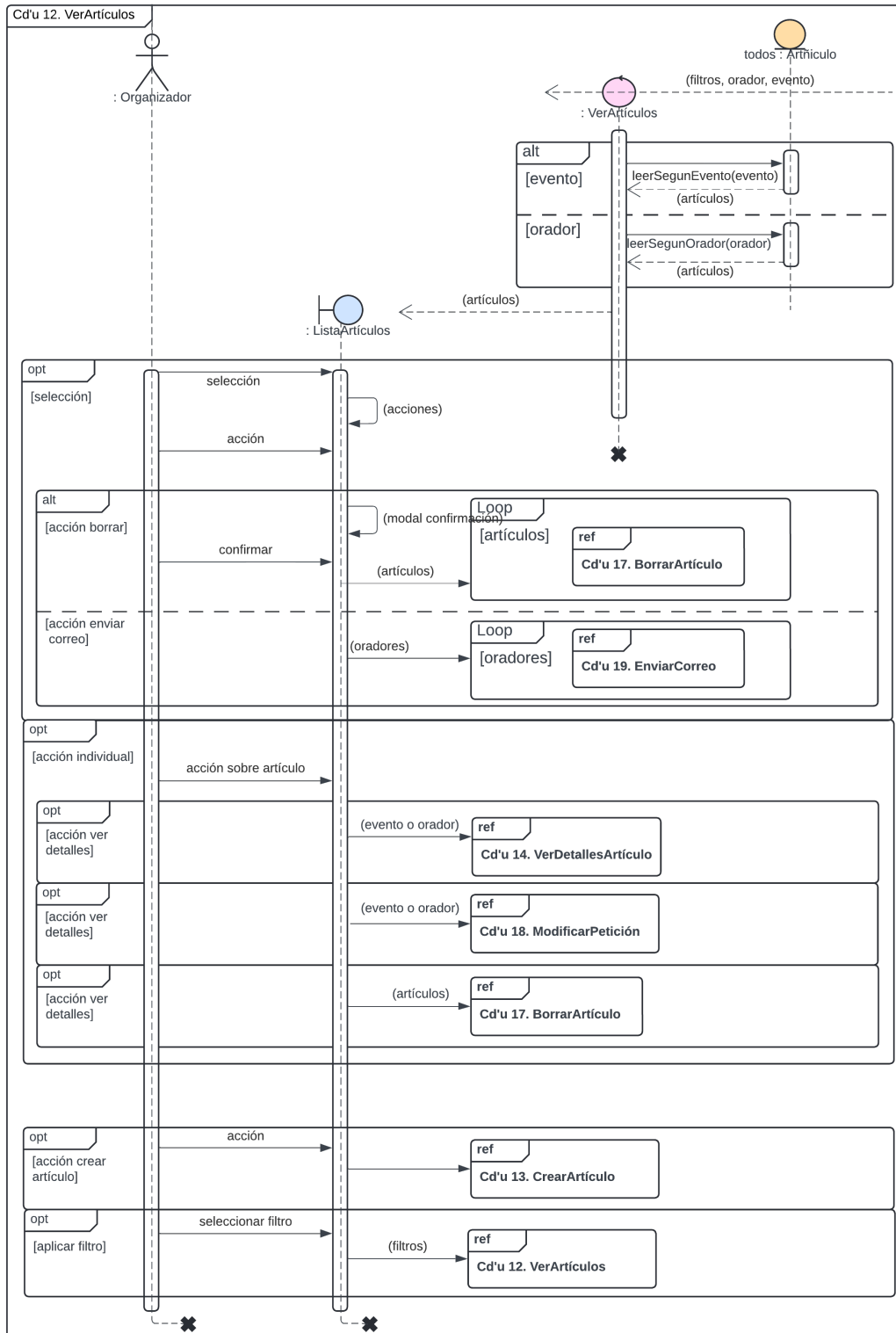


Figura 4.38: Diagrama de secuencias del caso de uso 12.

### 4.2.13. Caso de uso 13. CrearArtículo

#### Diagrama de actividades

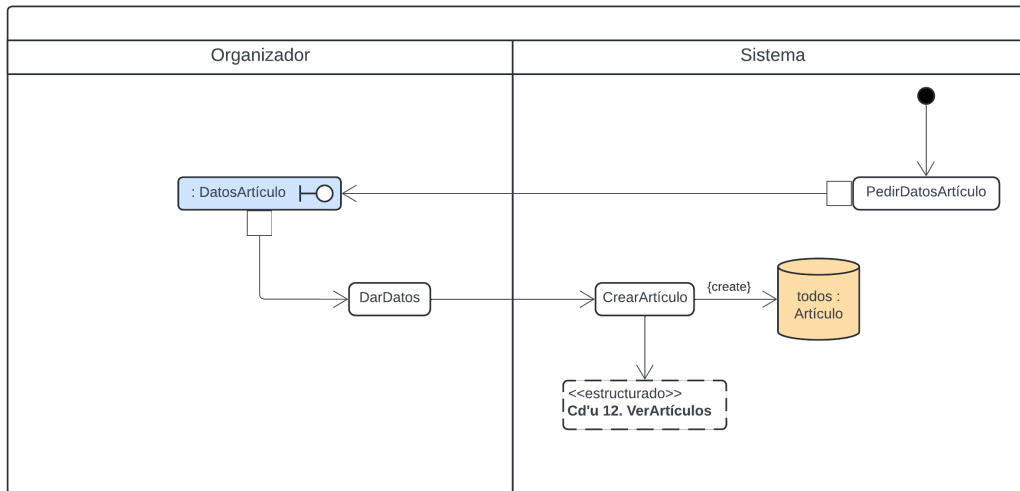


Figura 4.39: Diagrama de actividades del caso de uso 13.

#### Diagrama de clases

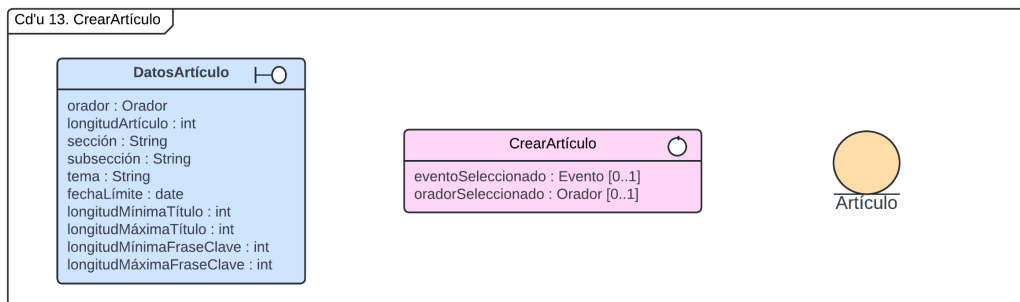


Figura 4.40: Diagrama de clases del caso de uso 13.

# Diagrama de secuencias

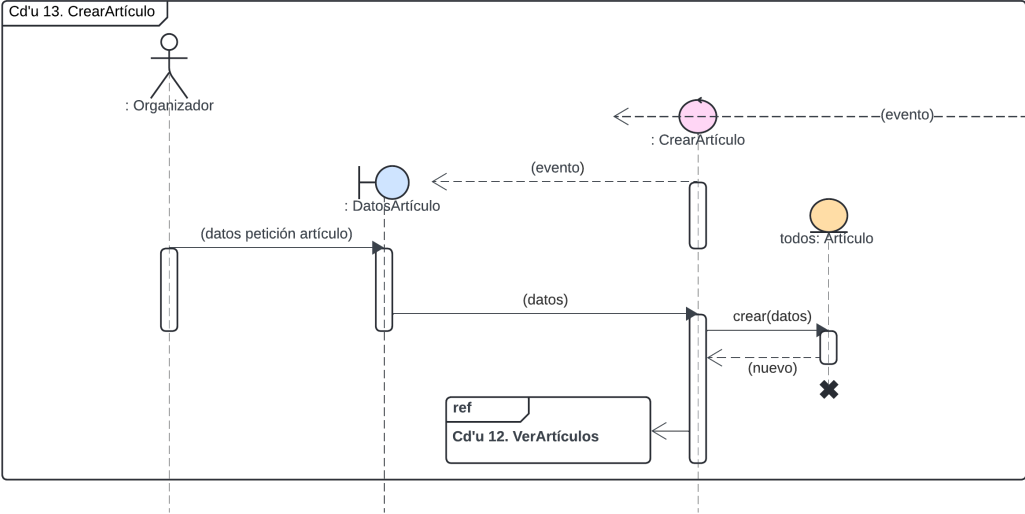


Figura 4.41: Diagrama de secuencias del caso de uso 13.

## 4.2.14. Caso de uso 14. VerDetallesArtículo

### Diagrama de actividades

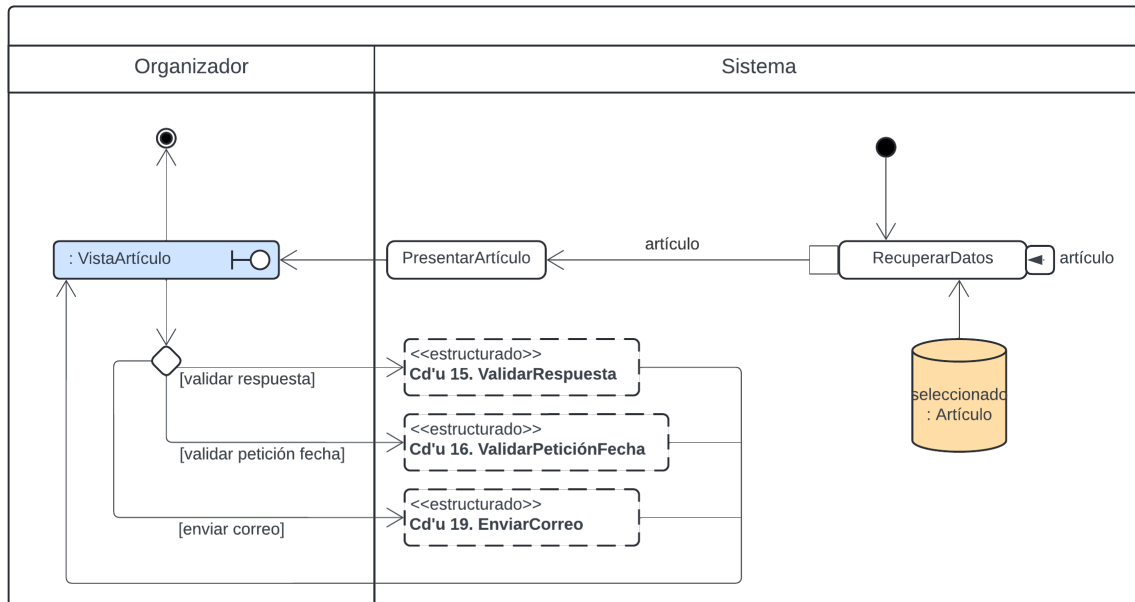


Figura 4.42: Diagrama de actividades del caso 14.

### Diagrama de clases

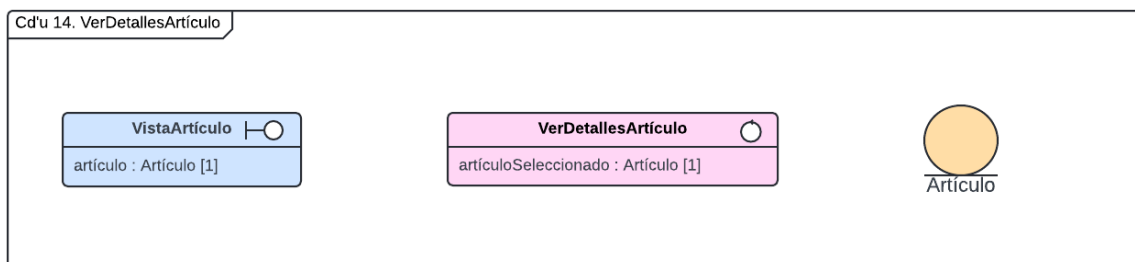


Figura 4.43: Diagrama de clases del caso 14.

# Diagrama de secuencias

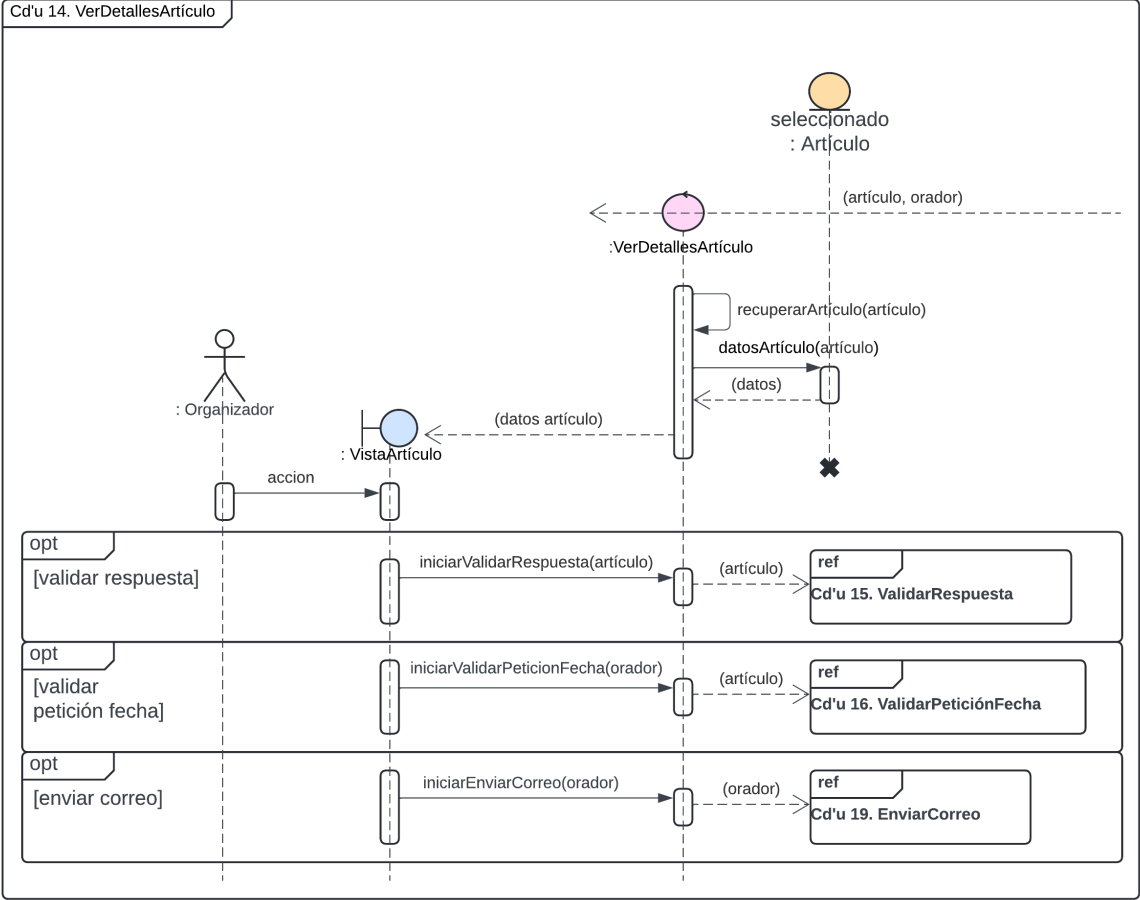


Figura 4.44: Diagrama de secuencias del caso 14.

## 4.2.15. Caso de uso 15. ValidarRespuesta

### Diagrama de actividades

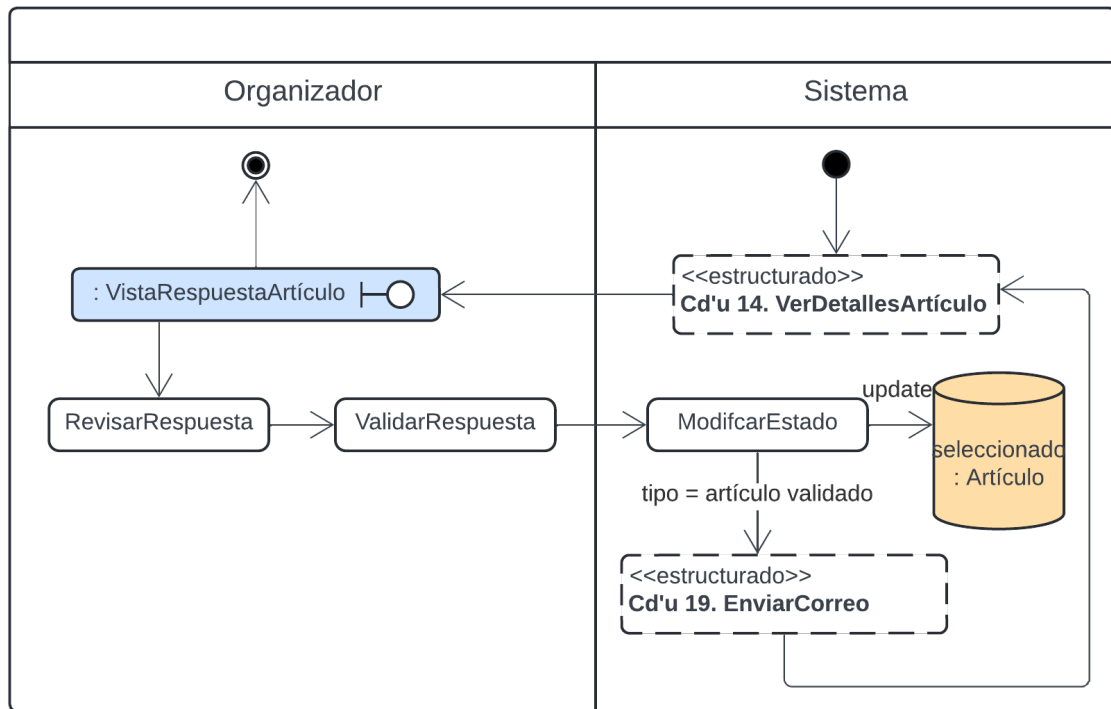


Figura 4.45: Diagrama de actividades del caso de uso 15.

### Diagrama de clases

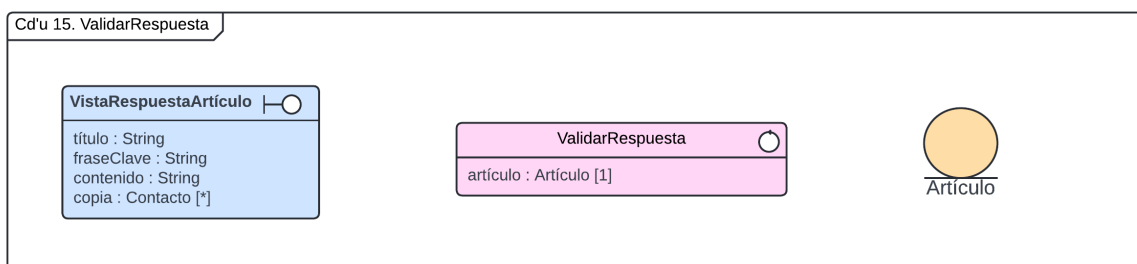


Figura 4.46: Diagrama de clases del caso de uso 15.

**Diagrama de secuencias**

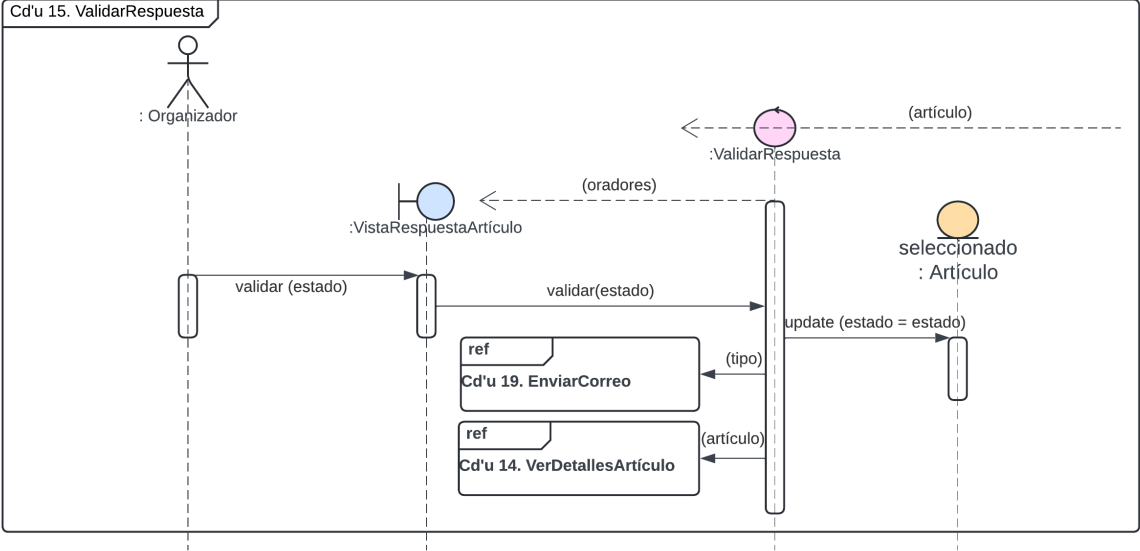


Figura 4.47: Diagrama de secuencias del caso de uso 15.

## 4.2.16. Caso de uso 16. ValidarPeticiónFecha

### Diagrama de actividades

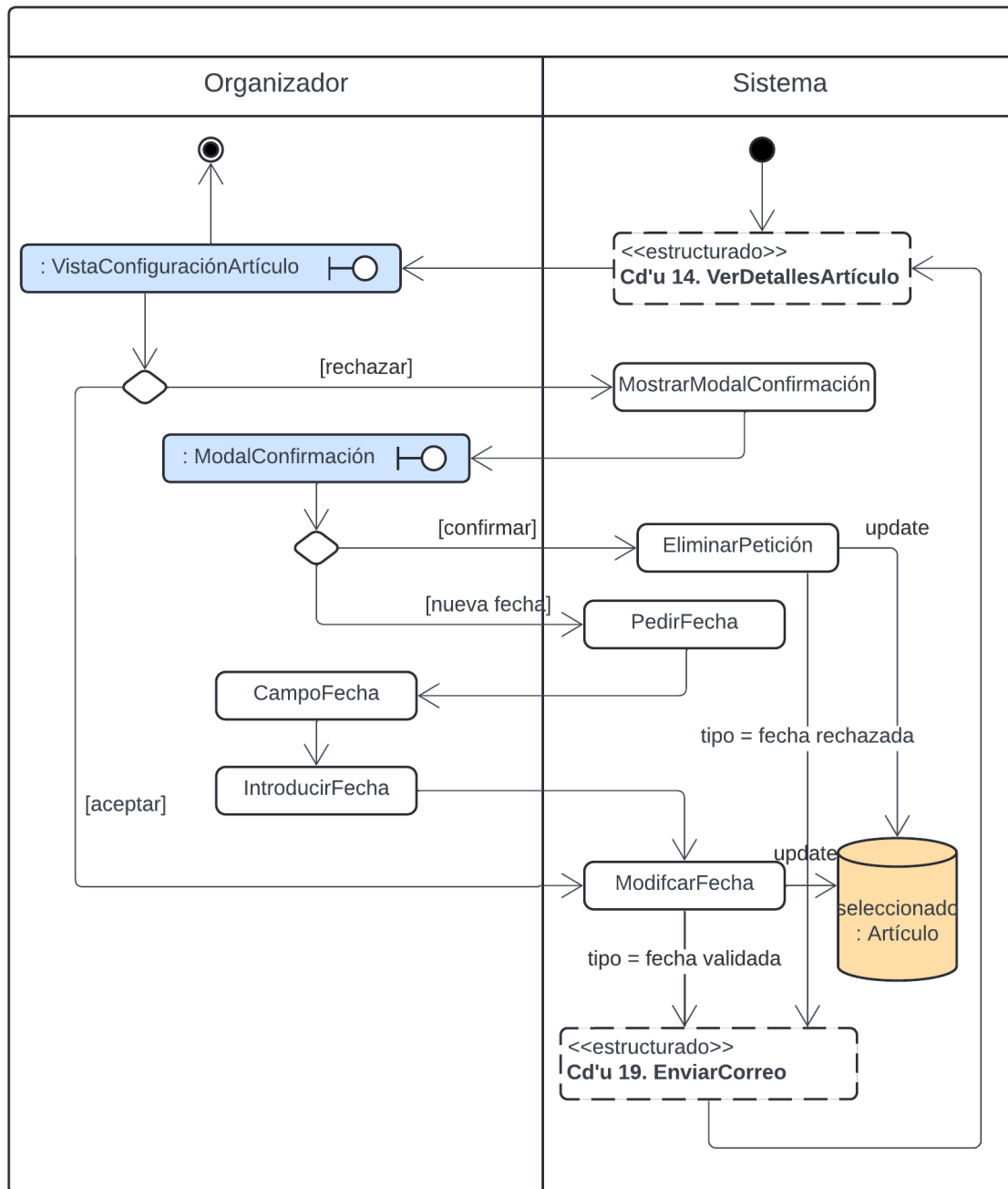


Figura 4.48: Diagrama de actividades del caso de 16.

## Diagrama de clases

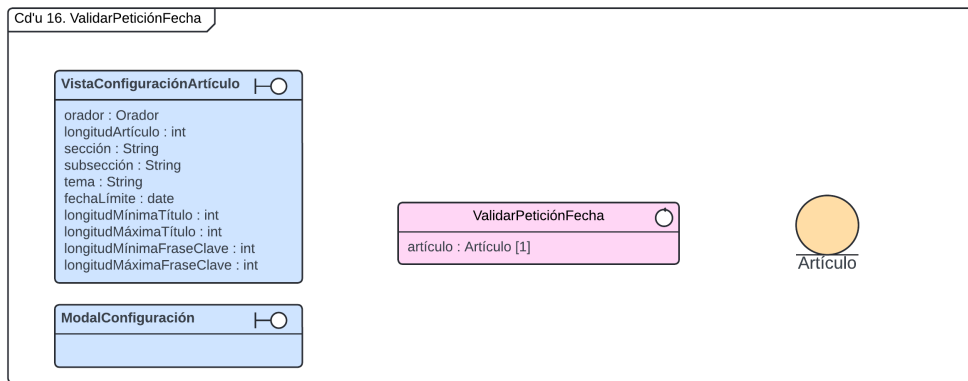


Figura 4.49: Diagrama de clases del caso de 16.

# Diagrama de secuencias

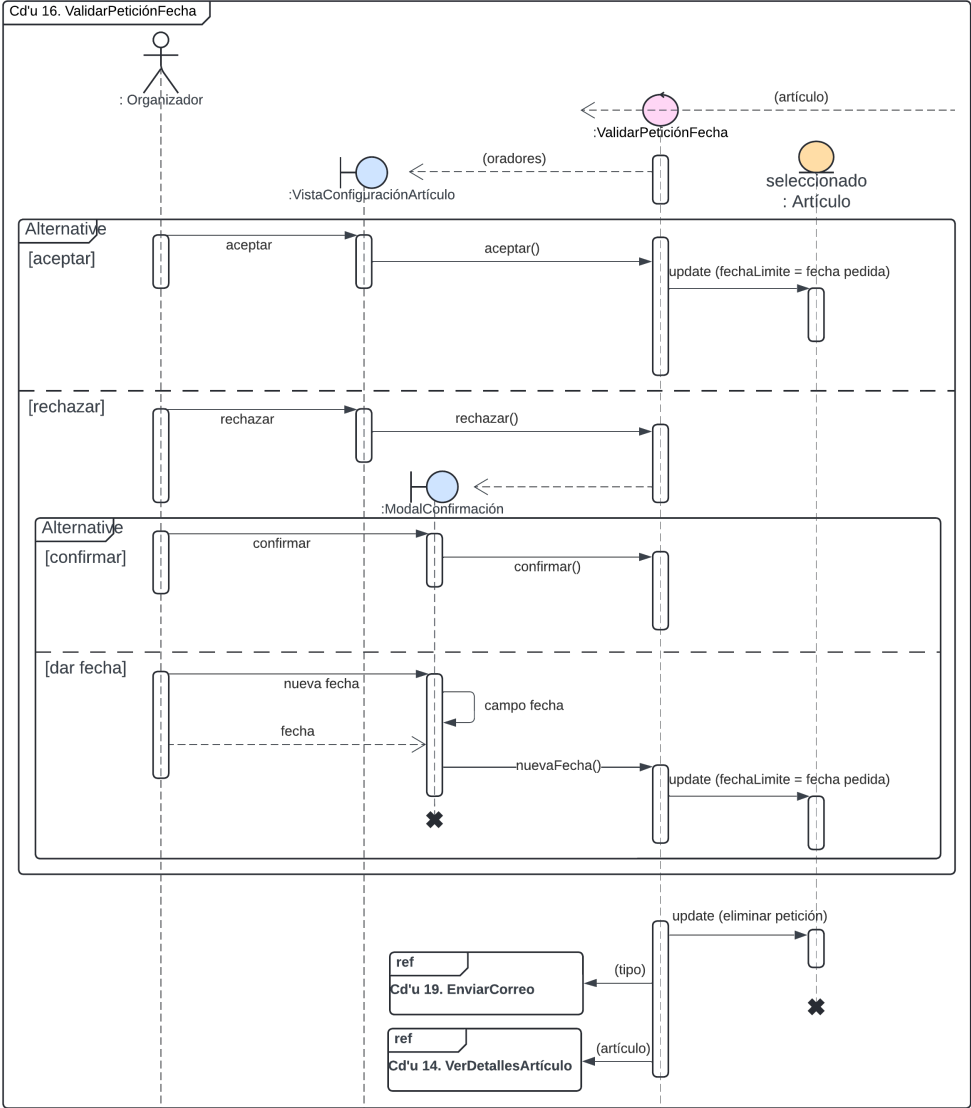


Figura 4.50: Diagrama de secuencias del caso de 16.

## 4.2.17. Caso de uso 17. BorrarArtículo

### Diagrama de actividades

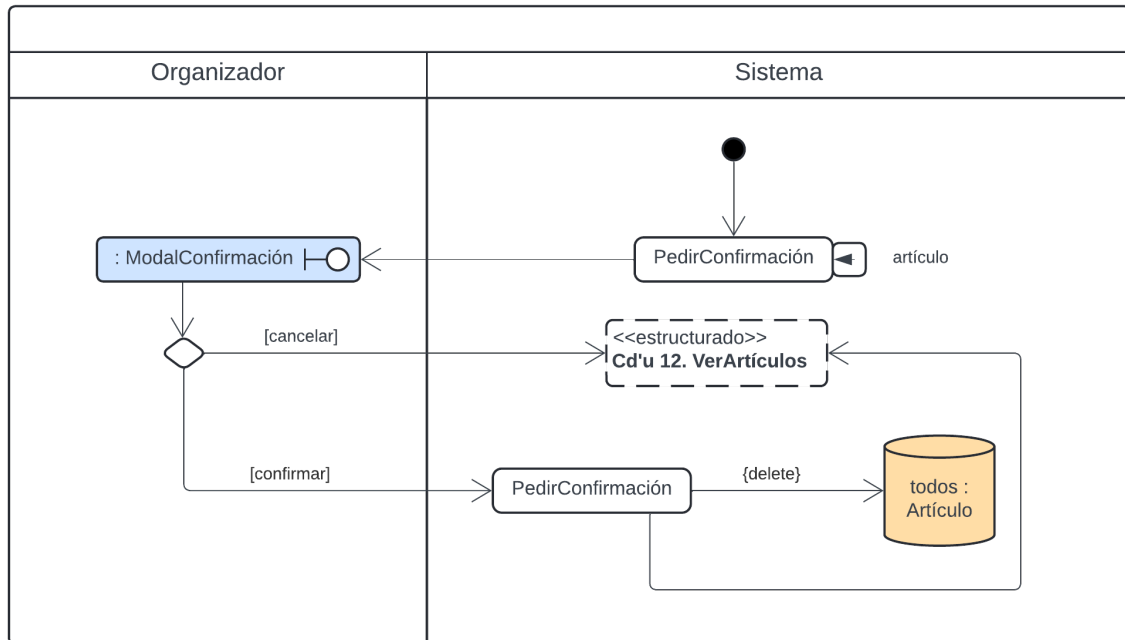


Figura 4.51: Diagrama de actividades del caso de uso 17.

### Diagrama de clases

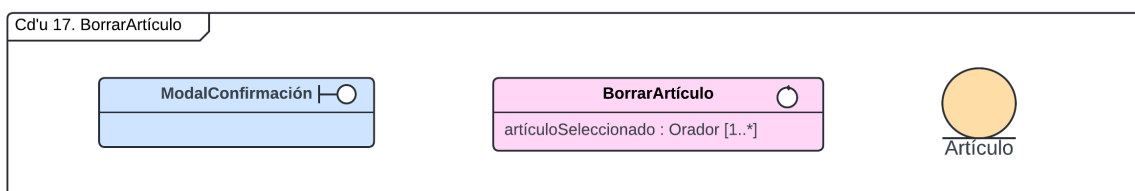


Figura 4.52: Diagrama de clases del caso de uso 17.

# Diagrama de secuencias

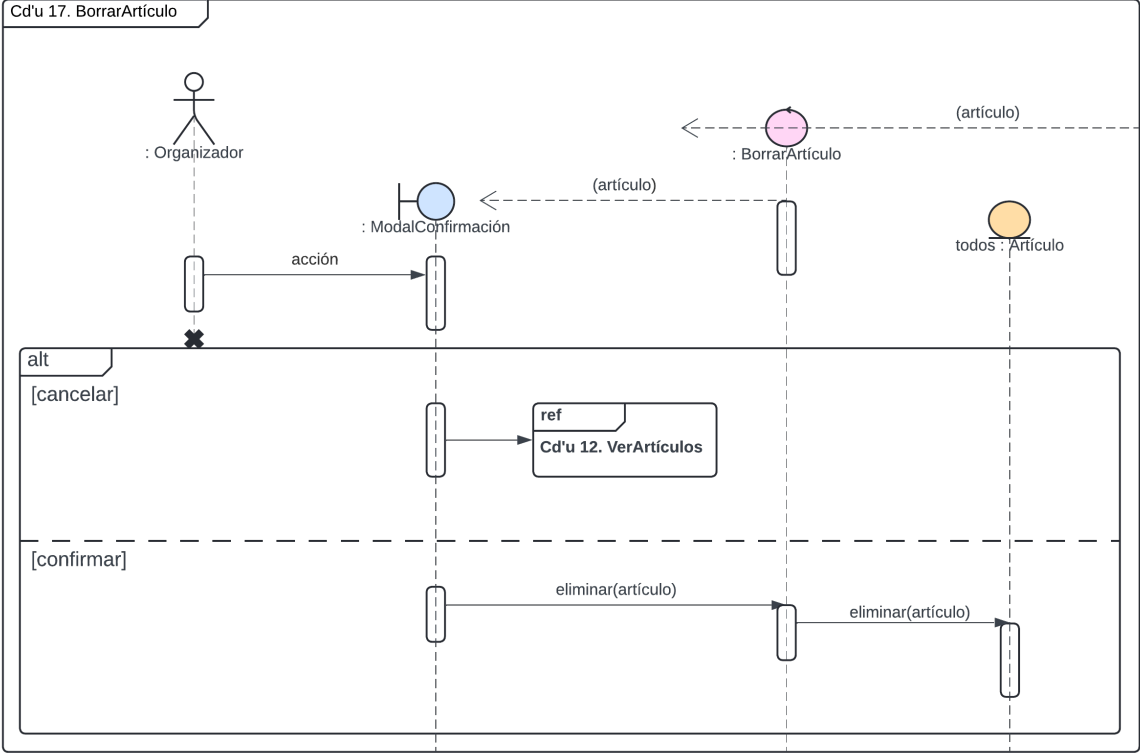


Figura 4.53: Diagrama de secuencias del caso de uso 17.

## 4.2.18. Caso de uso 18. ModificarPetición

### Diagrama de actividades

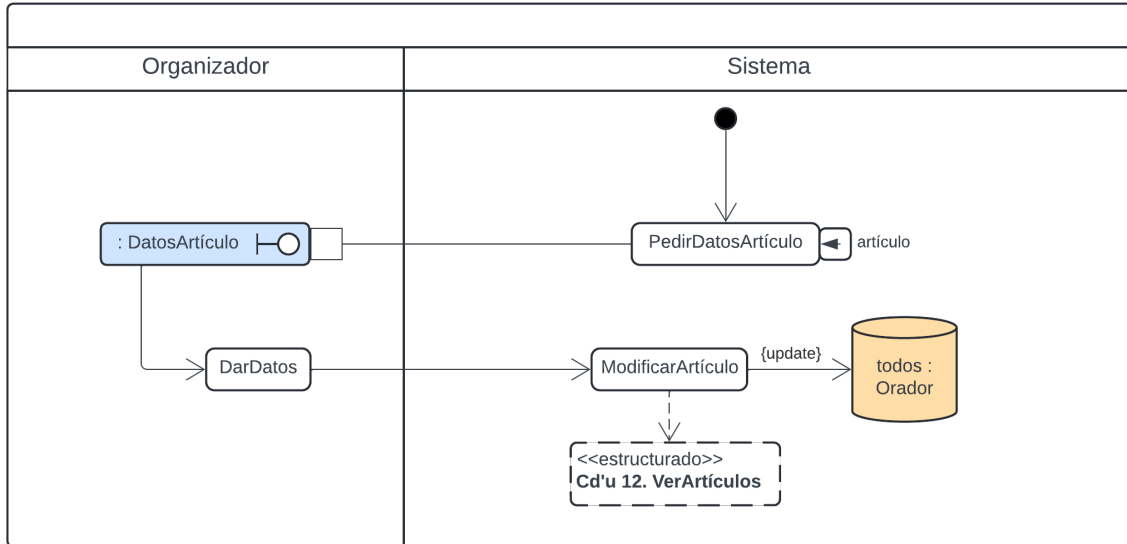


Figura 4.54: Diagrama de actividades del caso de uso 18.

### Diagrama de clases

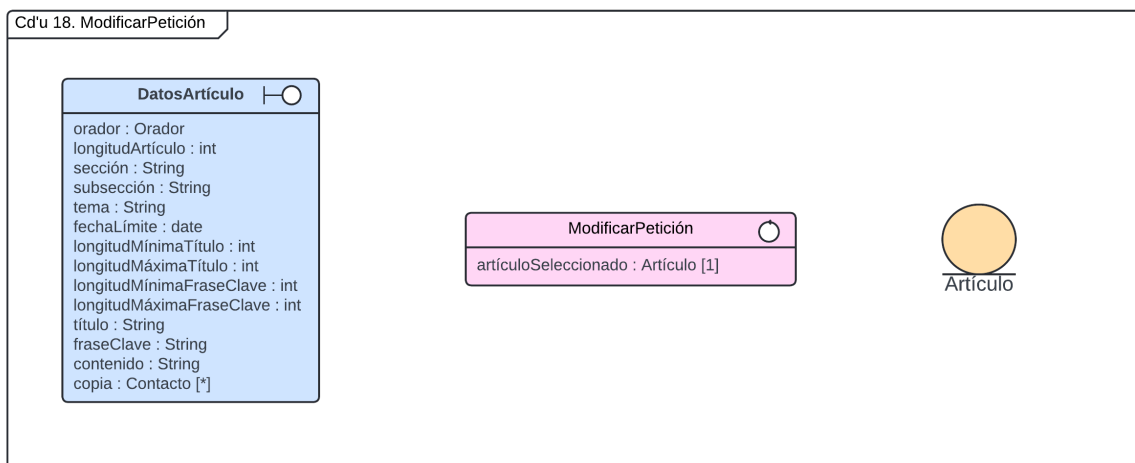


Figura 4.55: Diagrama de clases del caso de uso 18.

# Diagrama de secuencias

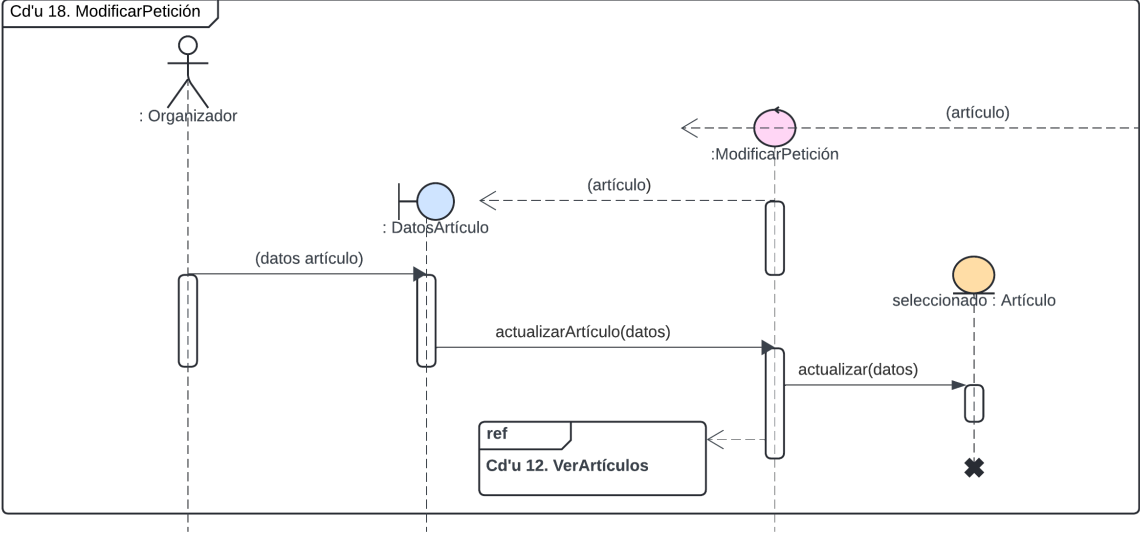


Figura 4.56: Diagrama de secuencias del caso de uso 18.

## 4.2.19. Caso de uso 19. EnviarCorreo

### Diagrama de actividades

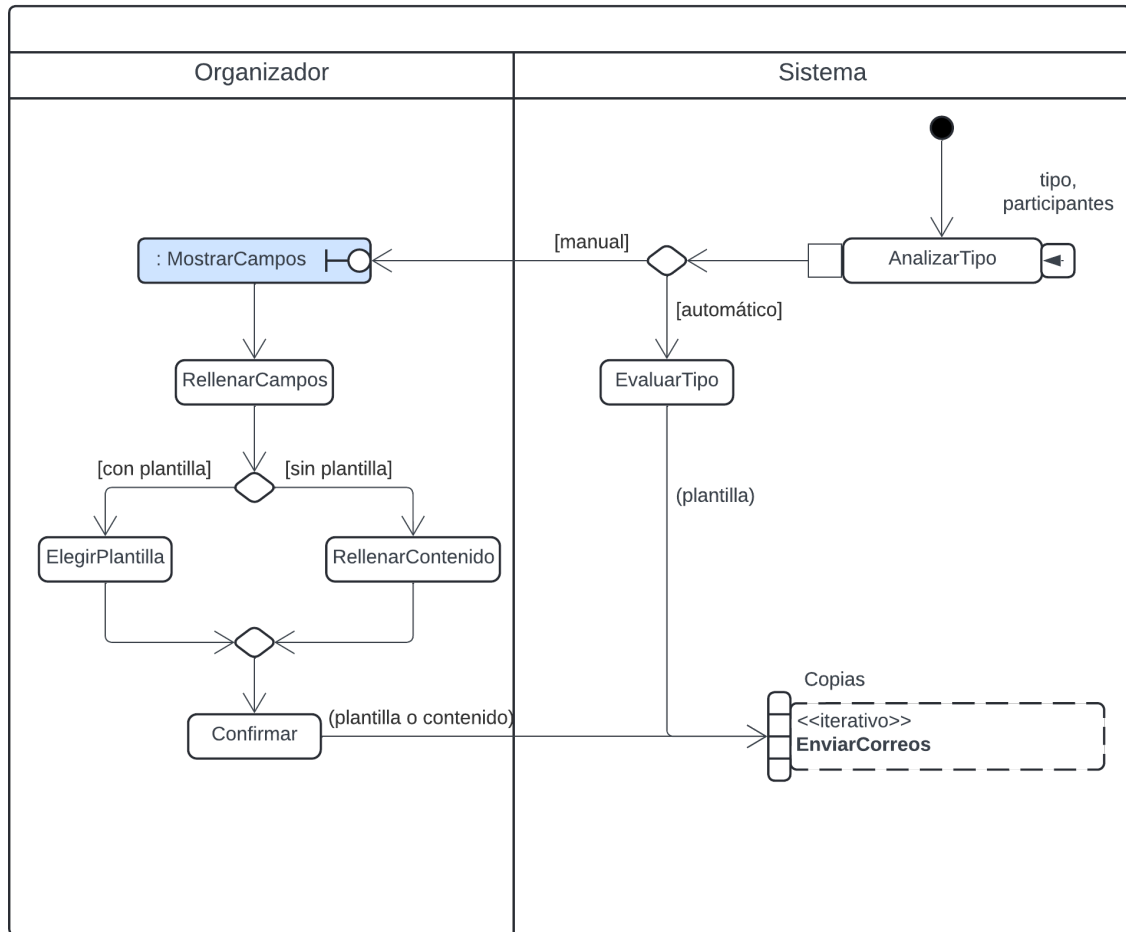


Figura 4.57: Diagrama de actividades del caso de uso 19.

### Diagrama de clases

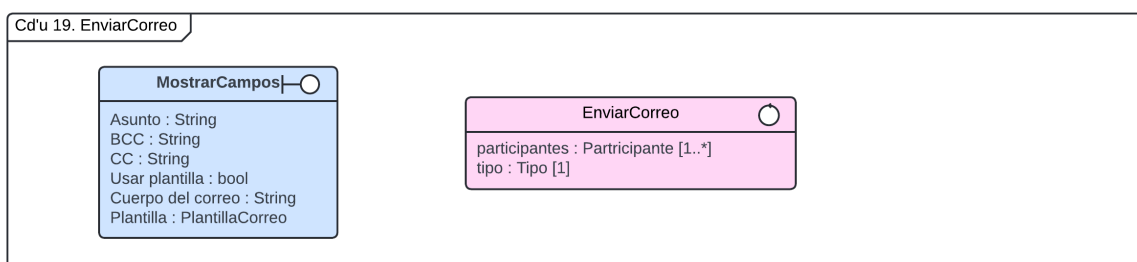


Figura 4.58: Diagrama de clases del caso de uso 19.

# Diagrama de secuencias

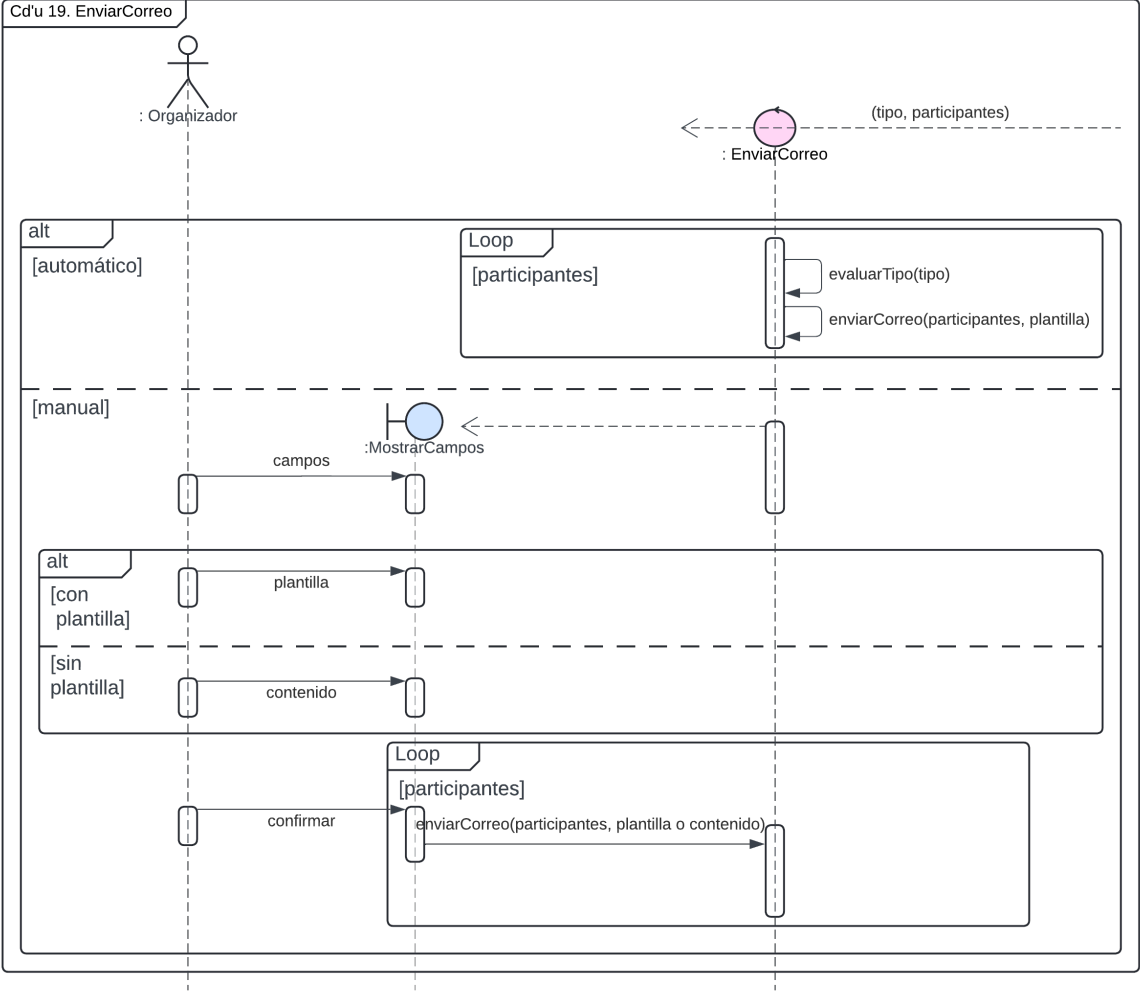


Figura 4.59: Diagrama de secuencias del caso de uso 19.

## 4.2.20. Caso de uso 20. Login

### Diagrama de actividades

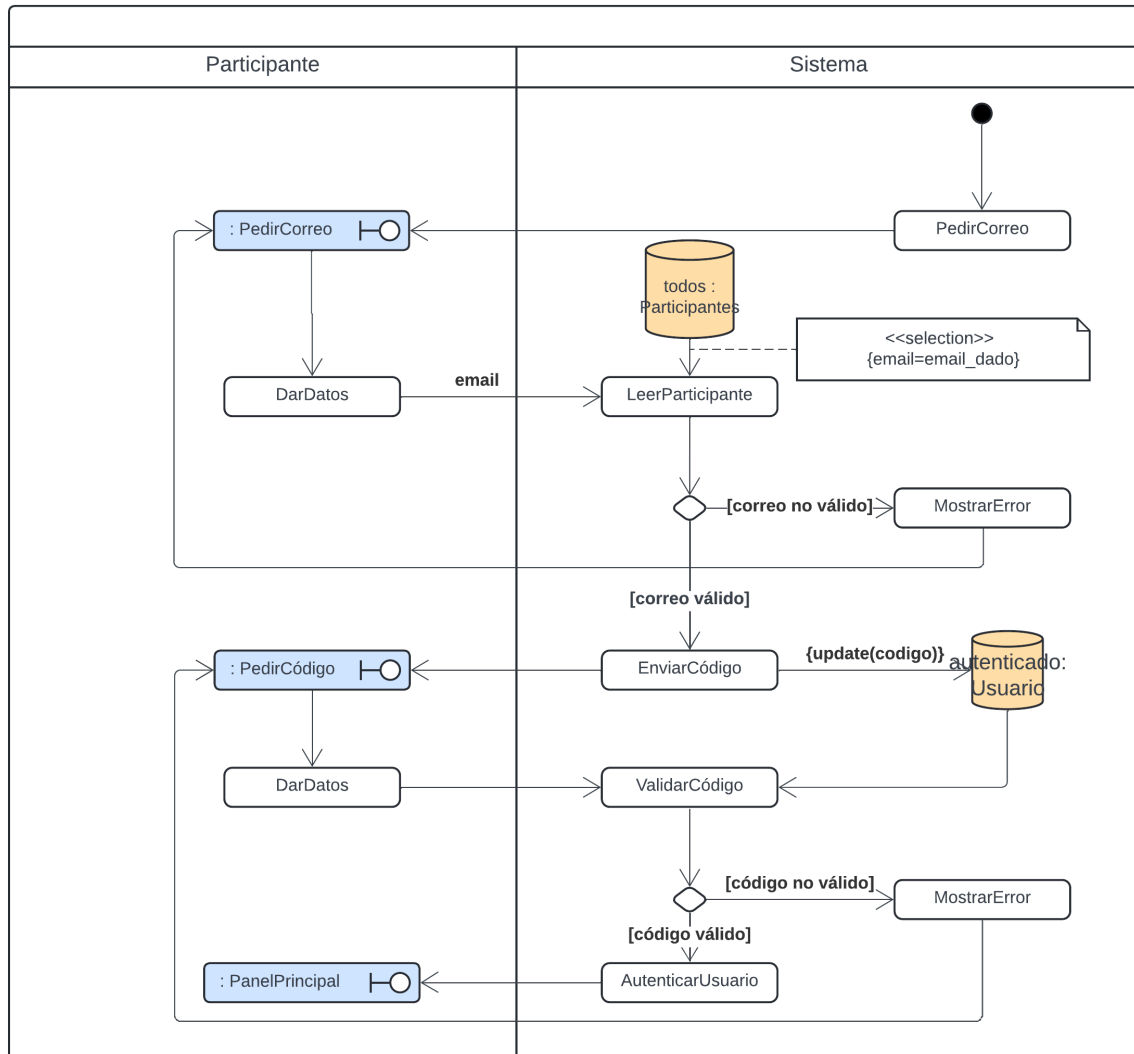


Figura 4.60: Diagrama de actividades del caso de uso 20.

## Diagrama de clases

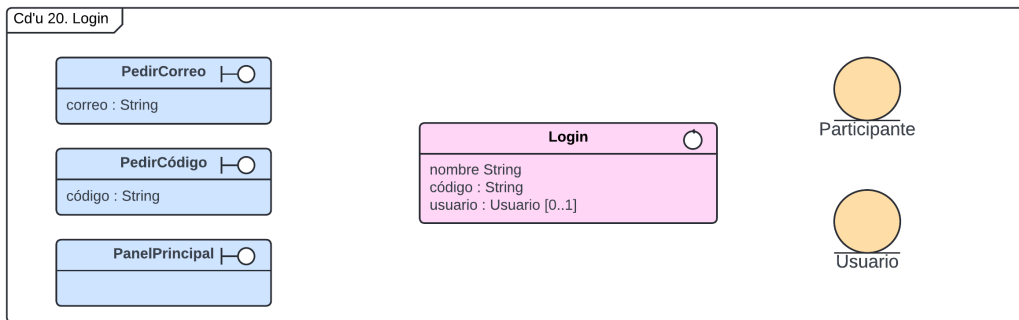


Figura 4.61: Diagrama de clases del caso de uso 20.

## Diagrama de secuencias

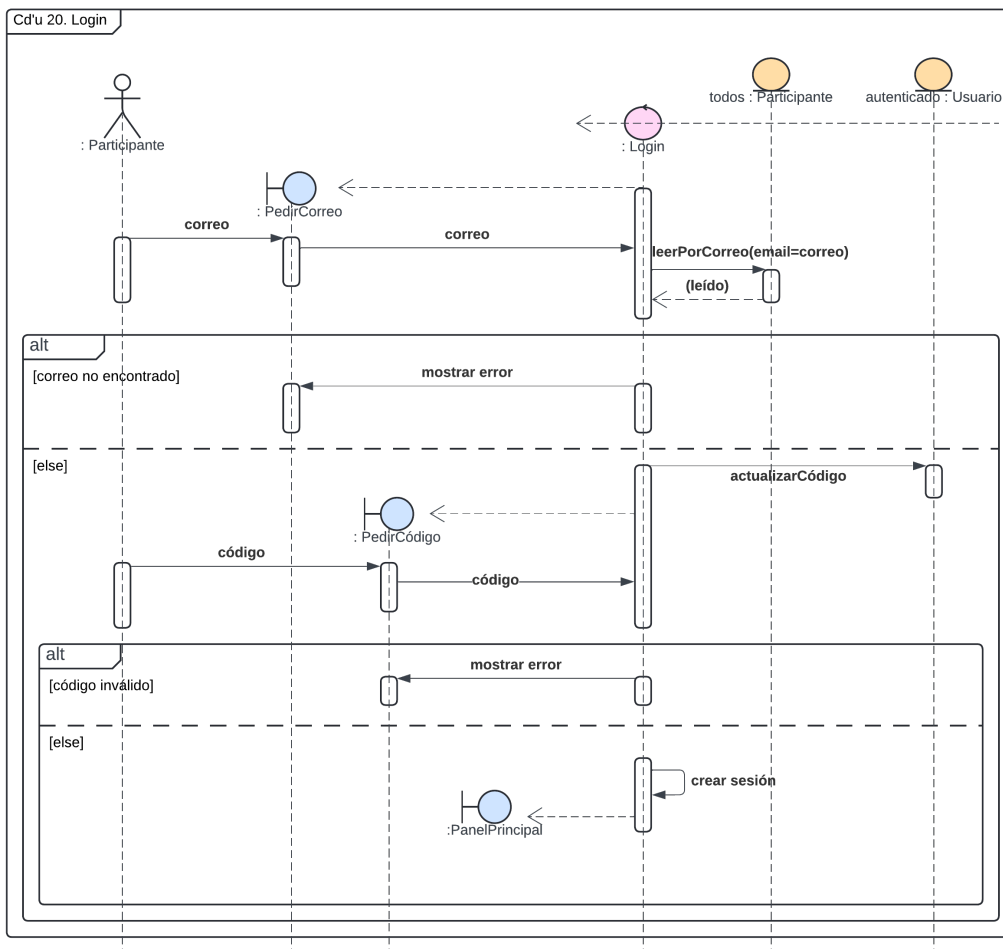


Figura 4.62: Diagrama de secuencias del caso de uso 20.

## 4.2.21. Caso de uso 21. SeleccionarOrganizador

### Diagrama de actividades

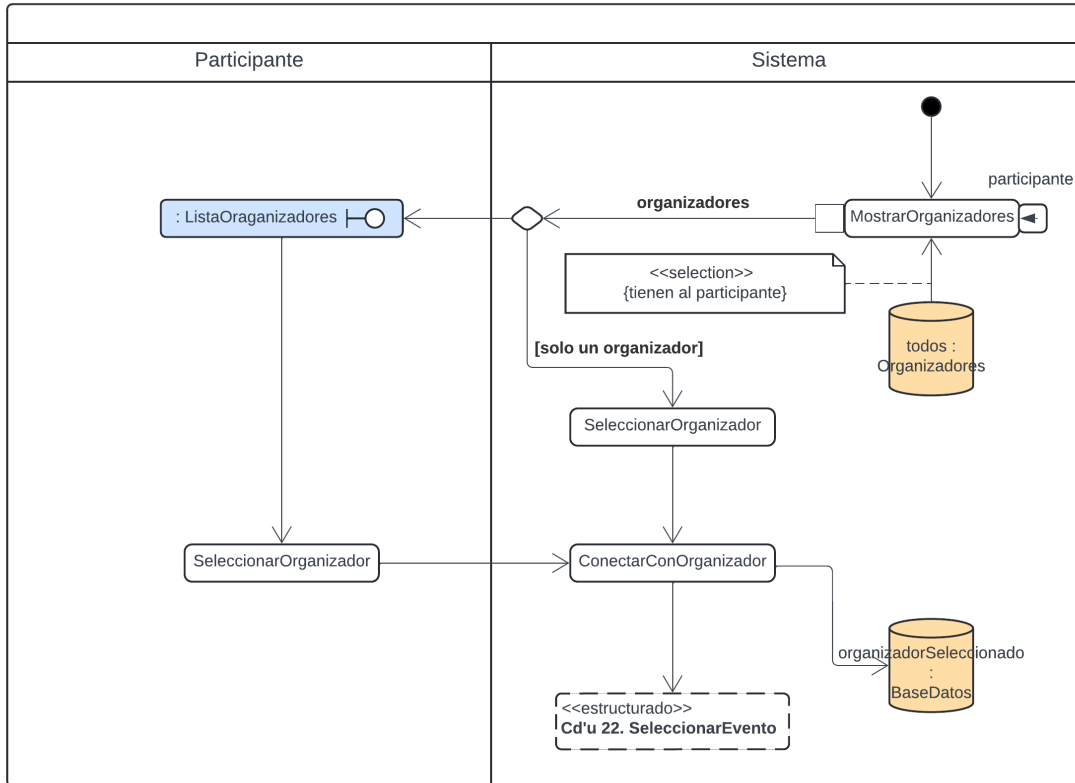


Figura 4.63: Diagrama de actividades del caso de uso 21.

### Diagrama de clases

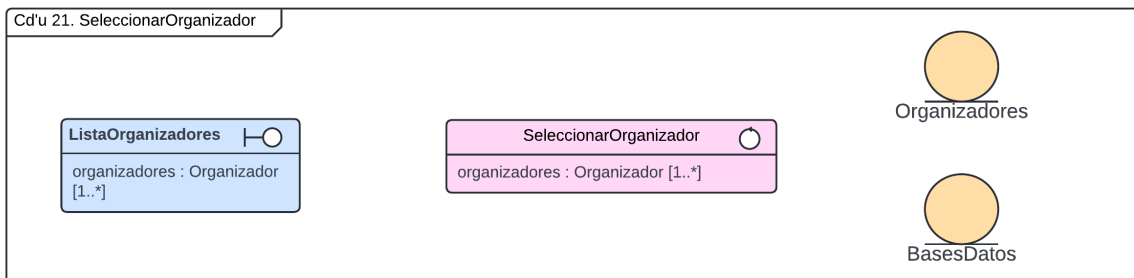


Figura 4.64: Diagrama de clases del caso de uso 21.

# Diagrama de secuencias

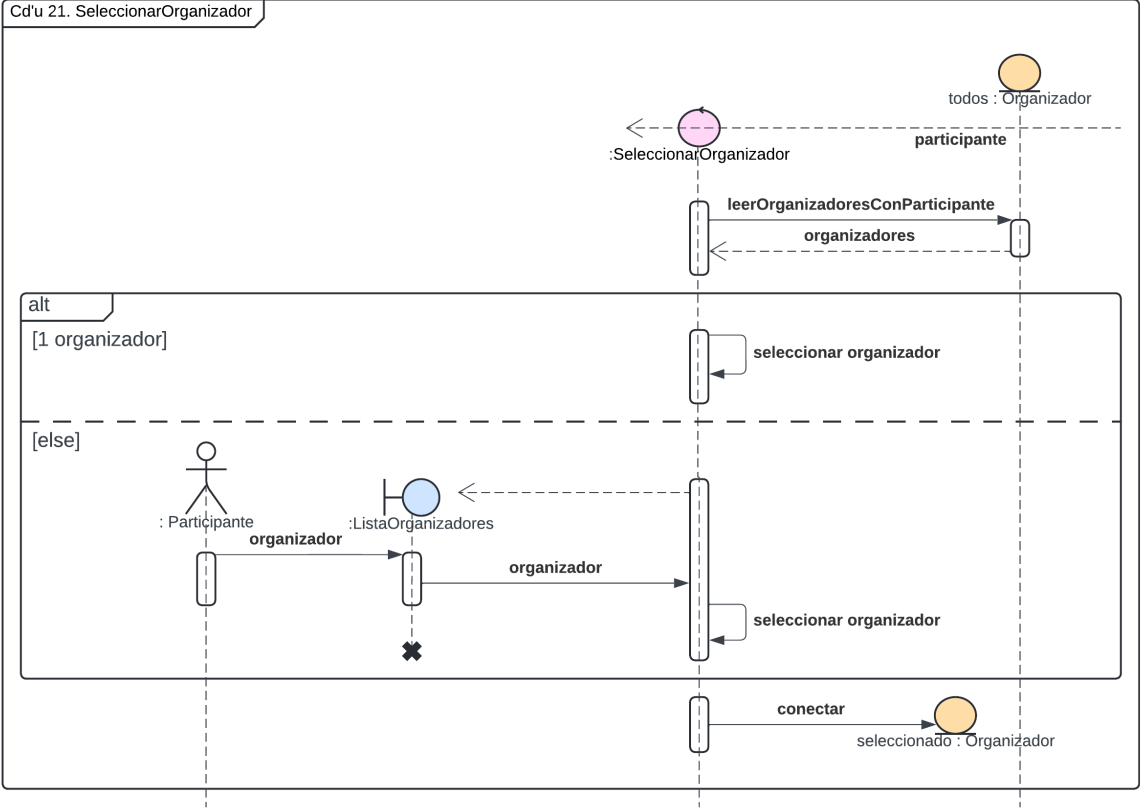


Figura 4.65: Diagrama de secuencias del caso de uso 21.

## 4.2.22. Caso de uso 22. SeleccionarEvento

### Diagrama de actividades

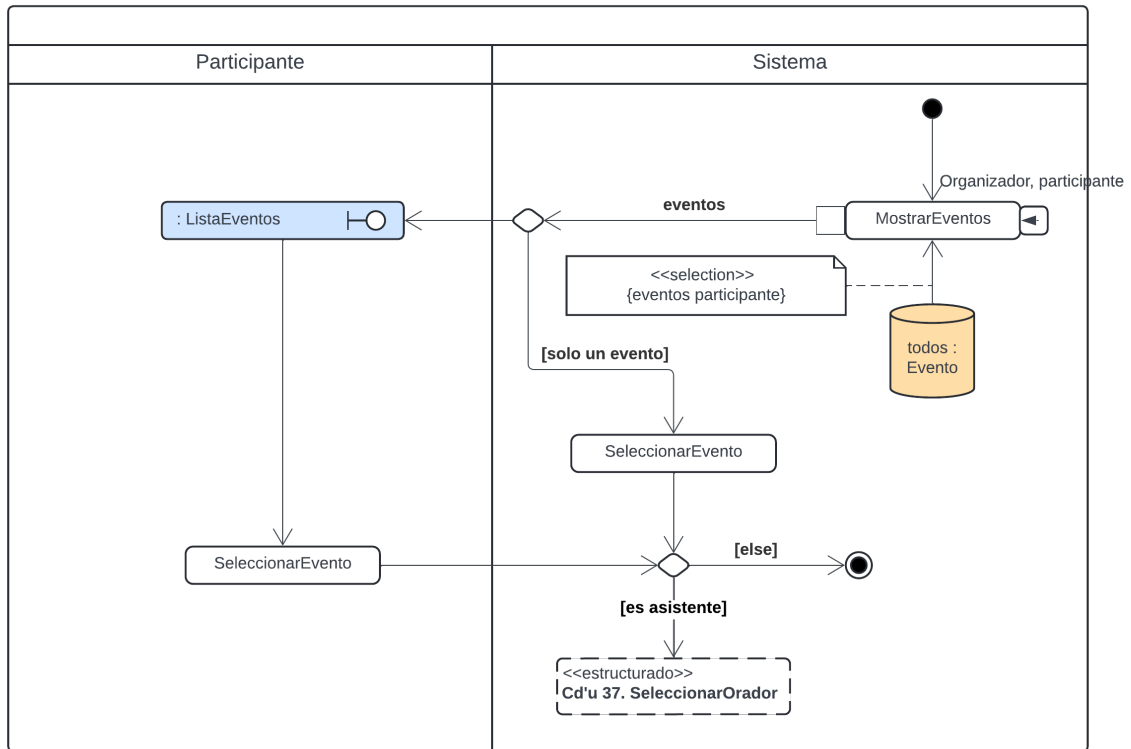


Figura 4.66: Diagrama de actividades del caso de uso 22.

### Diagrama de clases

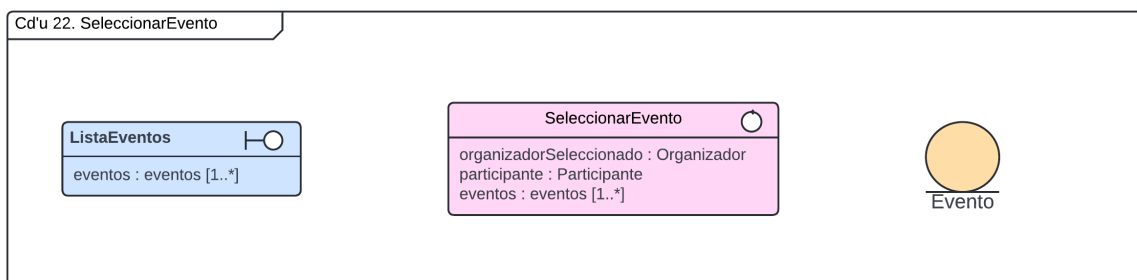


Figura 4.67: Diagrama de clases del caso de uso 22.

# Diagrama de secuencias

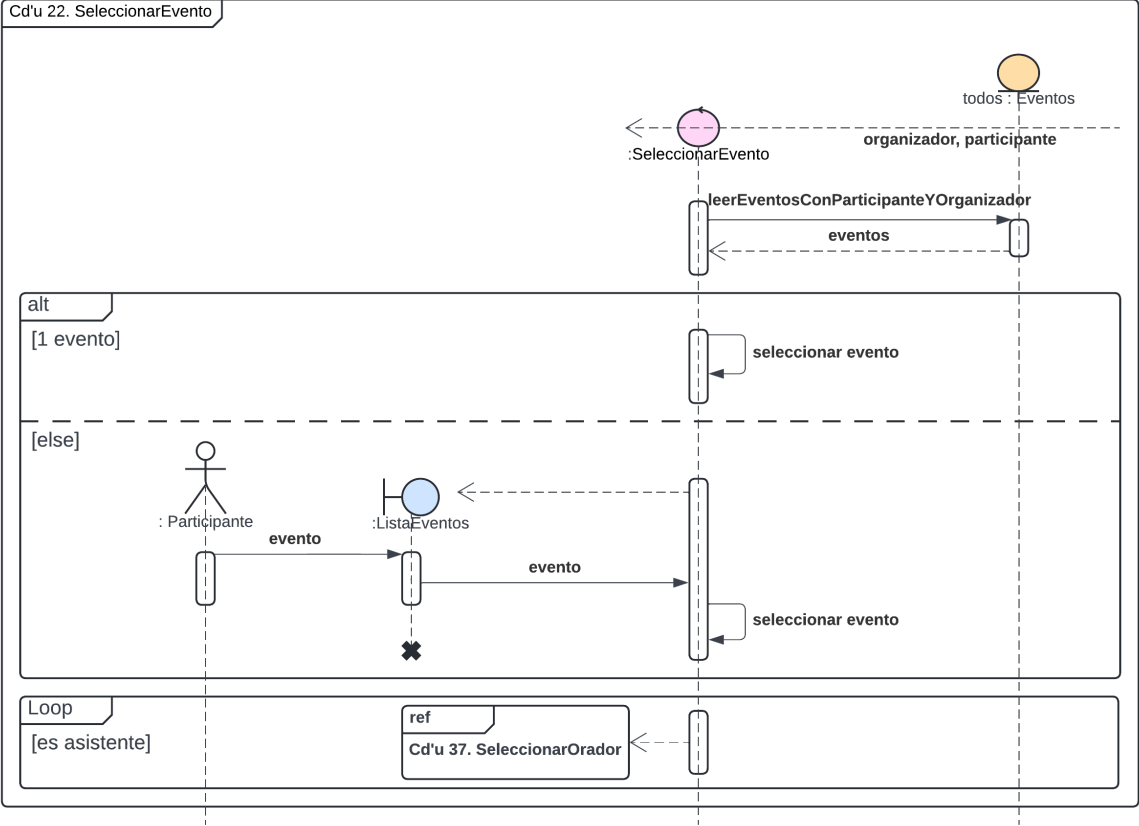


Figura 4.68: Diagrama de secuencias del caso de uso 22.

### 4.2.23. Caso de uso 23. VerDocumentos

#### Diagrama de actividades

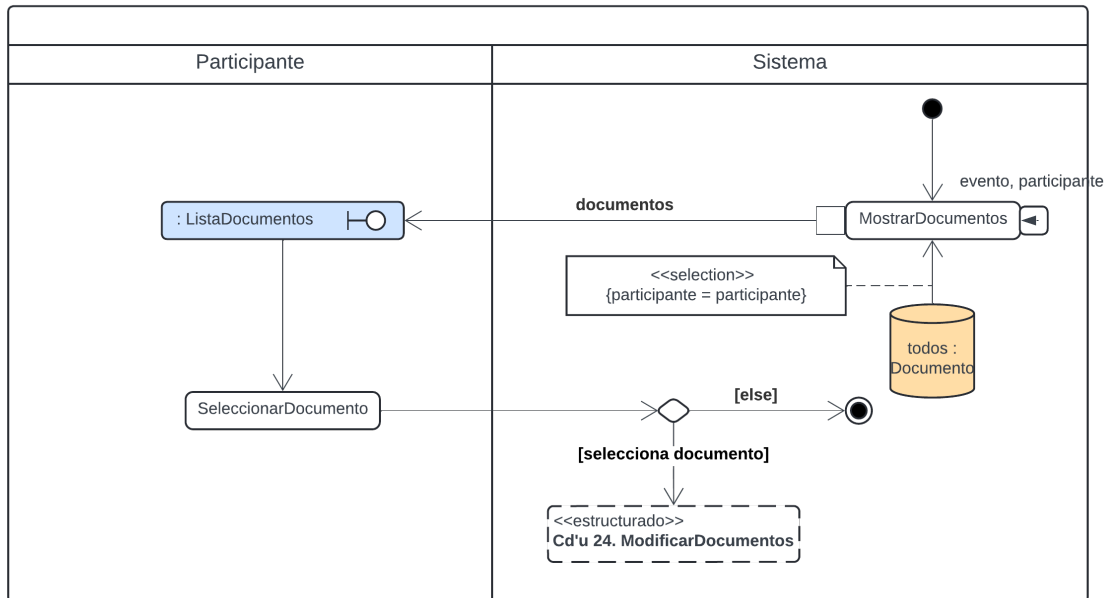


Figura 4.69: Diagrama de actividades del caso de uso 23.

#### Diagrama de clases

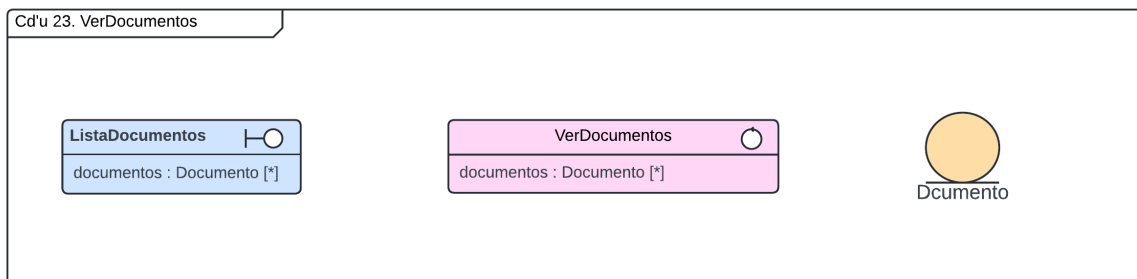


Figura 4.70: Diagrama de clases del caso de uso 23.

# Diagrama de secuencias

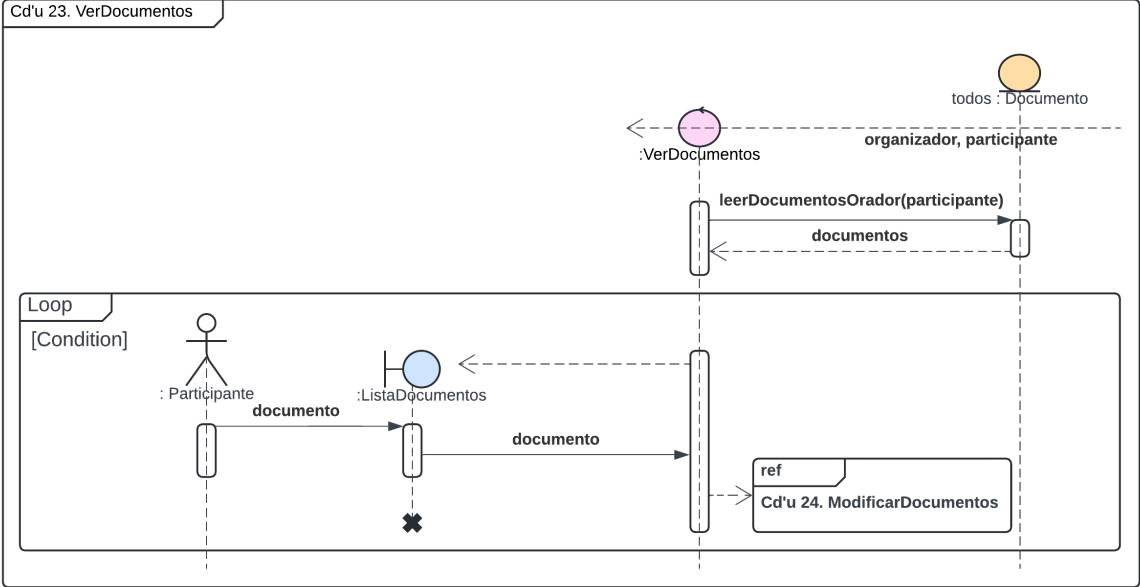


Figura 4.71: Diagrama de secuencias del caso de uso 23.

## 4.2.24. Caso de uso 24. ModificarDocumentos

### Diagrama de actividades

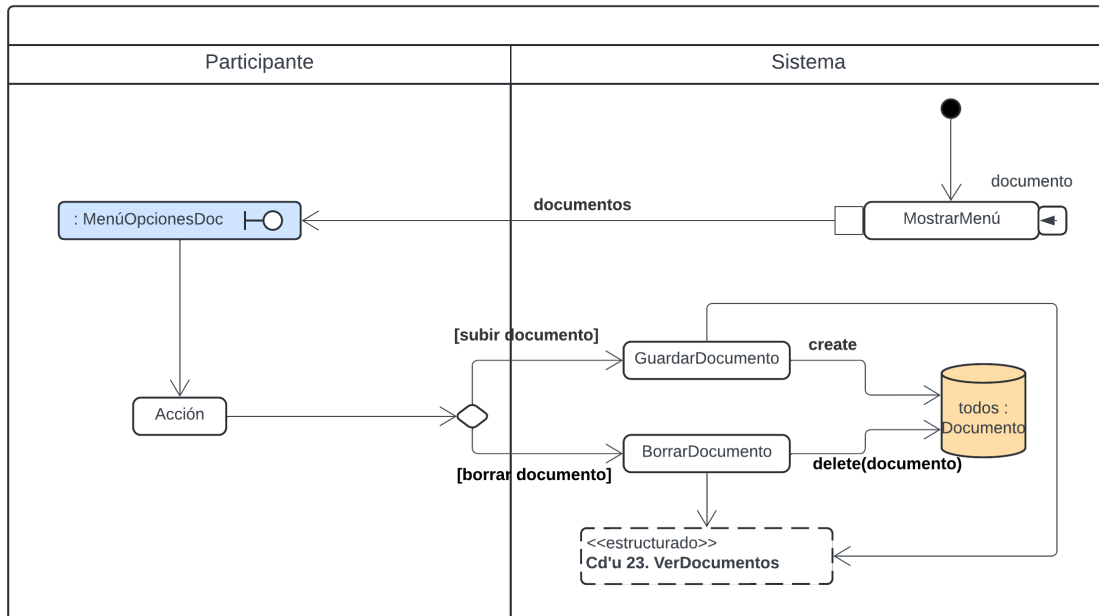


Figura 4.72: Diagrama de actividades del caso de uso 24.

### Diagrama de clases

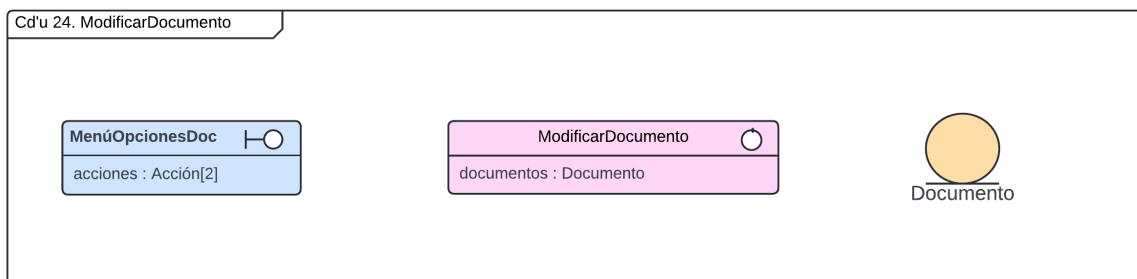


Figura 4.73: Diagrama de clases del caso de uso 24.



## 4.2.25. Caso de uso 25. VerParticipación

### Diagrama de actividades

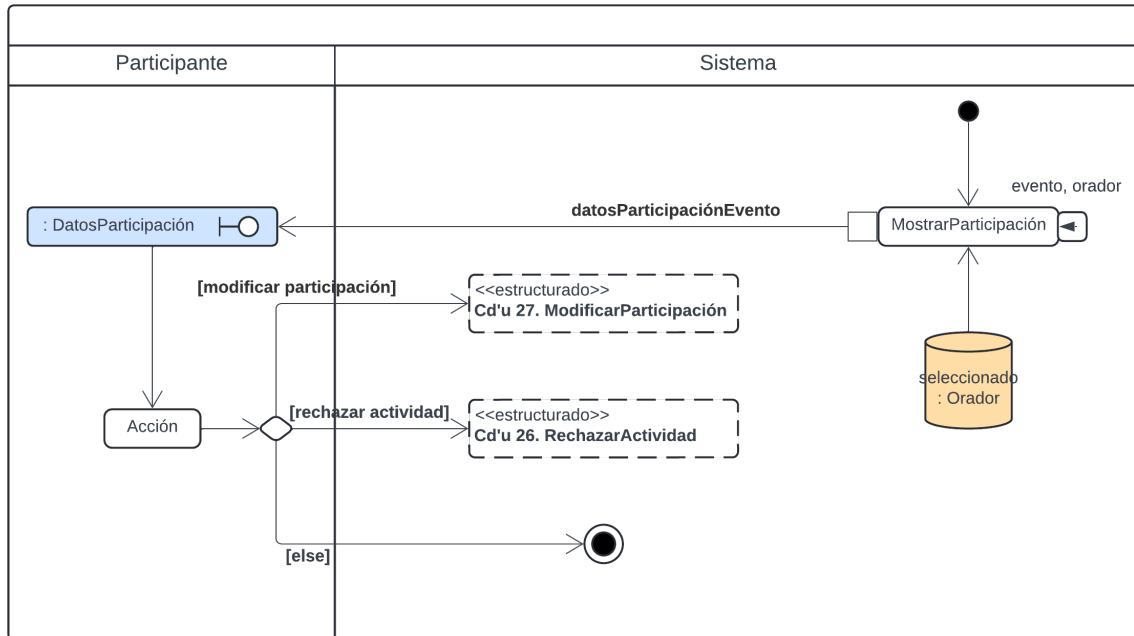


Figura 4.75: Diagrama de actividades del caso de uso 25.

### Diagrama de clases

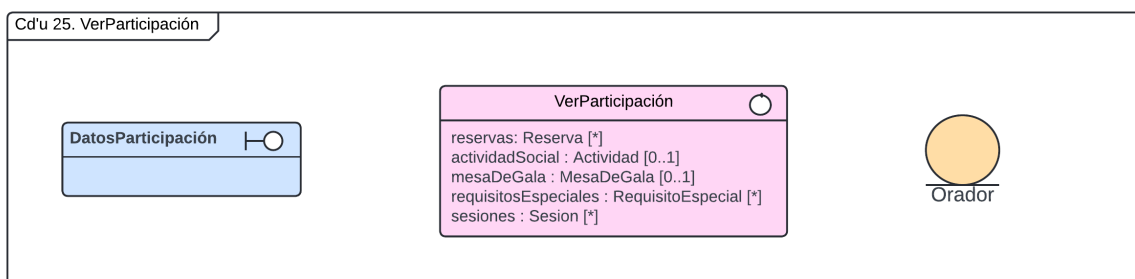


Figura 4.76: Diagrama de clases del caso de uso 25.

# Diagrama de secuencias

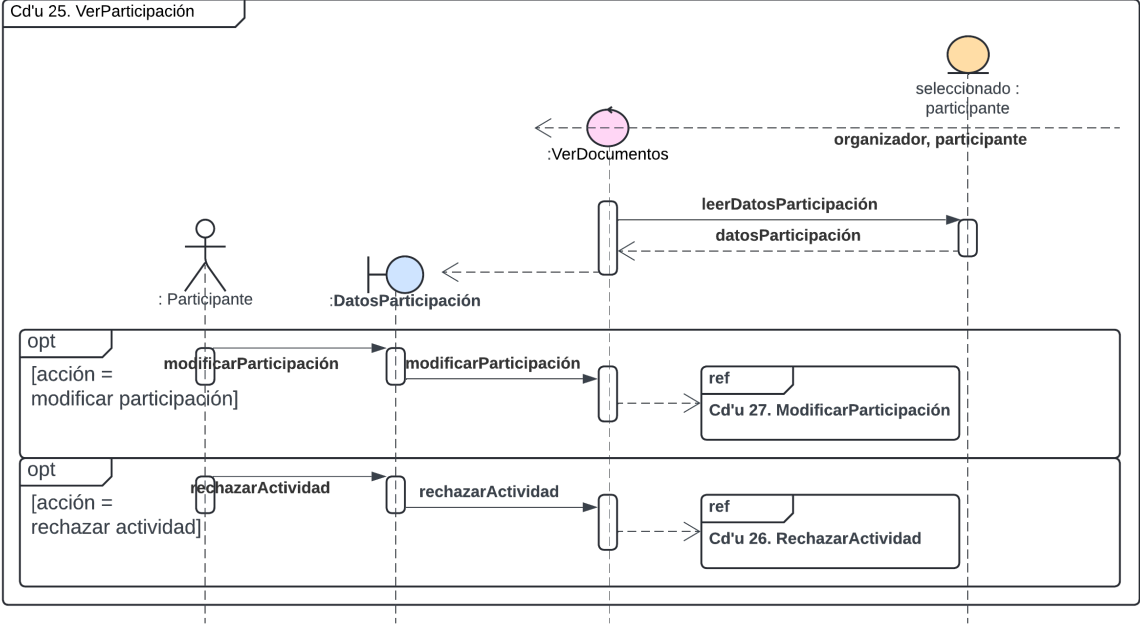


Figura 4.77: Diagrama de secuencias del caso de uso 25.

## 4.2.26. Caso de uso 26. RechazarActividad

### Diagrama de actividades

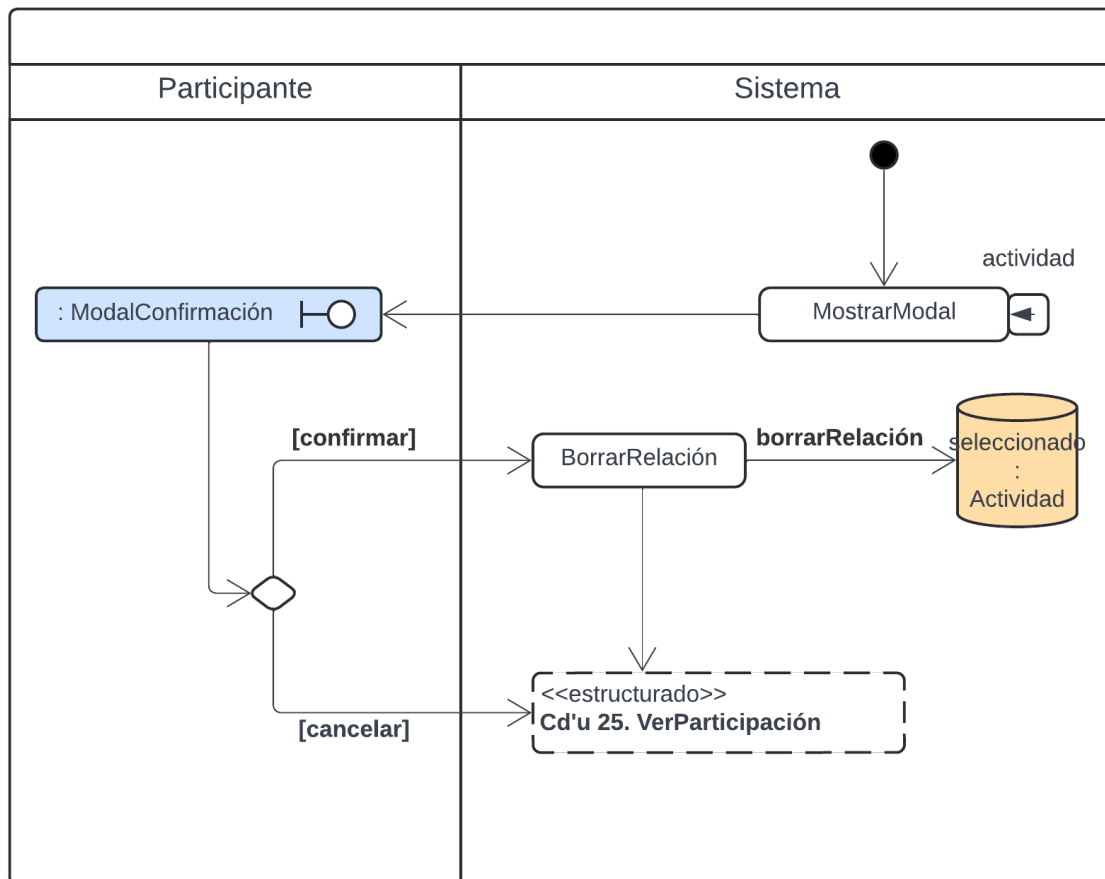


Figura 4.78: Diagrama de actividades del caso de uso 26.

### Diagrama de clases

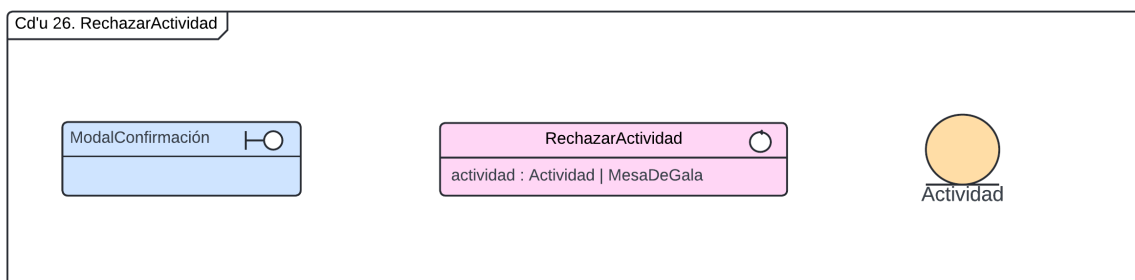


Figura 4.79: Diagrama de clases del caso de uso 26.

Diagrama de secuencias

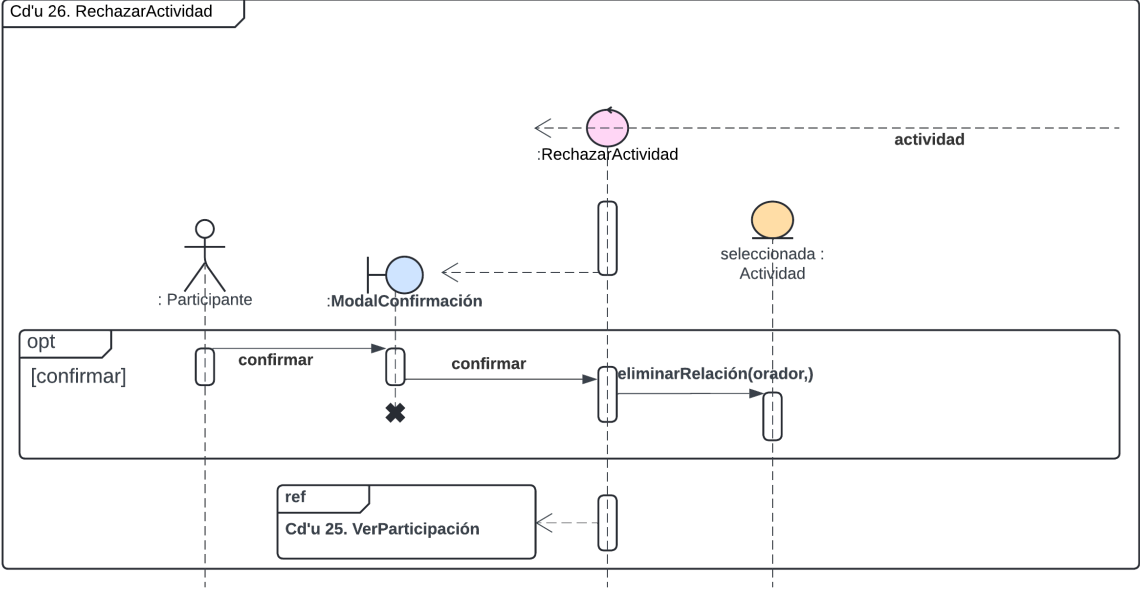


Figura 4.80: Diagrama de secuencias del caso de uso 26.

## 4.2.27. Caso de uso 27. ModificarParticipación

### Diagrama de actividades

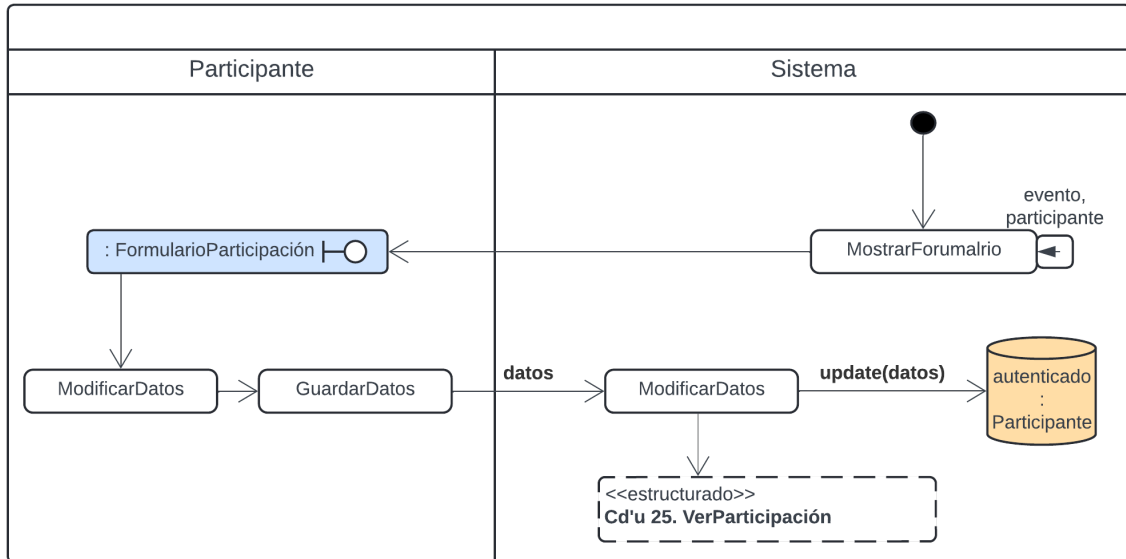


Figura 4.81: Diagrama de actividades del caso de uso 27.

### Diagrama de clases

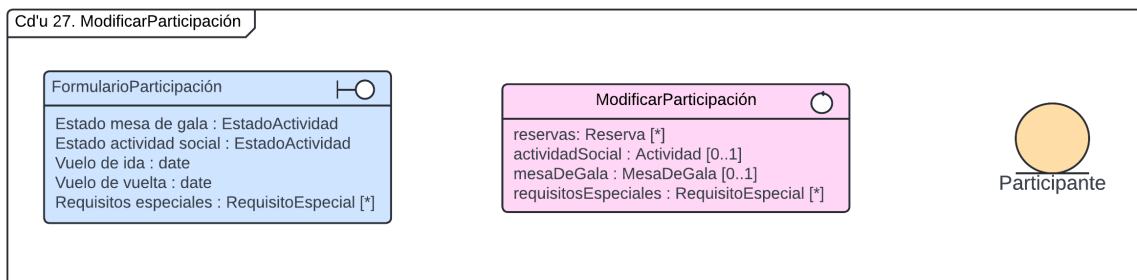


Figura 4.82: Diagrama de clases del caso de uso 27.

Diagrama de secuencias

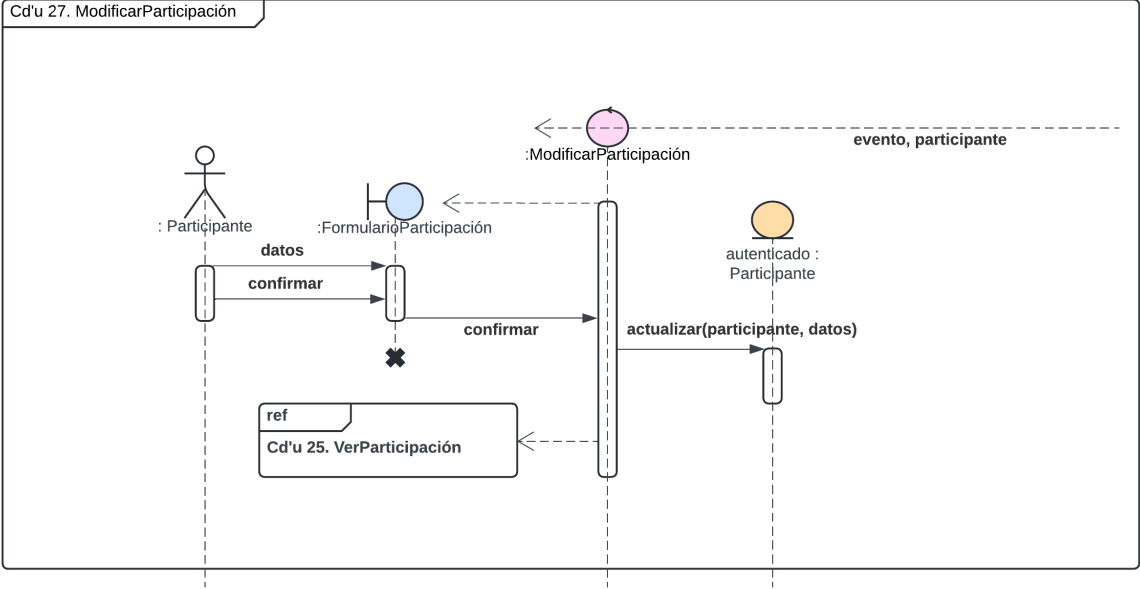


Figura 4.83: Diagrama de secuencias del caso de uso 27.

## 4.2.28. Caso de uso 28. VerSesiones

### Diagrama de actividades

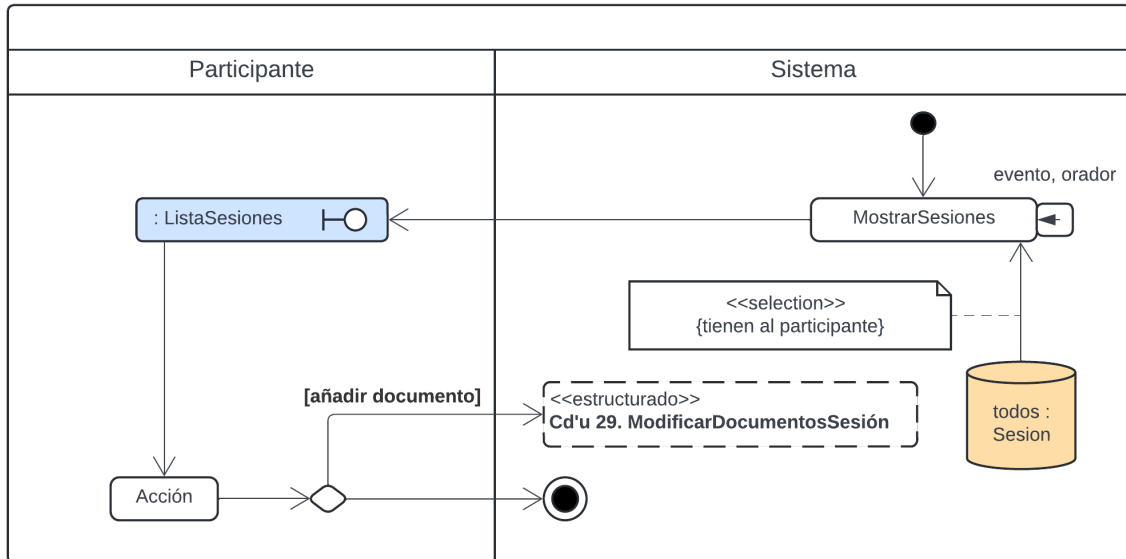


Figura 4.84: Diagrama de actividades del caso de uso 28.

### Diagrama de clases

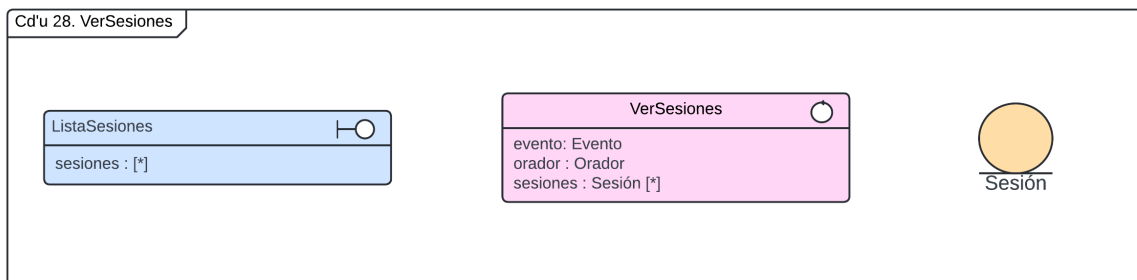


Figura 4.85: Diagrama de clases del caso de uso 28.

Diagrama de secuencias

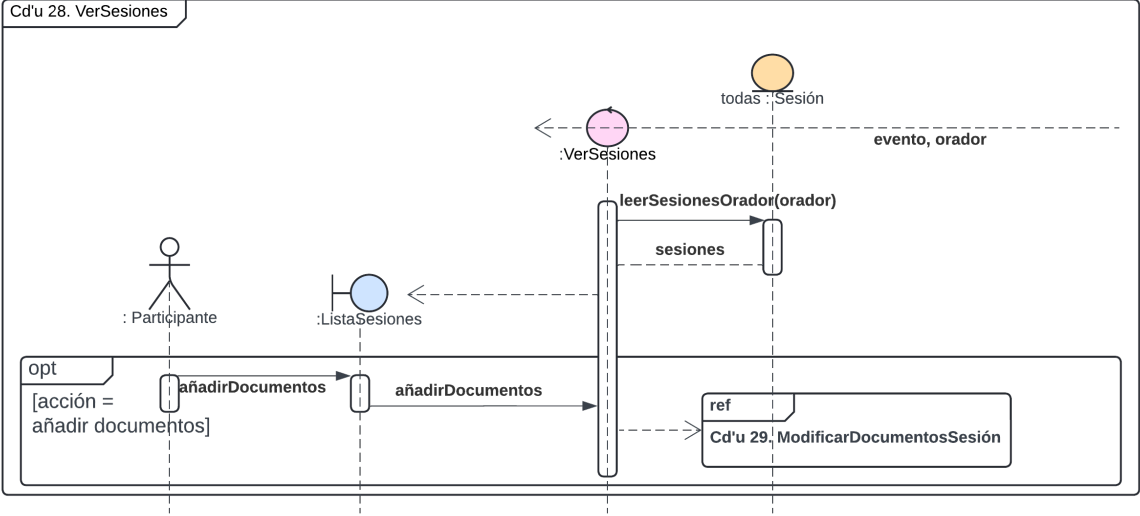


Figura 4.86: Diagrama de secuencias del caso de uso 28.

## 4.2.29. Caso de uso 29. ModificarDocumentosSesión

### Diagrama de actividades

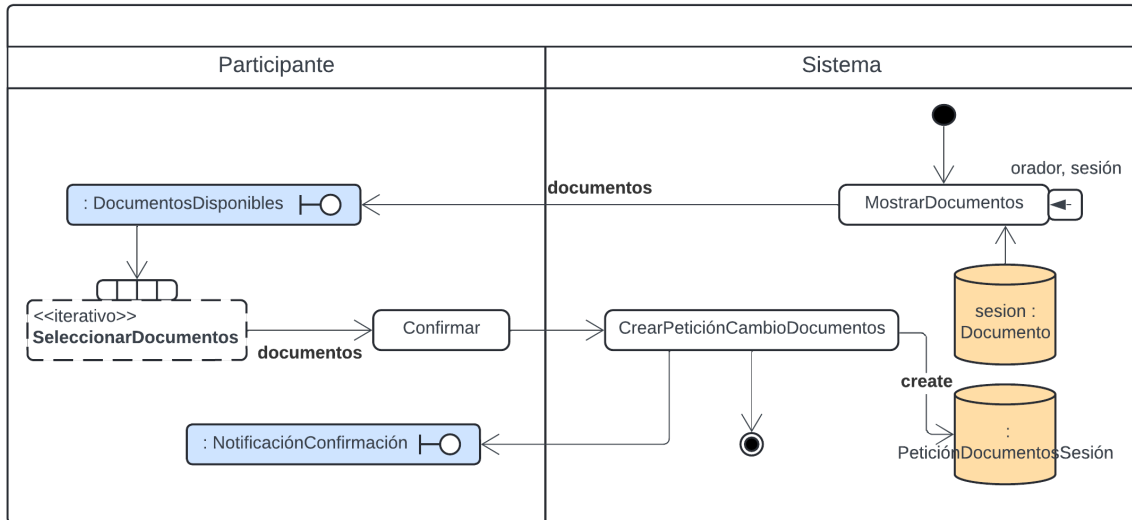


Figura 4.87: Diagrama de actividades del caso de uso 29.

### Diagrama de clases

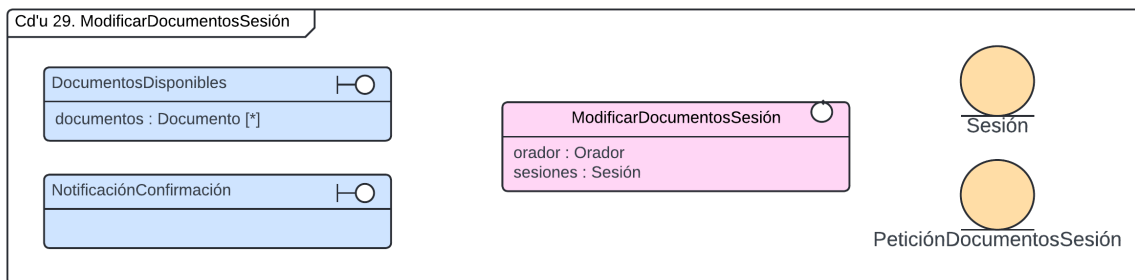


Figura 4.88: Diagrama de clases del caso de uso 29.

# Diagrama de secuencias

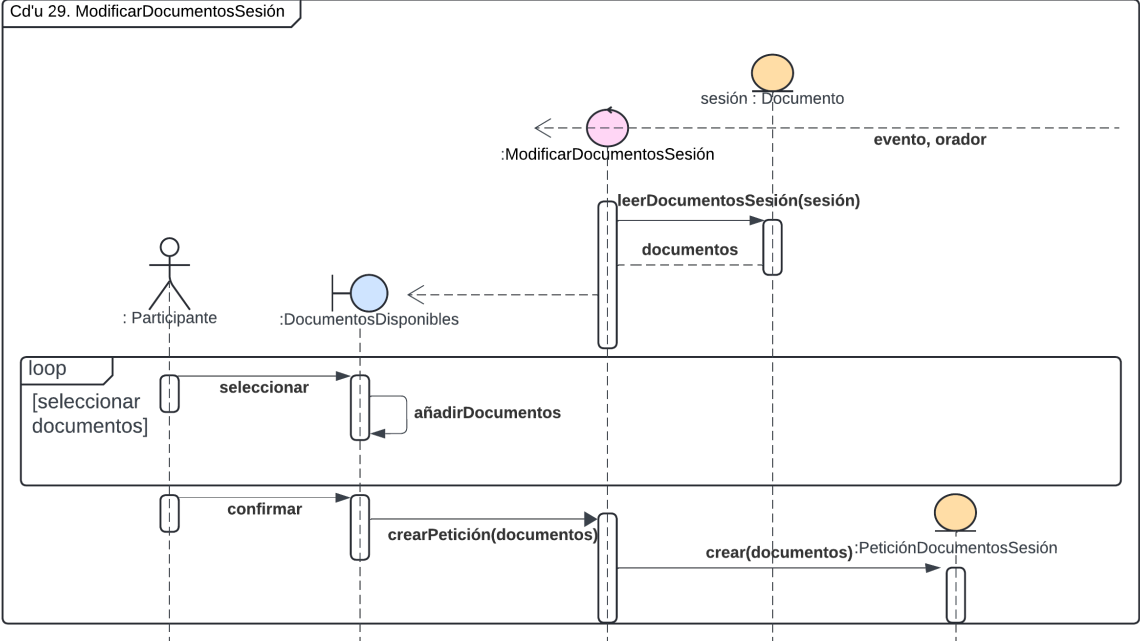


Figura 4.89: Diagrama de secuencias del caso de uso 29.

## 4.2.30. Caso de uso 30. VerArtículosOrador

### Diagrama de actividades

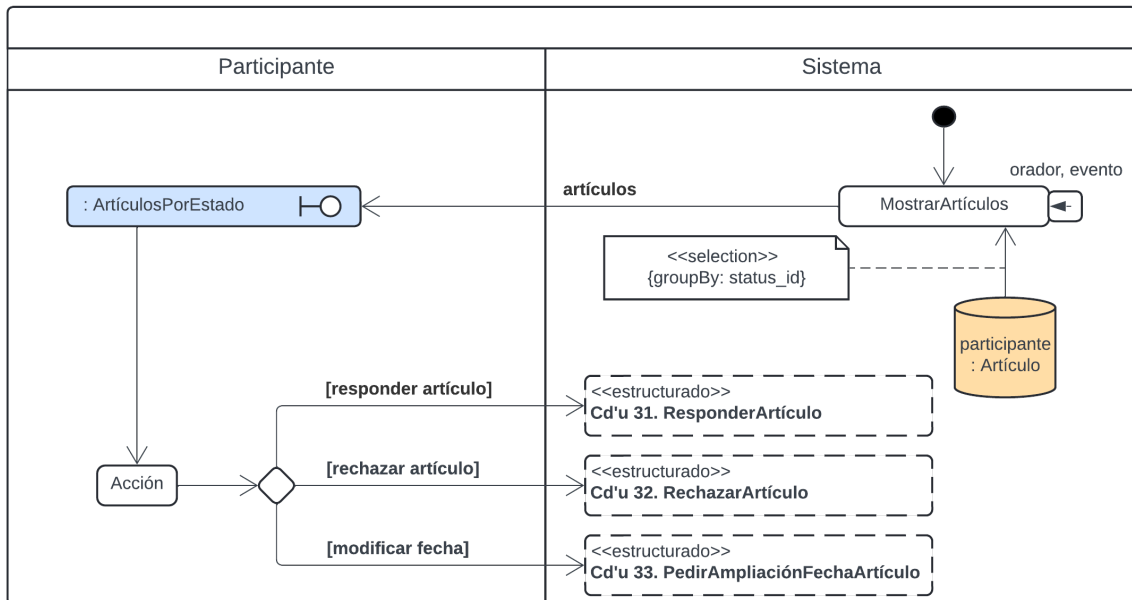


Figura 4.90: Diagrama de actividades del caso de uso 30.

### Diagrama de clases

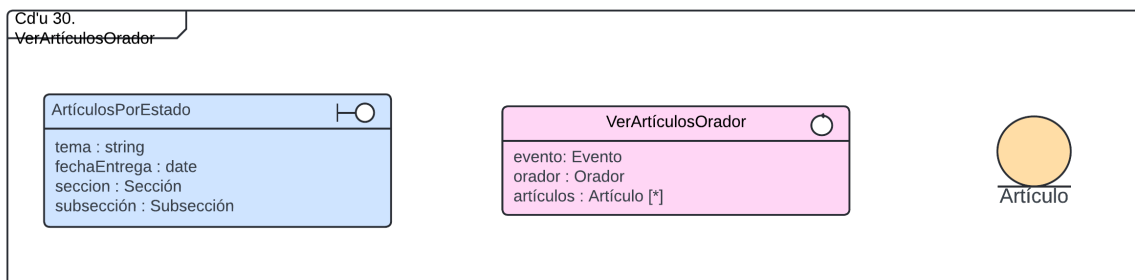


Figura 4.91: Diagrama de clases del caso de uso 30.

# Diagrama de secuencias

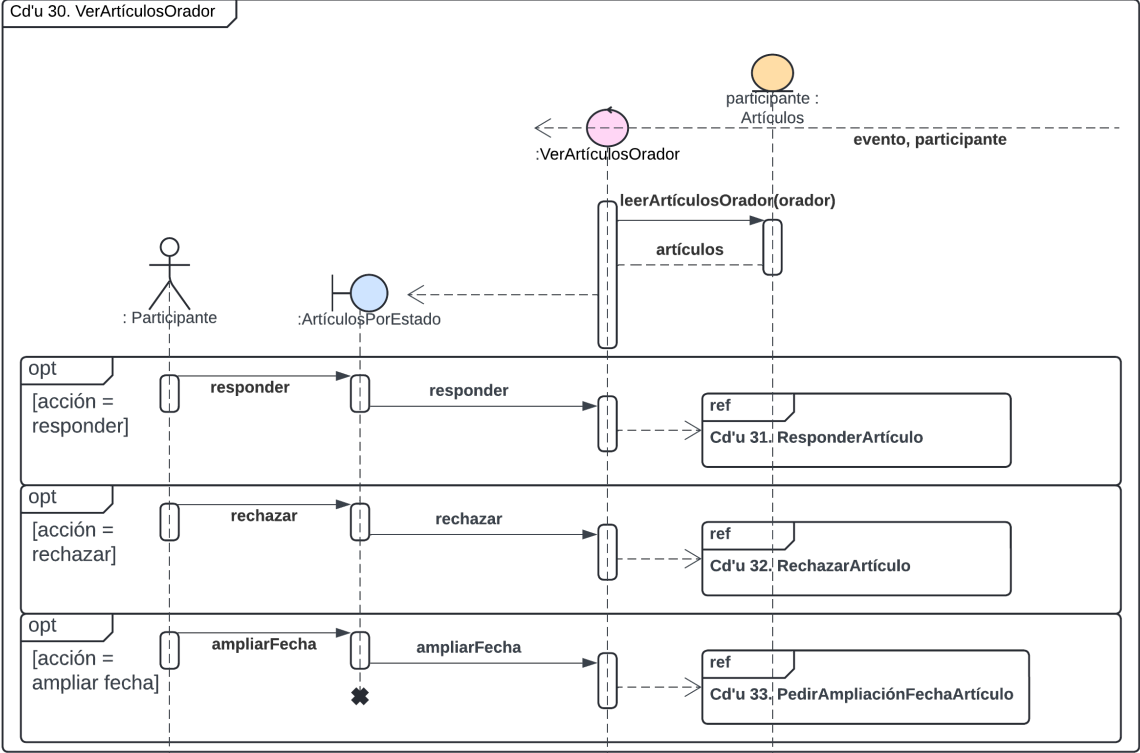


Figura 4.92: Diagrama de secuencias del caso de uso 30.

## 4.2.31. Caso de uso 31. ResponderArtículo

### Diagrama de actividades

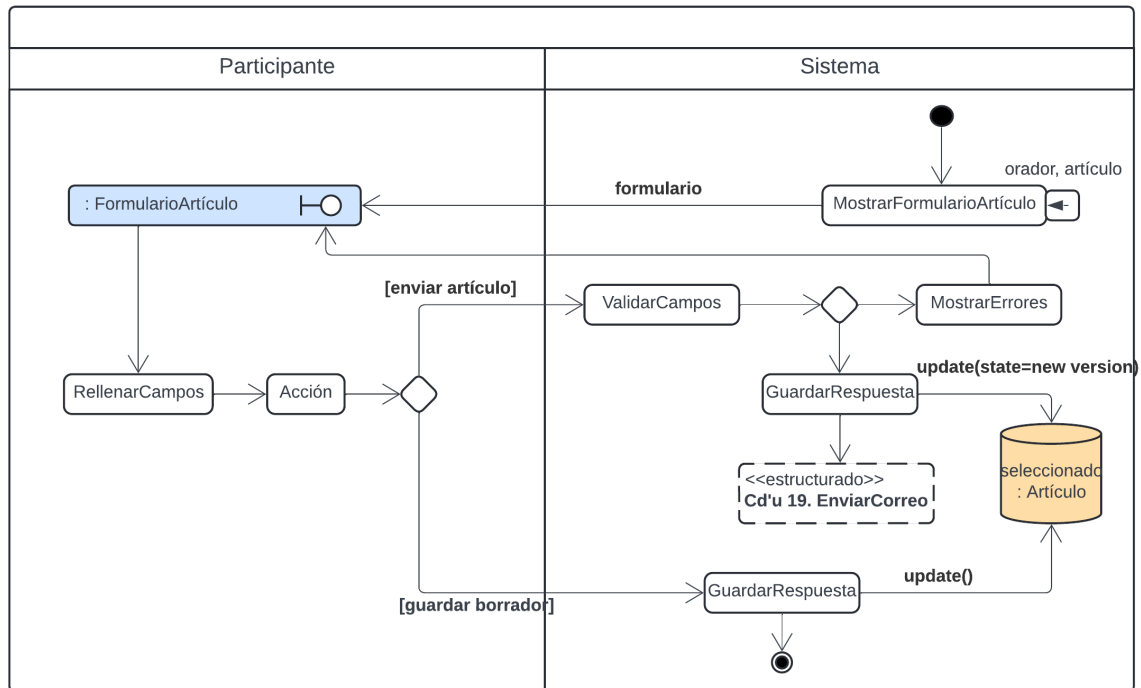


Figura 4.93: Diagrama de actividades del caso de uso 31.

### Diagrama de clases

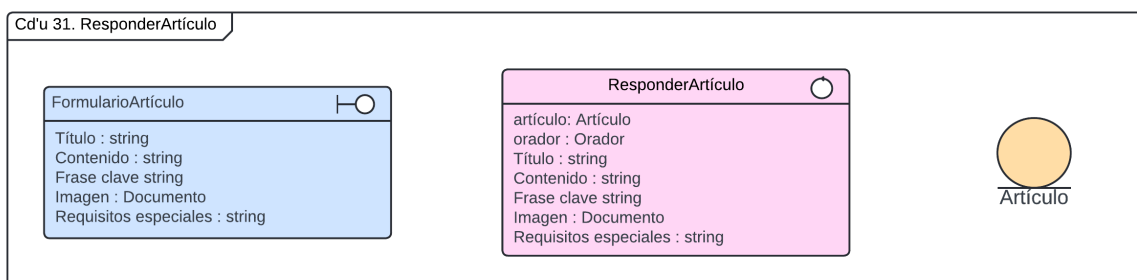


Figura 4.94: Diagrama de clases del caso de uso 31.

# Diagrama de secuencias

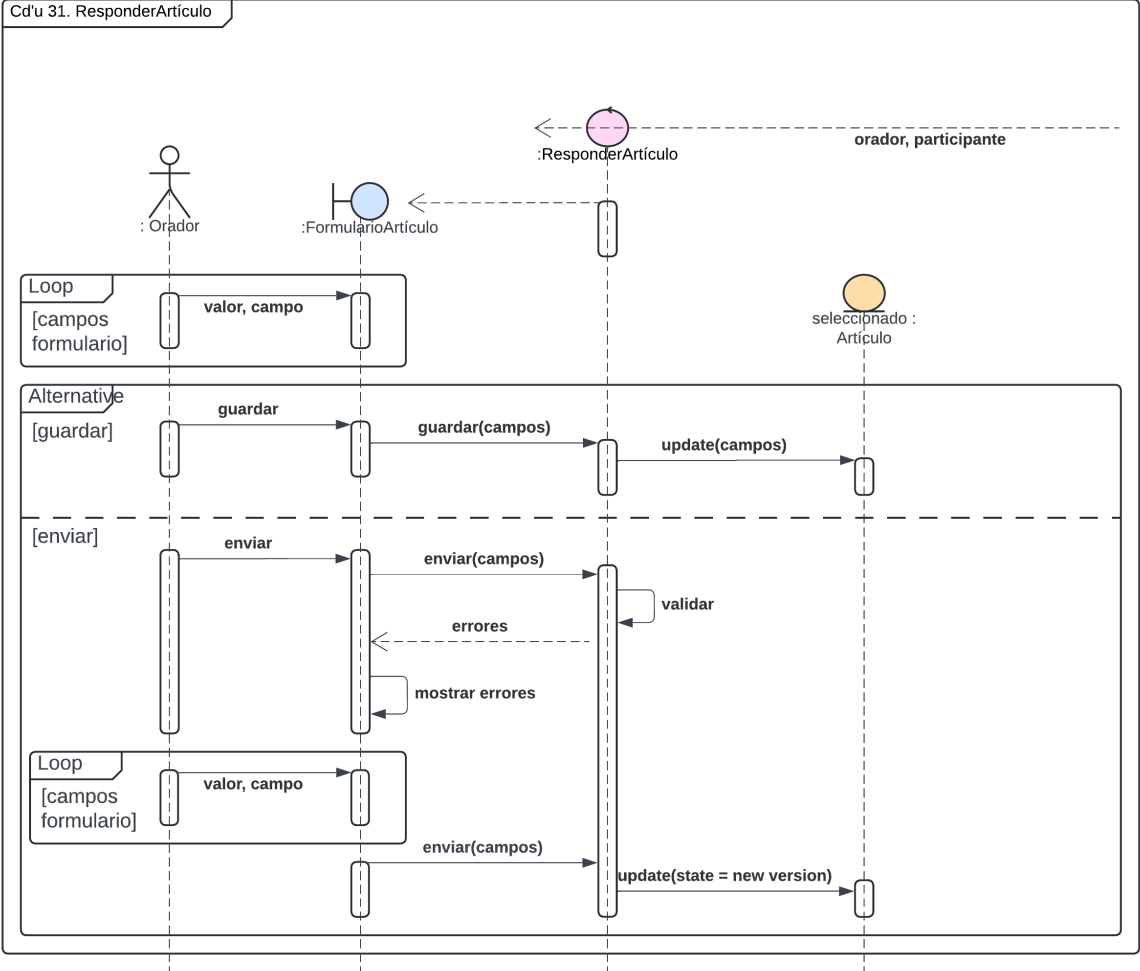


Figura 4.95: Diagrama de secuencias del caso de uso 31.

## 4.2.32. Caso de uso 32. RechazarArtículo

### Diagrama de actividades

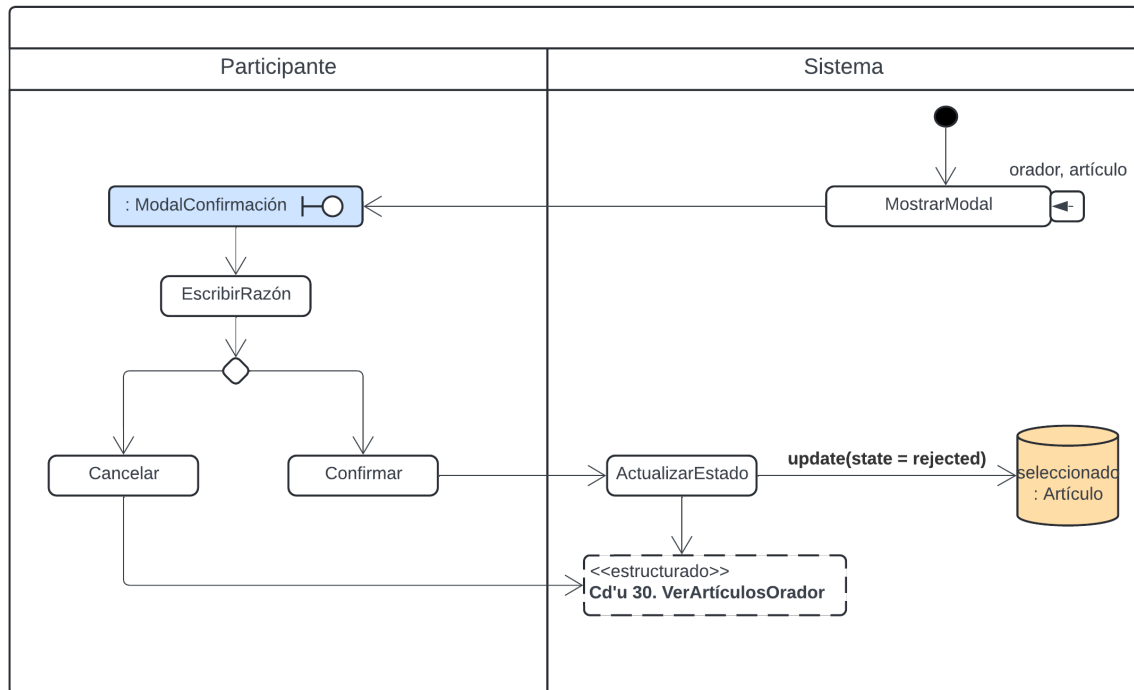


Figura 4.96: Diagrama de actividades del caso de uso 32.

### Diagrama de clases

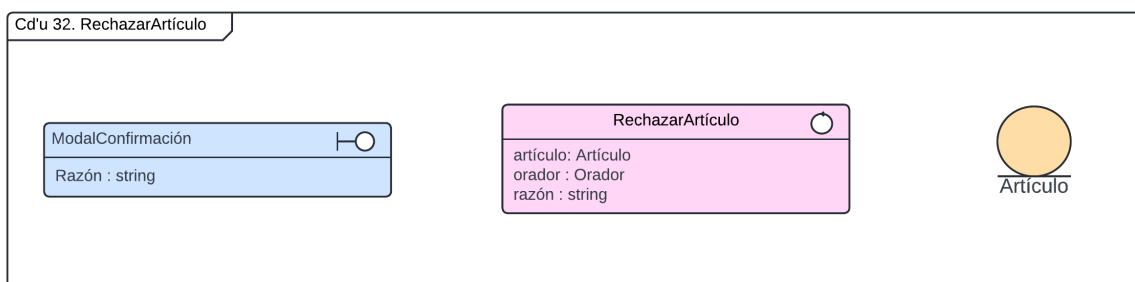


Figura 4.97: Diagrama de clases del caso de uso 32.

# Diagrama de secuencias

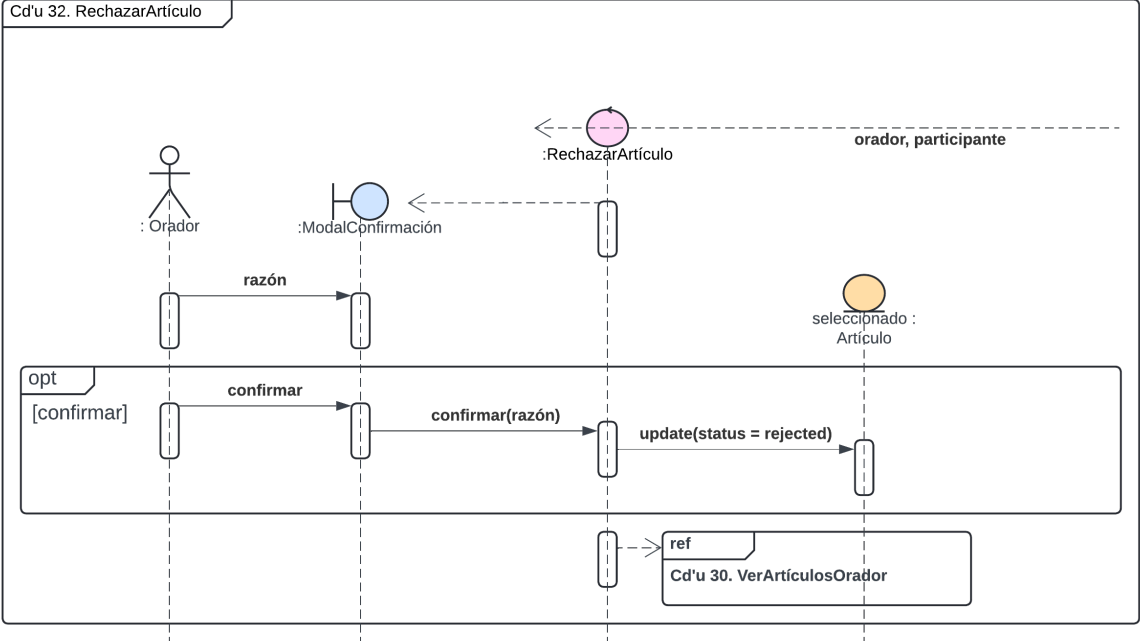


Figura 4.98: Diagrama de secuencias del caso de uso 32.

### 4.2.33. Caso de uso 33. PedirAmpliaciónFechaArtículo

#### Diagrama de actividades

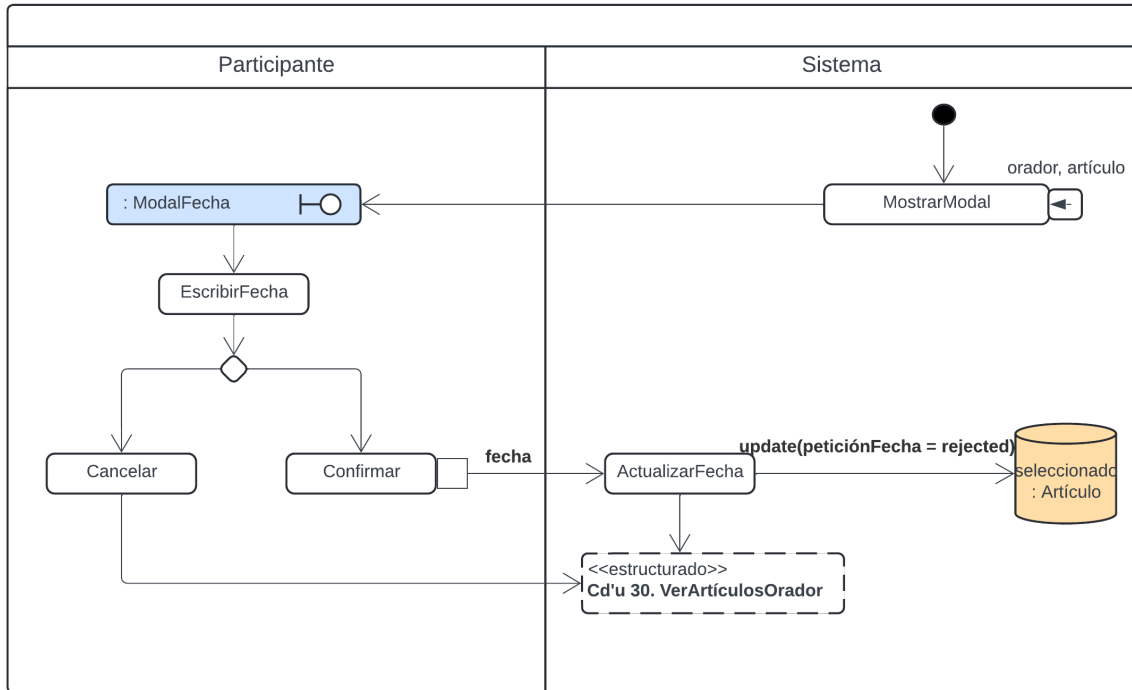


Figura 4.99: Diagrama de actividades del caso de uso 33.

#### Diagrama de clases

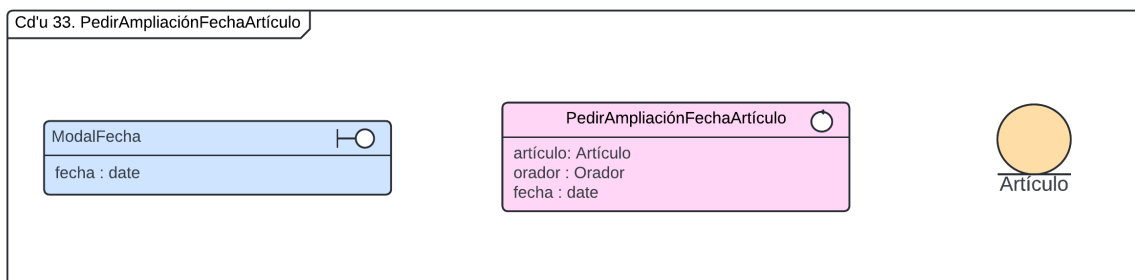


Figura 4.100: Diagrama de clases del caso de uso 33.

# Diagrama de secuencias

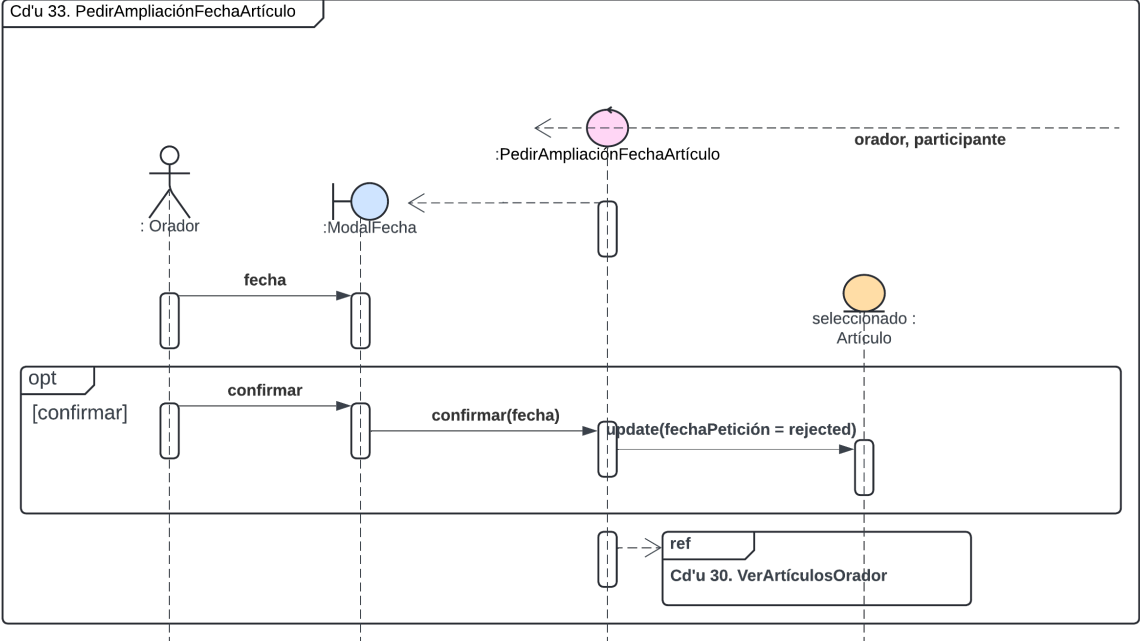


Figura 4.101: Diagrama de secuencias del caso de uso 33.

## 4.2.34. Caso de uso 34. VerInformaciónPersonal

### Diagrama de actividades

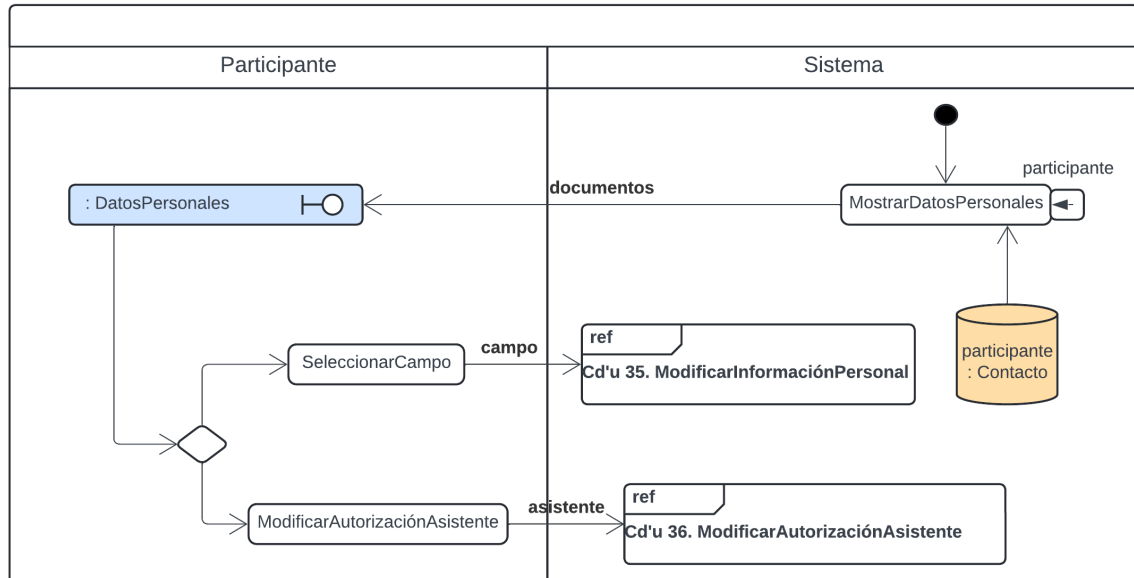


Figura 4.102: Diagrama de actividades del caso de uso 34.

### Diagrama de clases

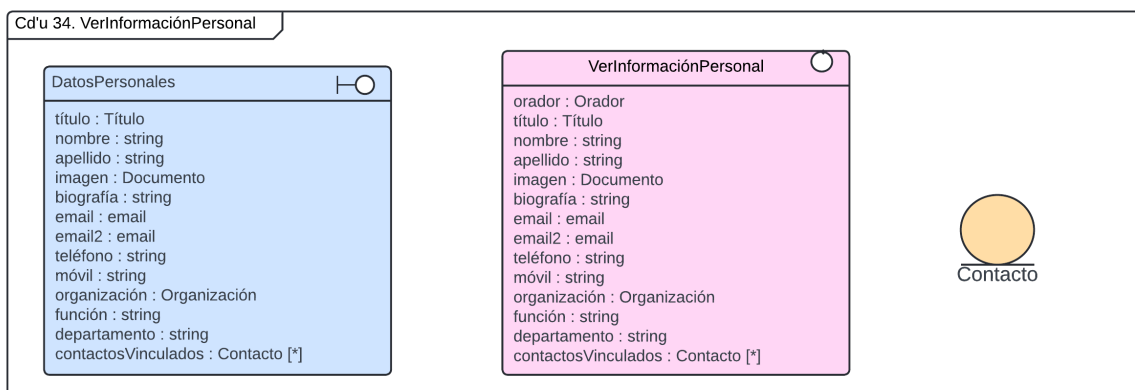


Figura 4.103: Diagrama de clases del caso de uso 34.

# Diagrama de secuencias

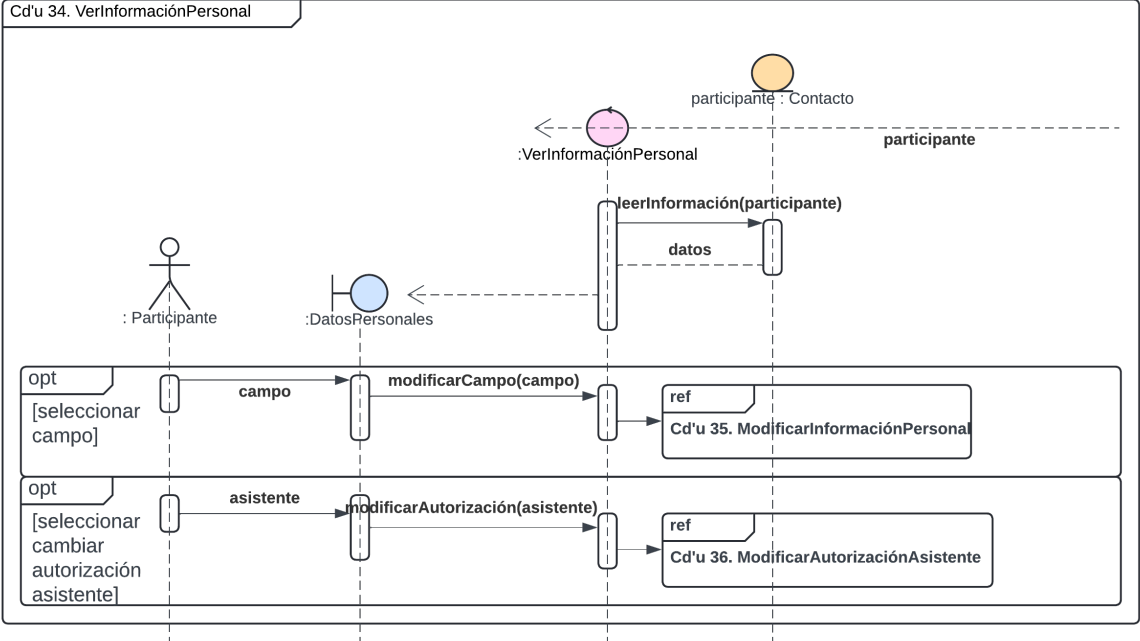


Figura 4.104: Diagrama de secuencias del caso de uso 34.

## 4.2.35. Caso de uso 35. ModificarInformaciónPersonal

### Diagrama de actividades

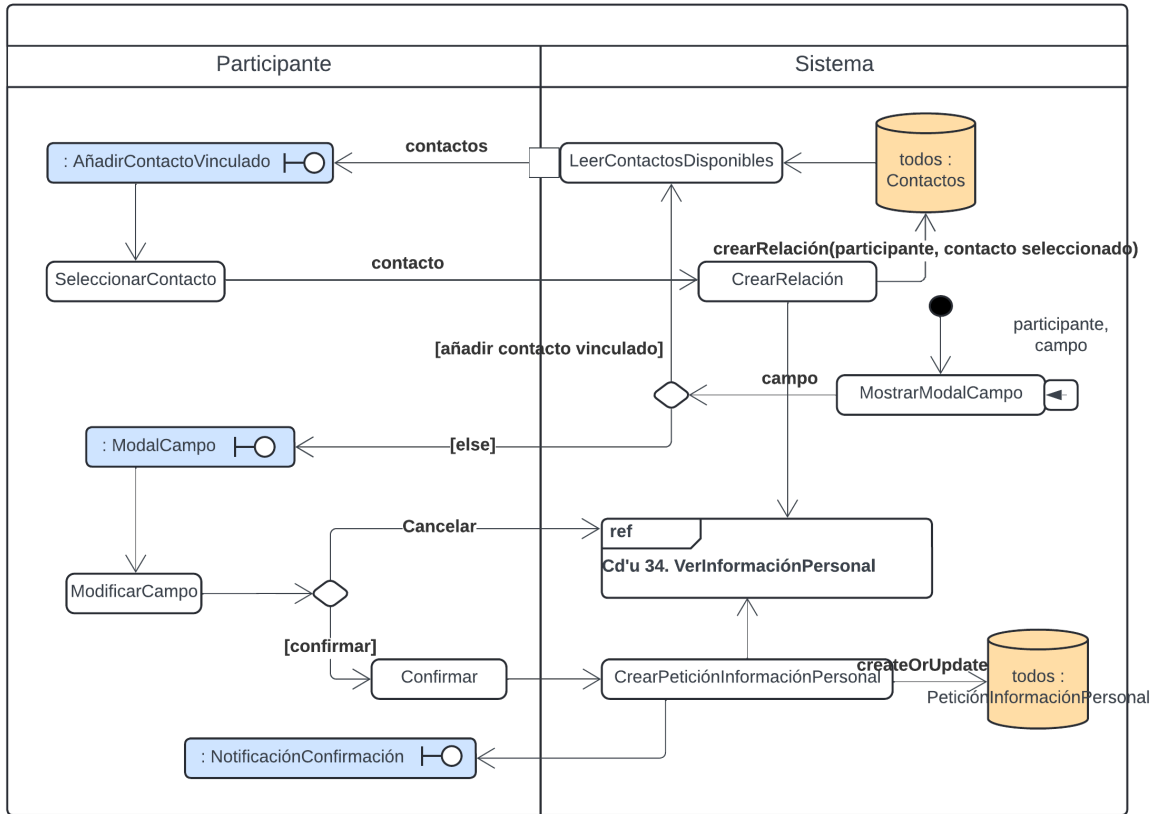


Figura 4.105: Diagrama de actividades del caso de uso 35.

### Diagrama de clases

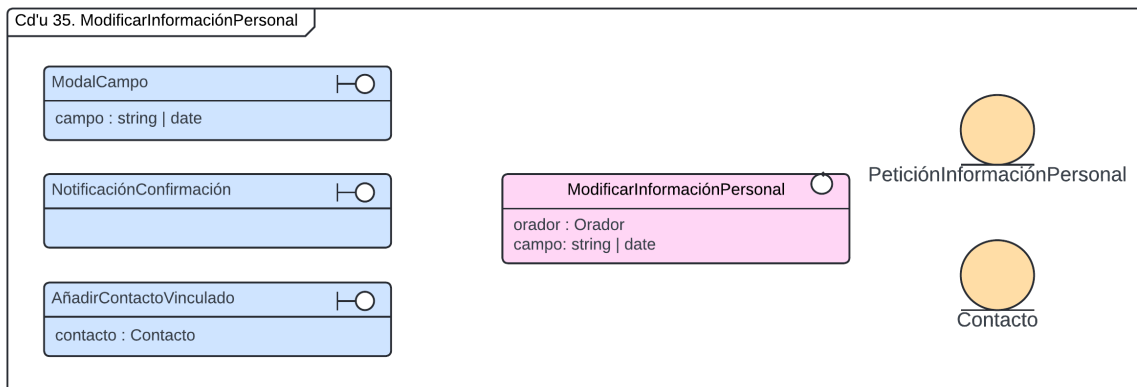


Figura 4.106: Diagrama de clases del caso de uso 35.

# Diagrama de secuencias

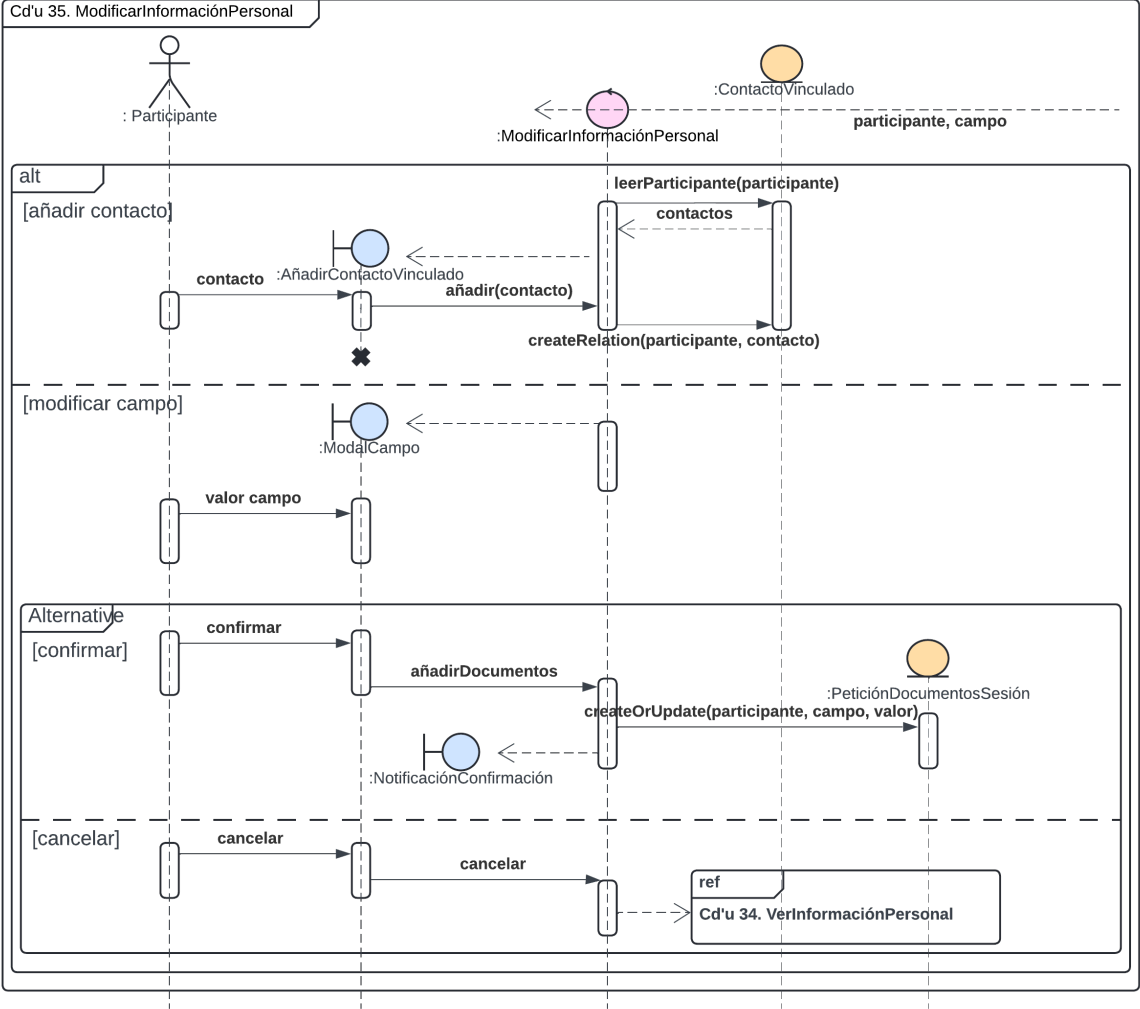


Figura 4.107: Diagrama de secuencias del caso de uso 35.

## 4.2.36. Caso de uso 36. ModificarAutorizaciónAsistente

### Diagrama de actividades

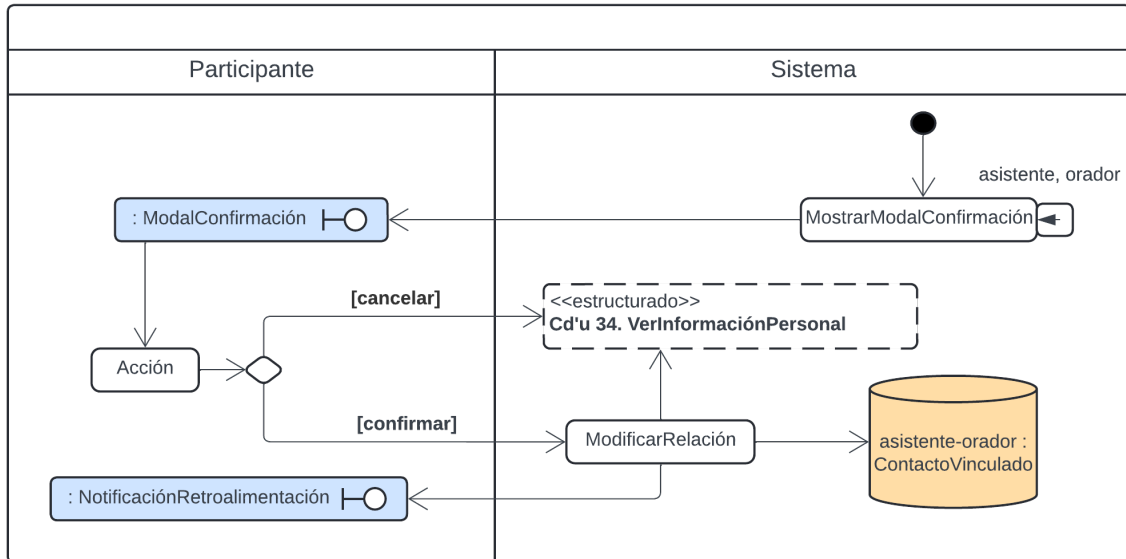


Figura 4.108: Diagrama de actividades del caso de uso 36.

### Diagrama de clases

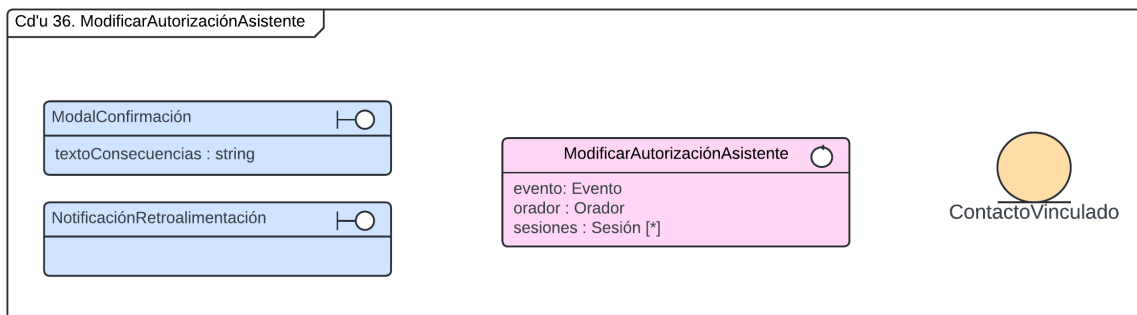


Figura 4.109: Diagrama de clases del caso de uso 36.



## 4.2.37. Caso de uso 37. SeleccionarOrador

### Diagrama de actividades

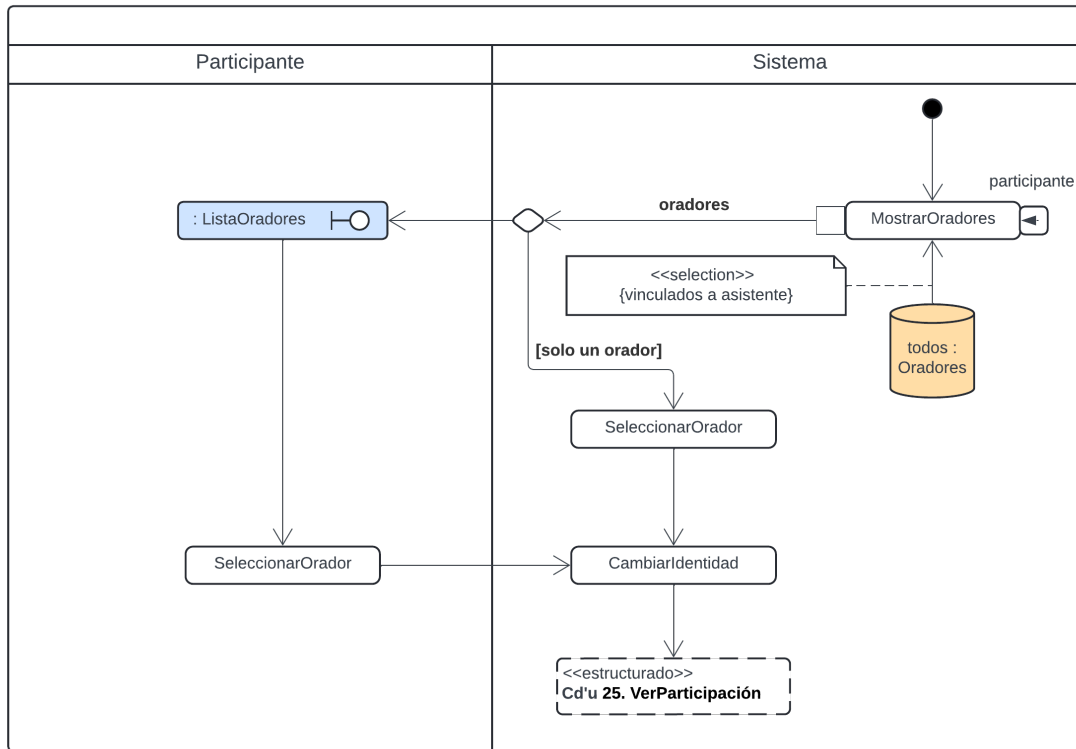


Figura 4.111: Diagrama de actividades del caso de uso 37.

### Diagrama de clases

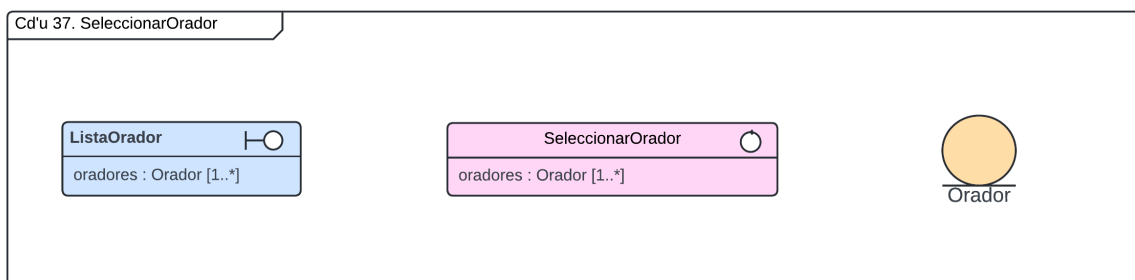


Figura 4.112: Diagrama de clases del caso de uso 37.

# Diagrama de secuencias

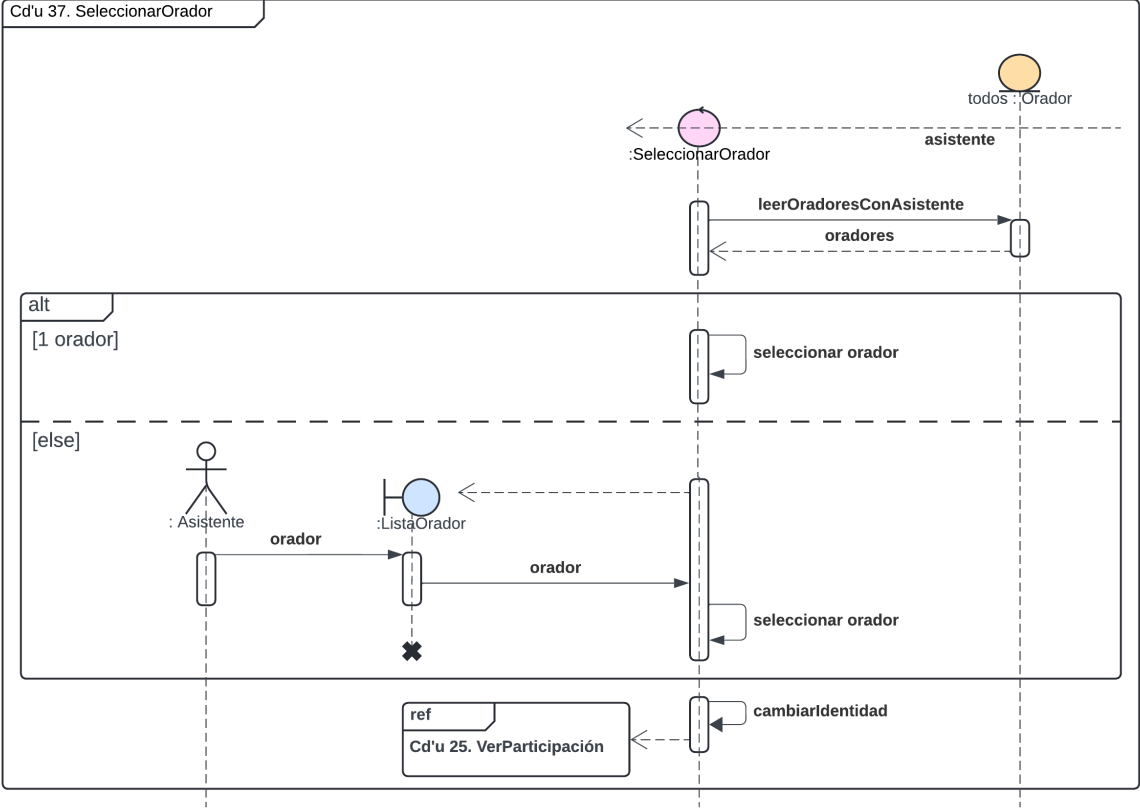


Figura 4.113: Diagrama de secuencias del caso de uso 37.



## 4.4. Especificación de las operaciones de las clases de entidad

### Clase Registro

#### buscarPorCorreo(correo : string) : Registro

Lee y devuelve el primer registro en base al correo pasado por parámetro.

#### datosParticipación() : ...datosParticipación

Devuelve los datos de participación de un registro, descrito en la *Lista 27.1*.

#### actualizar(...campos, ...valores)

Modifica los detalles de la participación de la ficha de registros a partir de los campos y valores recibidos por parámetro.

### Clase Orador

#### leerOradoresConAsistente(contacto : Contacto) : Oradores [\*]

Lee todos los oradores a los que está vinculado un asistente (contacto).

### Clase ContactoVinculado

#### ContactoVinculado(master : Contacto, slave : Contacto, rol : string) : ContactoVinculado

Crea la relación entre dos contactos recibidos por parámetro, del tipo también recibido.

#### leerDeParticipante(participante) : Contacto [\*]

Lee todos los contactos vinculados de un participante recibido por parámetro.

#### modificarAutorización(a : bool)

Cambia el estado de autorización del asistente de un participante.

### Clase PeticiónContacto

#### buscarPeticiónContacto(orador : Orador) : PeticiónContacto

Lee la petición de modificación de los datos de contacto de un orador recibido por parámetro.

#### eliminar()

Elimina una petición de modificación de los datos de contacto.

### Clase PeticiónDocumento

#### buscarPeticiónDocumento(orador : Orador) : PeticiónDocumento

Lee las peticiones de cambio de documentos de las sesiones de un orador recibido por parámetro.

### **eliminar()**

Elimina una petición de modificación de documentos.

## **Clase Usuario**

### **actualizarCódigo()**

Genera un nuevo código de acceso para un participante.

## **Clase Actividad**

### **eliminarRelación(orador : Orador)**

Se elimina la relación entre una actividad y un orador recibido por parámetro.

## **Clase Documento**

### **Documento(...datosDocumento)**

Crea un nuevo documento con los datos recibidos por parámetro.

### **leerDocumentosOrador(orador : Orador) : Documento [\*]**

Lee todos los documentos del orador recibido por parámetro.

### **leerDocumentosSesión(sesión : Sesión) : Documento [\*]**

Lee todos los documentos de la sesión recibida por parámetro.

### **borrarRelación(orador : Orador)**

Borra la relación entre un documento y un orador recibido por parámetro.

### **eliminar()**

Elimina un documento.

## **Clase Artículo**

### **Artículo(...datosArtículo) : Artículo**

Crea un artículo con los datos recibidos por parámetro.

### **leerSegúnEvento(evento : Evento) : Artículo [\*]**

Lee los artículos de un evento recibido por parámetro.

### **leerSegúnOrador(orador : Orador) : Artículo [\*]**

Lee los artículos de un orador recibido por parámetro.

**datosArtículo(artículo) : DatosArtículo**

Devuelve todos los campos de un artículo recibido por parámetro.

**actualizar(...campos, ...valores)**

Actualiza los campos de un artículo en función a los valores recibidos. Ambos elementos recibidos por parámetro.

**eliminar()**

Elimina un artículo.

**Clase Sesión**

**leer(orador : Orador) : Sesión [\*]**

Lee todas las sesiones relacionadas con un orador recibido por parámetro.

**leerDisponibles (orador : Orador) : Sesión [\*]**

Lee las sesiones **disponibles** (no están asignadas y podrían asignarse cumpliendo con la regla de negocio 2.4) para un orador recibido por parámetro.

**desvincular(orador : Orador)**

Desvincula de una sesión al orador recibido por parámetro.

**vincular(orador : Orador)**

Vincula a una sesión el orador recibido por parámetro.

**relacionarDocumentos(documentos)**

Relaciona los documentos recibidos por parámetro con la sesión.

**Clase Organizador**

**leerConParticipante (email : string) : Sesión [\*]**

Lee las sesiones relacionadas con un orador cuyo correo se recibe por parámetro.

**conectar()**

El sistema se conecta a la base de datos de un organizador.

## 5 Diseño

En este apartado se detallará el diseño de la arquitectura de la aplicación, la interfaz de usuario y la persistencia de datos. El objetivo es proporcionar una visión estructurada de como se implementaran los casos de uso en función de los requisitos funcionales y no funcionales descritos en las fases anteriores del proyecto.

### 5.1. Diseño de la arquitectura

#### 5.1.1. Arquitectura básica

Para la implementación del aplicativo, se ha decidido utilizar el patrón de diseño MVC [6]. Este patrón es ampliamente utilizado en el desarrollo de aplicaciones web, principalmente debido a cómo separa las responsabilidades en tres capas: la gestión de datos (**Modelo**), la presentación de datos (**Vista**) y la lógica de negocio (**Controlador**).

En una arquitectura típica de 3 capas para aplicaciones web, estas responsabilidades se distribuyen de la siguiente manera:

- **Modelo:** La capa de modelo es responsable de gestionar la lógica de datos de la aplicación, incluyendo la recuperación y almacenamiento de datos en la base de datos. Maneja las operaciones Create, Read, Update, Delete (CRUD) (creación, lectura, actualización y borrado de datos) y asegura la integridad y coherencia de los datos.
- **Vista:** La capa de vista se encarga de la presentación de los datos al usuario. Genera el Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) y JavaScript que se envían al cliente (navegador del usuario), asegurando que la interfaz de usuario sea intuitiva y receptiva.
- **Controlador:** La capa de controlador actúa como un intermediario entre la vista y el modelo. Procesa las peticiones del cliente, realiza las operaciones necesarias sobre los datos a través del modelo y devuelve las vistas apropiadas al cliente. Gestiona la lógica de negocio y el flujo de la aplicación.

Esta separación por capas asegura la separación de responsabilidades, generando un código mas estructurado, escalable y mantenible en el futuro. El patrón permite que se realicen cambios en una capa de forma transparente al resto mientras se mantengan los parámetros de entrada y salida.

### 5.1.2. Multitenencia

La multitenencia [7] es una de las principales dificultades de este proyecto debido a la necesidad de gestionar datos de múltiples clientes/organizadores de eventos de manera aislada y segura. En un modelo B2B orientado a organizadores de eventos, es crucial evitar que los datos de un cliente se filtren a otro.

#### Opciones de multitenencia

Existen varias opciones para implementar la multitenencia:

- **Base de datos compartida, esquema por cliente:** Todos los clientes comparten la misma base de datos, pero cada uno tiene su propio esquema.
- **Base de datos compartida, tablas por cliente:** Todos los clientes comparten la misma base de datos y las mismas tablas, pero cada fila incluye un identificador de cliente.
- **Base de datos separada por cliente:** Cada cliente tiene su propia base de datos.

#### Opción elegida y justificación

Para este proyecto, se ha optado por la **base de datos separada por cliente**. Esta opción fue seleccionada por varias razones:

- **Seguridad:** Aislar los datos de cada cliente en bases de datos separadas minimiza el riesgo de filtraciones de datos debido a peticiones mal construidas o ataques malintencionados.
- **Escalabilidad:** Permite escalar de manera independiente cada base de datos según las necesidades específicas de cada cliente.
- **Mantenimiento:** Facilita el mantenimiento y la gestión de datos, ya que las actualizaciones y backups se pueden realizar de forma aislada.

### 5.1.3. Despliegue *serverless*

Para manejar los picos de demanda y asegurar una escalabilidad eficiente, se ha optado por una arquitectura *serverless*.

#### Arquitectura *serverless*

La arquitectura *serverless* delega la gestión de la infraestructura a un proveedor de nube. Los servicios clave incluyen:

- **FaaS (Function as a Service):** Se utiliza para servir las solicitudes Hypertext Transfer Protocol (HTTP), permitiendo una escalabilidad automática y bajo demanda. Esto es crucial para manejar los picos de tráfico en días cercanos a los eventos.

- **Base de datos en la nube:** Se emplea una base de datos relacional gestionada en la nube, que proporciona alta disponibilidad y escalabilidad.
- **Otros servicios gestionados:** Utilización de servicios como almacenamiento de objetos en la nube, servicios de envío de correos electrónicos, y servicios de manejo de colas en la nube.

### Motivación para la elección de serverless

La elección de una arquitectura *serverless* se basa en varias ventajas significativas:

- **Escalabilidad automática:** Una de las características implícitas a la creación de un software de eventos es la existencia de picos de uso en los días cercanos a un evento, manteniendo el uso muy bajo el resto del tiempo. Los servicios FaaS pueden escalar automáticamente en respuesta a aumentos de tráfico, lo que es ideal para manejar la situación descrita.
- **Coste efectivo:** Pago por uso real, lo que reduce los costos durante los períodos de baja actividad.
- **Mantenimiento reducido:** El proveedor de nube maneja la infraestructura, lo que reduce la carga de mantenimiento y permite al equipo de desarrollo centrarse en la lógica de negocio.

En resumen, la combinación de una arquitectura MVC para una clara separación de responsabilidades, una estrategia de multitenencia basada en bases de datos separadas para cada cliente, y el despliegue *serverless* para manejar eficientemente los picos de demanda, proporciona una solución robusta, segura y escalable para la gestión de eventos.

## 5.2. Diseño de la interfaz

Se ha diseñado una interfaz capaz de permitir el acceso a todos los casos de uso descritos con anterioridad. La interfaz se ha creado con la usabilidad y la conveniencia como pilares centrales del diseño, descritos como requisitos no funcionales de la aplicación. A continuación, se mostrará el diseño de la interfaz y se relacionará con los casos de uso listados.

### 5.2.1. Listado de oradores

Para cumplir con el *Caso de uso 01. VerOradores*, se ha diseñado una tabla que lista los oradores con todos sus atributos. Se puede iniciar el *Caso de uso 02. CrearOrador* directamente con un botón en la cabecera de la tabla.

Cada fila de la tabla dispone de botones que permiten iniciar las acciones individuales referentes a:

- *Caso de uso 03. ModificarOrador*
- *Caso de uso 04. BorrarOrador*

Además, si se hace clic sobre cualquiera de las filas, se inicia el *Caso de uso 05. VerDetallesOrador*.

Reg. Number	Notifications	First name	Last name	Function	Organisation	Email	Pax status	Featured	Status	Badge Template
22		Ramon	Garcia	Minister of Economy	Municipality of Grimbergen	ramon.garcia@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>
23		Michal	Adam		Credit Agricole Group	MichalAdam.KAPA@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>
24		Edgar	Miralles	FullStack developer	STAB-21 SL	tester@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>

Figura 5.1: Interfaz del listado de oradores.

Los oradores de la tabla pueden ser seleccionados para iniciar las siguientes acciones en masa:

- *Caso de uso 04. BorrarOrador*
- *Caso de uso 19. EnviarCorreo*

Reg. Number	Notifications	First name	Last name	Function	Organisation	Email	Pax status	Featured	Status	Badge Template
<input checked="" type="checkbox"/>		Ramon	Garcia	Minister of Economy	Municipality of Grimbergen	ramon.garcia@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>

Figura 5.2: Acciones en masa disponibles al seleccionar uno o varios oradores.

## Peticiones de validación

Para mejorar la conveniencia de la lista, se ha añadido una columna que muestra el estado de las peticiones de cambio de cada uno de los oradores. Por ejemplo, en el caso de que un orador genere suba nuevos documentos o cambie su información personal, aparecerá una etiqueta mostrando concretamente la acción realizada.

Reg. Number	Notifications	First name	Last name	Function	Organisation	Email	Pax status	Featured	Status	Badge Template
22	Special Requests Updated	Ramon	Garcia	Minister of Economy	Municipality of Grimbergen	ramon.garcia@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>
23		Michal Adam			Crédit Agricole Group	MichalAdam.KAPA@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>
24	Validation Needed Media Updated	Edgar	Miralles	FullStack developer	STAS-21 SL	tester@nudato.com	Speaker	<input type="checkbox"/>	Registered	<a href="#">Edit</a> <a href="#">Delete</a>

Figura 5.3: Listado de Oradores con peticiones pendientes de revisión.

## Filtros

Tal y como se indica en el caso de uso 01. VerOradores, se pueden establecer filtros que actúan como parámetro de entrada en la búsqueda de Oradores. Los filtros disponibles son:

- Etiqueta
- Estado
- Validación pendiente
- Estado de actividad
- País
- Organización

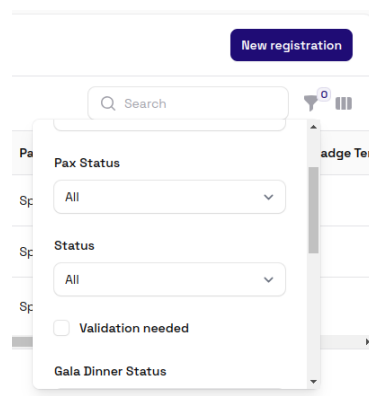


Figura 5.4: Filtros para oradores.

### Caso de uso 03. ModificarOrador

Al iniciar este caso de uso a partir del anterior, se genera un modal que cumple con la descripción del caso en particular:

The image shows a modal window titled "Edit 23" with a close button in the top right corner. It contains two tabs: "Speaker Information" (active) and "Speaker Media". The form is organized into two columns. The left column includes fields for "Contact" (with a dropdown menu showing "Crédit Agricole Group | Michal Adam | MichalAdam.KAPA@nudato.com"), "Confirmation date" (calendar icon), "Status" (dropdown menu with "Registered"), "Hotel" (dropdown menu with "Select an option"), "Social event" (dropdown menu with "Cocktail de mi padre jamon"), "Gala dinner table" (dropdown menu with "Gala dinner amiga 1"), "Flights in date time" (calendar icon), "Flights out date time" (calendar icon), and "Badge template" (dropdown menu with "Select an option"). The right column includes fields for "Login Code", "Badge date" (calendar icon), "Payment status" (dropdown menu with "Select an option"), "Number of nights" (text input), "Social event status" (dropdown menu with "Not Attending"), and "Gala dinner status" (dropdown menu with "Attending"). There are also two checkboxes: "On site registration" and "Featured". At the bottom of the form, there are two buttons: "Add to sessions" and "Add to special requirements". At the very bottom of the modal, there are two buttons: "Save changes" and "Cancel".

Figura 5.5: Modal de edición del orador del Caso de uso 03. ModificarOrador.

### Caso de uso 04. BorrarOrador

En iniciarse, se muestra un modal, tal y como describe el caso de uso:

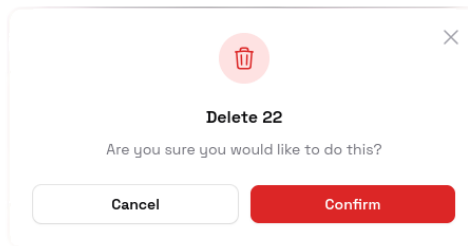


Figura 5.6: Modal de confirmación de borrado.

### Caso de uso 19. EnviarCorreo

Al iniciarse, se abre un modal con los campos necesarios para el envío de correo, descritos en el caso de uso

A modal dialog box titled 'Send email' with a close button in the top right corner. It contains several form fields: 'Sender Email\*' with a dropdown menu showing 'Select an option'; 'Reply To\*' with a dropdown menu showing 'Select an option'; 'BCC' with an 'Add BCC' input field; 'CC' with an 'Add CC' input field; a toggle switch for 'No template' which is currently turned off; and 'Template\*' with a dropdown menu showing 'Select an option'. At the bottom, there are two buttons: a dark blue 'Submit' button and a white 'Cancel' button.

Figura 5.7: Modal de configuración de correo.

## 5.2.2. Detalles del orador

Tal y como indica la descripción del *Caso de uso 05. VerDetallesOrador*, se presenta una interfaz que muestra todos los datos y relaciones de un orador seleccionado. Además, permite cambiar entre las diferentes ventanas de información y realizar acciones de gestión sobre el orador.

### Ventana de información general

Esta ventana permite visualizar la información personal del contacto, su participación al evento y añadir vínculos a otros contactos. Además, permite iniciar los siguientes casos de uso:

- *Caso de uso 07. ModificarInformación*: Accesible haciendo click en el icono de editar en la carta que muestra la información personal, la de participación o la biografía.
- *Caso de uso 08. AñadirContactoVinculado*: Accesible haciendo click al botón de '+' en la carta de contactos vinculados.

The screenshot displays the 'Speaker Information' window for MR. RAMON GARCIA. It features a navigation bar with 'Speaker Information', 'Sessions and Media', and 'Articles'. The main content is organized into several panels:

- Speaker Information:** Includes a profile picture, name (MR. RAMON GARCIA), and a list of fields: TITLE (Mr.), FIRST NAME (Ramon), LAST NAME (Garcia), EMAIL (ramon.garcia@nudato.com), PHONE (123123), MOBILE (123123), ORGANISATION (Municipality of Grimbergen), DEPARTMENT (Dep Dep Dep), OFFICE ADDRESS (Address Address Address Address), and FUNCTION (Minister of Economy).
- Sessions and Media:** A green card for MR. RAMON GARCIA, Minister of Economy - Municipality of Grimbergen, with contact details: La calle, El edificio, CW - La ciudad - Curaçao, ramon.garcia@nudato.com, +599 123123, +599 123123.
- Articles:** A section for the speaker's biography, currently showing 'Biografía Biografía Biografía Biografía'.
- LINKED CONTACTS:** A section for assistants and assistant for, currently showing 'This contact has no assistants linked yet.' and 'This contact is not copy to for any other.'
- EVENT PARTICIPATION:** A table showing event details: REG STATUS (Registered), HOTEL (Hotel Test), Nº OF NIGHTS (3), SOCIAL EVENT (-), GALA DINNER TABLE (Mesa Test, Invited).
- SPECIAL REQUIREMENTS:** A section for special requirements, currently showing 'Requisito 1234 Test' and '26 May 2024 12:28'.

Figura 5.8: Ventana de información de los detalles del orador.

### Caso de uso 06. ValidarPeticiones

En caso de haber alguna petición pendiente de validar, al acceder al caso de uso se iniciará el caso de uso 06. ValidarPeticiones. Aquí se mostrarán al organizador los cambios que ha pedido el participante sobre su información. Estas peticiones pueden ser:

- Cambios en la información personal.
- Cambios en los documentos de alguna sesión.
- Cambios en la biografía

Todos los posibles cambios se tratan de la misma manera: el organizador ve resaltados los cambios que el participante ha pedido. El primero tiene 2 opciones: aceptar la modificación en

el campo o rechazarla. Una vez se encuentra satisfecho con las decisiones, guarda los cambios.

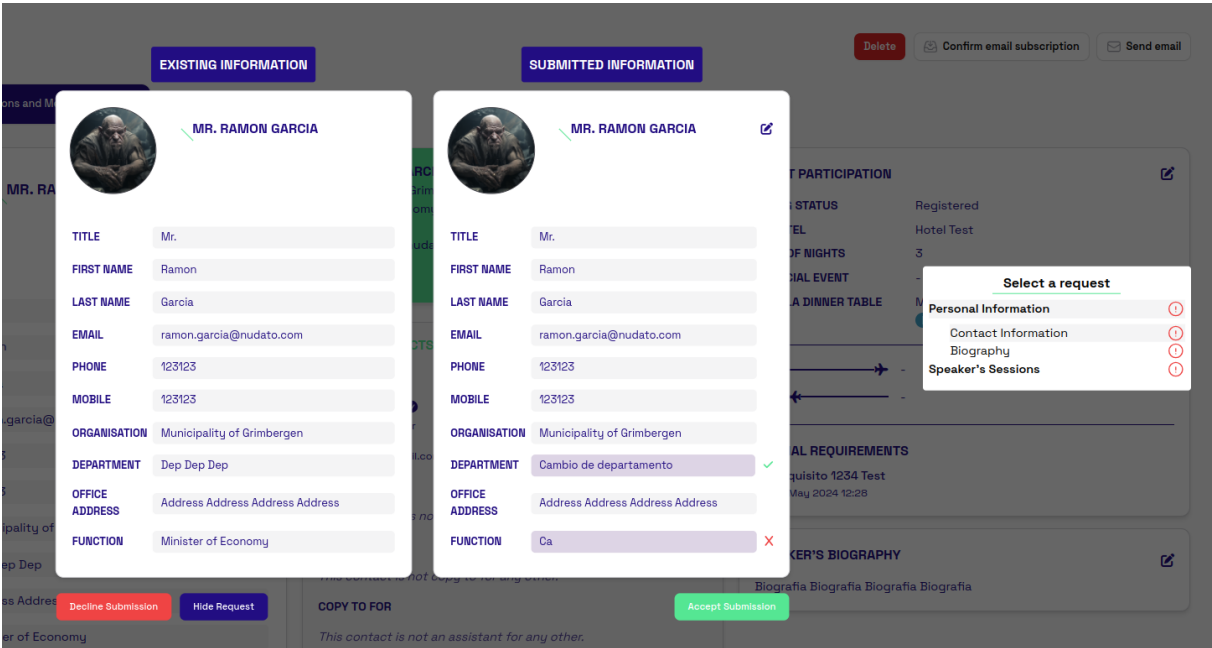


Figura 5.9: Vista de validación de los datos de contacto del participante.

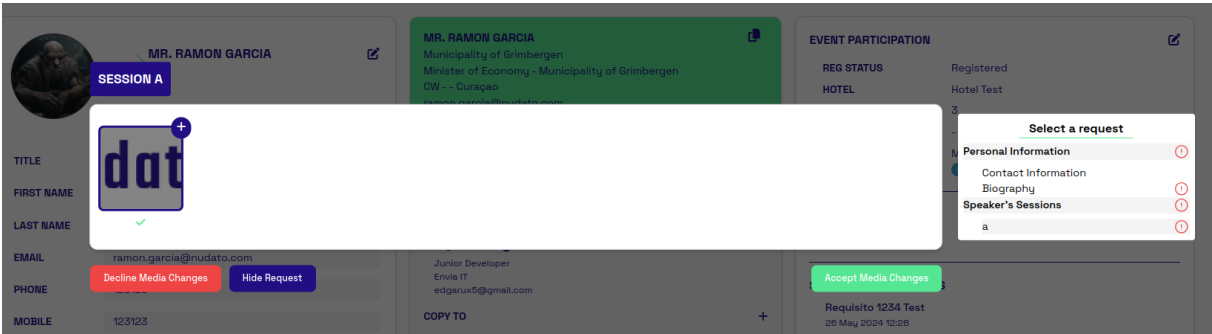
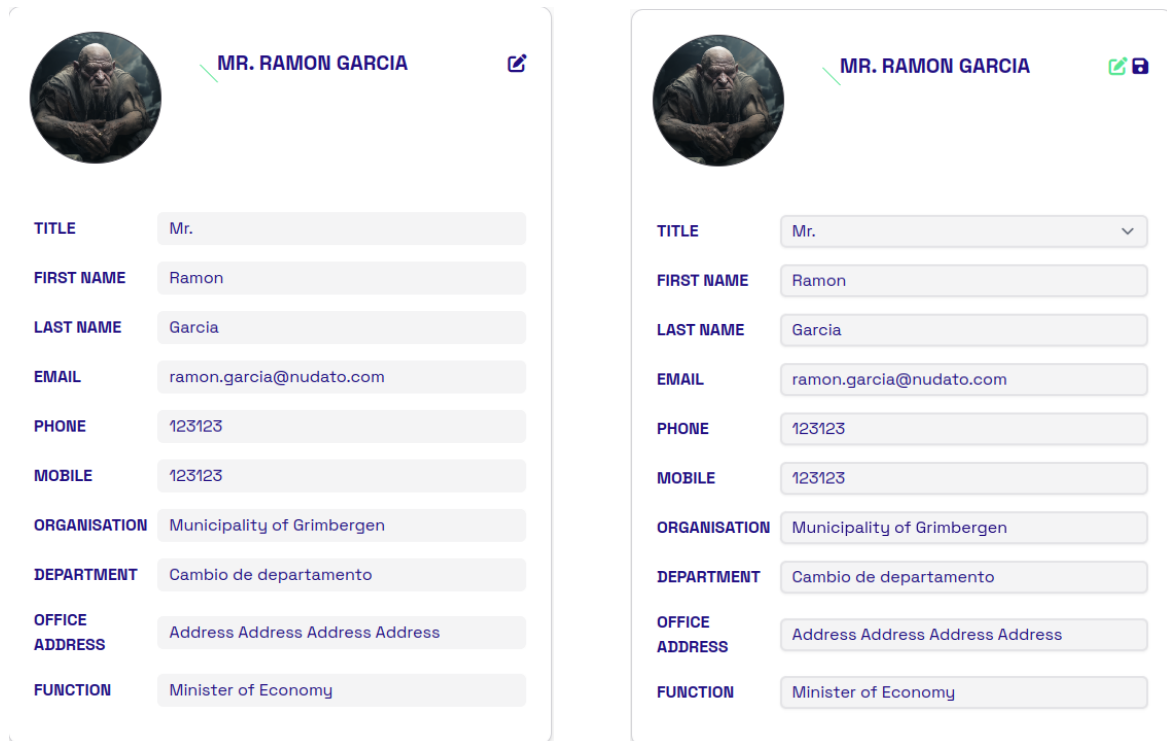


Figura 5.10: Vista de validación de los documentos de una sesión del orador.

## Caso de uso 07. ModificarInformación

Al iniciar el caso de uso, la carta elegida para ser editada pasará a mostrar campos editables y un botón de guardar.



Field	Value
TITLE	Mr.
FIRST NAME	Ramon
LAST NAME	Garcia
EMAIL	ramon.garcia@nudato.com
PHONE	123123
MOBILE	123123
ORGANISATION	Municipality of Grimbergen
DEPARTMENT	Cambio de departamento
OFFICE ADDRESS	Address Address Address Address
FUNCTION	Minister of Economy

Figura 5.11: Carta de información de contacto antes y después de iniciar el caso de uso 07. ModificarInformación.

## Caso de uso 08. AñadirContactoVinculado

Al iniciarse el caso de uso, la carta original cambia, incorporando un buscador para buscar entre los contactos disponibles y una casilla de verificación en caso de que sea un asistente para marcarlo como verificado para ese orador.

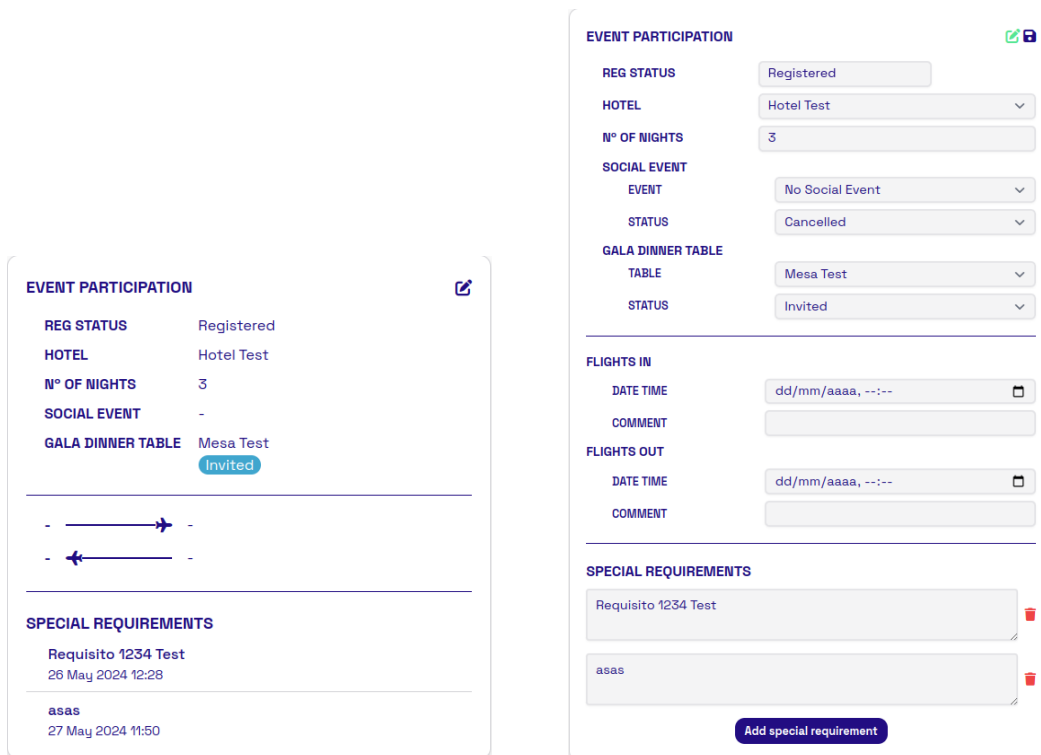


Figura 5.12: Carta de información de participación antes y después de iniciar el caso de uso 07. ModificarInformación.

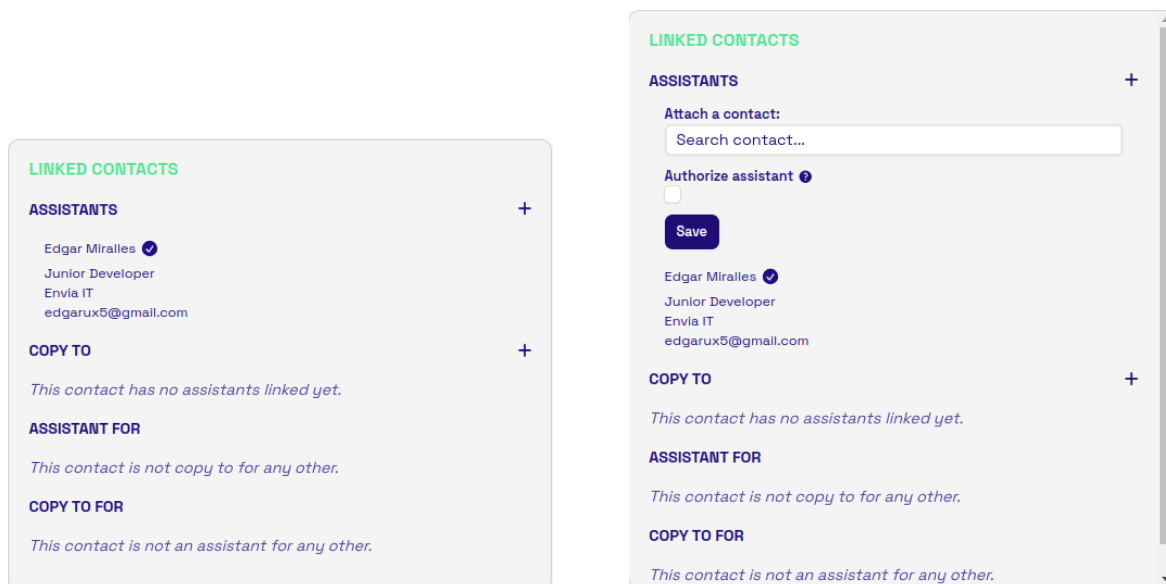


Figura 5.13: Carta de contactos vinculados antes y después de iniciar el caso de uso 08. Añadir-ContactoVinculado.

## Ventana de sesiones y documentos

Esta ventana permite visualizar las sesiones del orador, la documentación del orador y los vínculos entre la documentación y sus sesiones. Además, permite iniciar los siguientes casos de uso:

- Caso de uso 09. BorrarSesion: Accesible haciendo clic al icono de borrar que aparece en cada cabecera de sesión al pasar el ratón por encima.
- Caso de uso 10. AñadirSesion: Accesible haciendo clic al botón *New Session* o al de *Add Session*.
- Caso de uso 11. ModificarDocumentos: Accesible haciendo clic al botón *Add Media* en los documentos generales del orador o en cualquier botón + en sus respectivas sesiones.

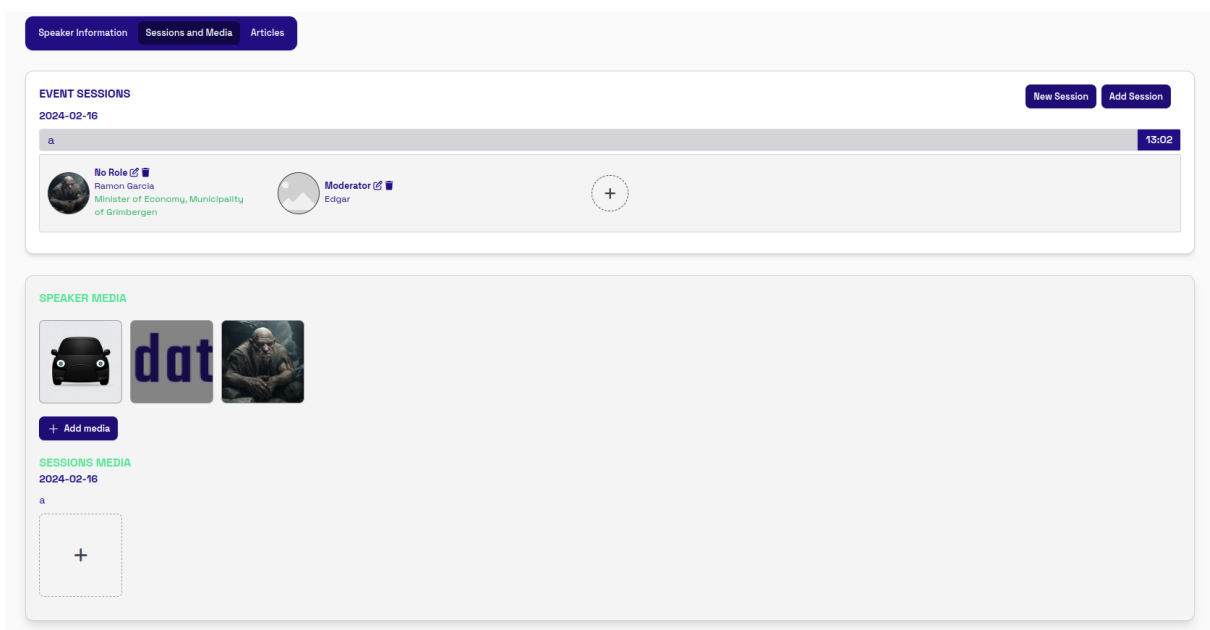


Figura 5.14: Ventana de sesiones y documentos del orador.

### Caso de uso 09. BorrarSesión

Al iniciar la acción, se muestra un modal de confirmación tal como indica la descripción del caso de uso.

### Caso de uso 10. AñadirSesión

Al iniciar la acción mediante el botón de *New Session*, se muestra un modal con los campos descritos en la *Lista 10.1*. Si se inicia mediante el botón de *Add Session*, solo se muestra un desplegable con las sesiones disponibles.

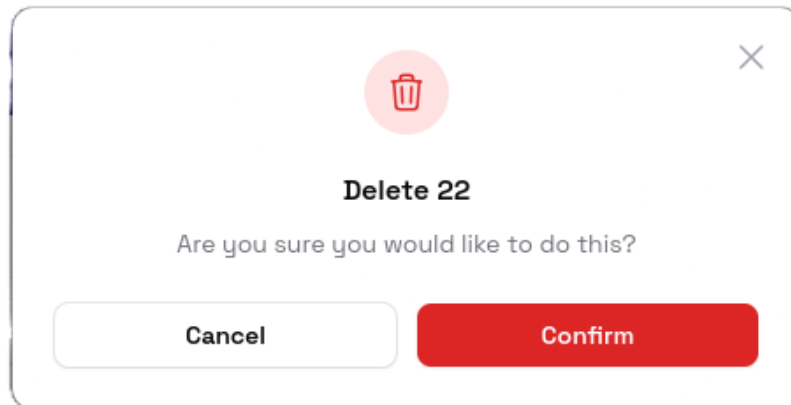


Figura 5.15: Modal de confirmación de borrado de sesión.

### **Caso de uso 11. Modificar Documentos**

Al iniciar la acción mediante el botón de *Add Media*, se muestra un modal con la carpeta de documentos del orador. Desde ahí, puede subir nuevos archivos o seleccionar los que utilizar. Si se inicia mediante el botón de + en una sesión, se muestra un modal con los documentos generales del orador, permitiendo seleccionar los que utilizar. Para borrar un documento de una sesión, se debe utilizar el botón con un icono de basura en la esquina superior derecha del documento.

**Create event session**
✕

01 **General**  
Define your session details

02 **Configuration**  
Set optional parameters

03 **Speakers**  
Set the session speakers

**Name\***

B I U S “ ☰ ☷ ↺ ↻

**Shorter session name**

B I U S “ ☰ ☷ ↺ ↻

The session name will be used if this field is empty

**Location**

Select an option ▼

**Category**

Select an option ▼

**Start\***

Session starts at 12/02/2024, 13:00:00 📅

**End**

Session ends at 12/02/2024, 14:00:00 📅

**Description**

B I U S 🔗 Heading Subheading “ </> ☰ ☷ 🖼️ ↺ ↻

Cancel

Next

Figura 5.16: Modal de creación de sesión.

**Create registration event session**
✕

**Session**

Select an option ▼

Create

Create & create another

Cancel

Figura 5.17: Modal de añadir sesión existente.

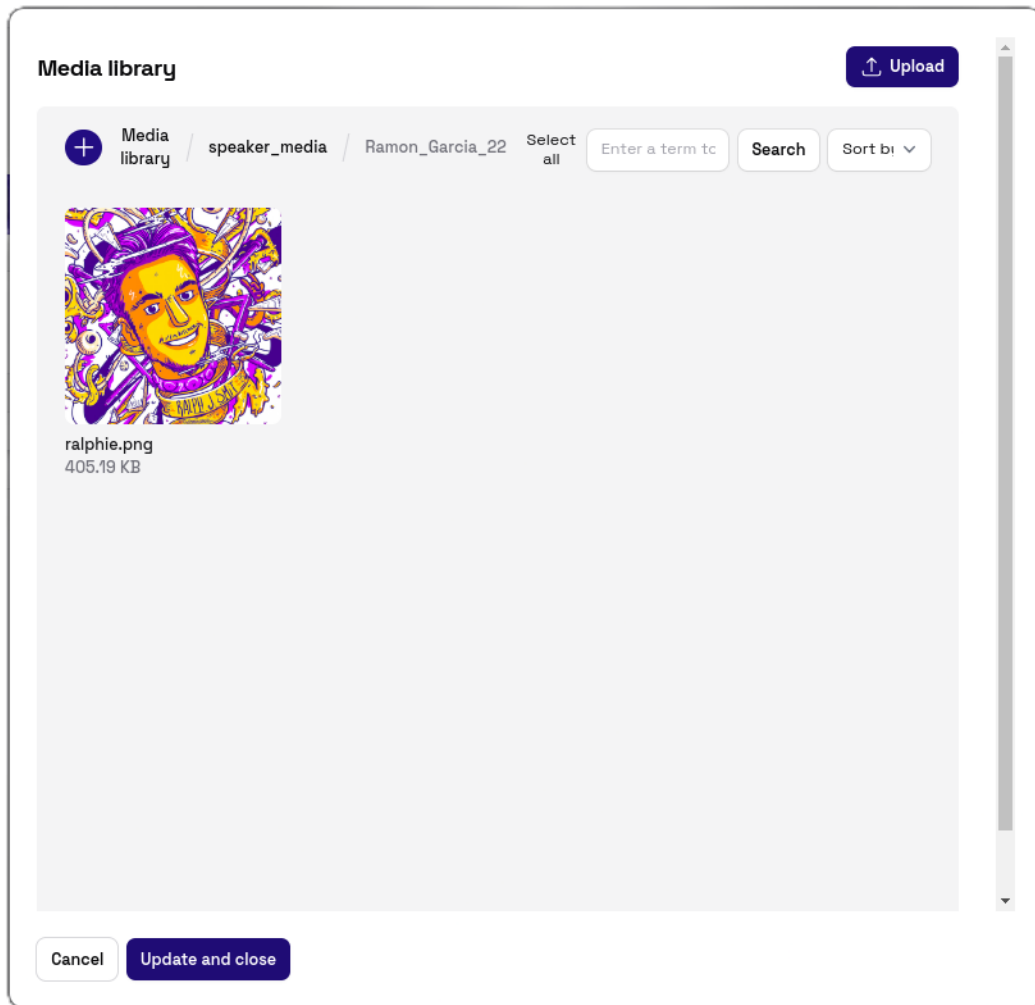


Figura 5.18: Modal de añadir sesión existente.

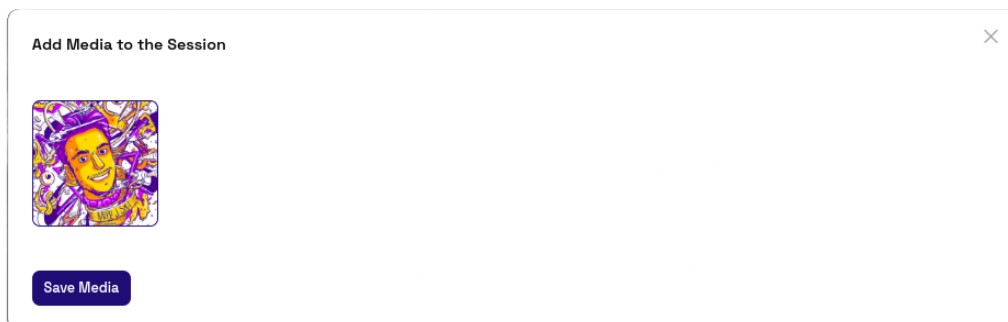


Figura 5.19: Modal con los documentos disponibles para ser añadidos a la sesión.

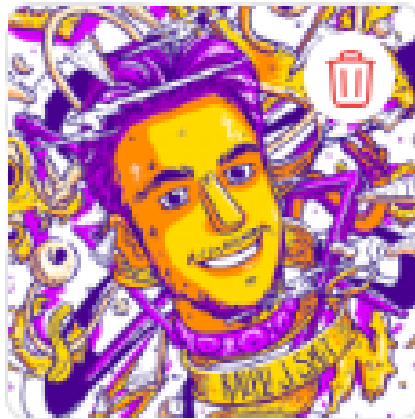


Figura 5.20: Documento en estado de *hover*, mostrando la basura para ser eliminado de la sesión.

### 5.2.3. Listado de artículos

Para cumplir con el *Caso de uso 12. VerArtículos*, se ha diseñado una tabla que lista los artículos con todos sus atributos. Los artículos se ordenan según su sección y subsección, establecidas a la hora de crearlo. Para facilitar el trabajo y proveer del orden, las secciones y subsecciones se configuran en la configuración básica de un evento a la hora de crearlo o editarlo.

The screenshot displays a web interface for configuring magazine sections. At the top, a header bar for 'Seccion2' includes a sort icon (up/down arrows) and a delete icon (trash can). Below this, a 'Name' field contains the text 'Seccion2'. A section titled 'Magazine subsections' features 'Collapse all' and 'Expand all' options. Two subsections are visible: '2.1 Subseccion1' and '2.2 Subseccion2'. Each subsection has its own header with a sort icon and a delete icon. Below each subsection header are two input fields: 'Index name' (containing '2.1' and '2.2' respectively) and 'Name' (containing 'Subseccion1' and 'Subseccion2' respectively). At the bottom of the interface is a button labeled 'Add Subsection'.

Figura 5.21: Interfaz de creación de una sección.

Se puede iniciar el *Caso de uso 13. CrearArtículo* directamente con un botón en la cabecera de la tabla.

**Create article**
✕

**Speaker\***

Select an option

**Max Length**

Select an option

Speakers must be registered to the event

**Section\***

Select an option

**Subsection\***

Select an option

**Topic\***

**Deadline date\***

dd/mm/yyyy

**Order in subsection**

**Title min length\***

**Title max length\***

**Key sentence min length\***

**Key sentence max length\***

Create

Create & create another

Cancel

Figura 5.22: Modal de creación de un artículo.

Cada fila de la tabla dispone de botones que permiten iniciar las acciones individuales referentes a:

- Caso de uso 17. BorrarArtículo
- Caso de uso 18. ModificarPetición

Además, si se hace clic sobre cualquiera de las filas, se inicia el Caso de uso 14. VerDetallesArtículo.

Articles													New article		
													Search		
<input type="checkbox"/>	Subsection	Actions needed	First Name	Last Name	Organisation	Max Length	Topic	Deadline date	Order	Article submission	Submission date	Status	Title	Picture	Key sentence
<input type="checkbox"/>	Section: Seccion2														
<input type="checkbox"/>	2.2		Edgar	Miralles	Envia IT	Unlimited	ds	May 25, 2024	1	No	No submission	New			

Showing 1 result

Per page 10

Figura 5.23: Interfaz del listado de artículos.

Los artículos de la tabla pueden ser seleccionados para iniciar las siguientes acciones en masa:

- Caso de uso 17. BorrarArtículo
- Caso de uso 19. EnviarCorreo

Subsection	Actions needed	First Name	Last Name	Organisation	Max Length	Topic	Deadline date	Order	Article submission	Submission date	Status	Title	Picture	Key sentence
2.2		Edgar	Miralles	Envia IT	Unlimited	ds	May 25, 2024	1	No	No submission	New			

Figura 5.24: Acciones en masa disponibles al seleccionar uno o varios artículos.

## Peticiones de validación

De la misma manera que en la lista de oradores, se ha añadido una columna que muestra el estado de las peticiones de cambio y notificaciones de cada uno de los artículos. Mostrará etiquetas al recibir nuevas respuestas, peticiones de cambio de fecha... Además, en el caso de que un orador rechace un artículo, será representado con un fondo de color rojo.

Subsection	Actions needed	First Name	Last Name	Organisation	Max Length	Topic	Deadline date	Order	Article submission	Submission date	Status	Title	Picture
1.1					12000	dsdsda	May 3, 2024	2	No	No submission	New		
		Edgar	Miralles	STAS-21 SL	Unlimited	sades	Mar 30, 2024	3	Yes	2024-03-20 17:28:38	New		
	No picture	Edgar	Miralles	STAS-21 SL	4500	What are the priorities for th...	Jun 4, 2024	9	Yes	2024-04-19 16:31:14	Validated		
1.2		Edgar	Miralles	STAS-21 SL	8000	aaaa	Apr 28, 2024	No order	No	No submission	New		
	New Version	Ramon	Garcla	Municipality of Grimbergen	Unlimited	das	May 20, 2024	No order	Yes	2024-05-25 13:46:24	New Version		
	No picture	Edgar	Miralles	STAS-21 SL	Unlimited	frefftds	Mar 30, 2024	1	No	No submission	Rejected		
	New Version	Edgar	Miralles	STAS-21 SL	Unlimited	dsadsadas	Mar 30, 2024	8	Yes	2024-04-19 16:07:40	New Version		

Figura 5.25: Listado de artículos con nuevas versiones y rechazos.

## Filtros

De la misma manera que ocurre con el listado de oradores, se pueden aplicar filtros al listado de artículos. Los filtros disponibles son:

- Entregados
- Filtro por estado
- Fecha de entrega

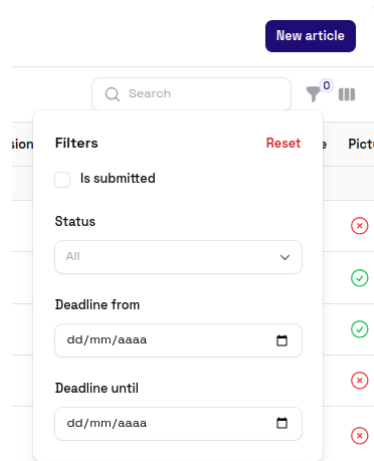


Figura 5.26: Filtros para artículos.

### Caso de uso 18. ModificarPetición

Al iniciar este caso de uso a partir del anterior, se genera un modal como el que se muestra en la Figura 5.22

### Caso de uso 17. BorrarArtículo

En iniciarse, se muestra un modal, tal y como describe el caso de uso, con el fin de pedir una confirmación previa al borrado. Este modal es idéntico al de la Figura 5.15.

### Caso de uso 19. EnviarCorreo

Seleccionando cualquiera de los artículos, puede enviarse un correo al/a los oradores correspondientes. Esto se lleva a cabo mediante el uso del mismo modal representado en la Figura 5.7.

## 5.2.4. Detalles del artículo

Tal y como indica la descripción del *Caso de uso 14. VerDetallesArtículo*, se presenta una interfaz que muestra todos los datos sobre la petición y la respuesta del artículo (en caso de haber una). Estos 2 conjuntos de datos se separan en dos ventanas distintas.

### Ventana de configuración de la petición

Esta ventana permite visualizar y editar la petición del artículo hacia un orador. Esta ventana muestra los campos descritos en la *Lista 13.1*. Además, permite iniciar los siguientes casos de uso:

- *Caso de uso 15. ValidarRespuesta*
- *Caso de uso 16. ValidarPeticiónFecha*

- Caso de uso 18. ModificarPetición
- Caso de uso 19. EnviarCorreo

Figura 5.27: Ventana de configuración de la petición del artículo.

## Ventana de configuración de la respuesta

Esta ventana muestra la respuesta del orador si existe y las posibles peticiones de cambio de fecha relacionadas con el Caso de Uso 16. ValidarPeticiónFecha.

## Caso de Uso 15. ValidarRespuesta

En el caso de querer validar la respuesta recibida, el organizador puede acceder al caso de uso desde la ventana de *Reply*. Allí, después de revisar el contenido de la respuesta, encontrará un botón que mostrará todos los estados asignables al artículo. Después de esto, el orador recibirá un correo de confirmación.

Configuration Reply

**Title and Key Sentence**

<b>Title</b> EU Comp: actions speak louder 28 characters	<b>Key Sentence</b> Europe faces an urgent chaaaa 30 characters
--	---

**Content**

A new year begins, but the feeling is the beginning of a new geopolitical era – the world continues to evolve at dazzling speed. In this regard, Europe faces an urgent challenge to change its trajectory: we need to fix the state of the EU's competitiveness. I am delighted EU leaders recognize the importance of the task at hand and that this cannot be business as usual.

If we look at where we are now, plenty of examples appear. Take the list of the global top 20 companies by market cap, we hardly find any European champions. In terms of market size, the US equity markets are the largest in the world and continue to be among the deepest, most liquid and most efficient, representing 42.9% of the \$106tn global equity market cap in 2023. This is 4.1x the next largest market, China, followed by Europe.

So, how do we push the power button?

First, looking at the distribution of power between European institutions and Member States. We need to be more ambitious about the EU's political and economic integration if we really want to advance towards a Single Market. Member States should cede competences to the EU institutions in areas such as defense, migration, energy or mobility infrastructure.

All that is needed to ensure the free movement of goods and persons. We have reached a point where not advancing in integration is leaving the EU behind other more integrated areas that can take full advantage of their economies of scale. If we renounce to scale EU solutions, we are renouncing to be on the race.

Second, changing our fiscal policy. As long as we do not have some kind of fiscal union, we will continue having a fragmented euro in our pockets with asymmetric fiscal policies. The response to the European shared challenges ahead (climate, digitization or security) will be suboptimal in the absence of some common fiscal policy, in a period where great strategic view and investment are needed.

Third, changes needed in the regulatory framework towards the completion of the Single Market:

- To complete the Capital Markets Union. Starting by boosting the EU securitization market and explore ways to enable banks to free up capital and liquidity for the express purpose of providing additional funding to EU businesses. This should include an immediate review of the EU securitization framework.
- Need to finalise the Banking Union by establishing a common risk sharing mechanism: the European Deposit Insurance Scheme (EDIS). Without it, there is not a level playing field for cross-border offering of retail financial services. If we want to build a real Single Market, depositors should feel that they are equally protected in all countries across Europe.
- We need to work on harmonizing the regulatory framework. Although the single rulebook is a fundamental piece, the most common regulatory tool still are Directives and still national rules play a key role – there are many examples: insolvency frameworks, consumer protection rules, etc. The existence of different regulatory frameworks is the main barrier to European consolidation, due to the number of resources needed to understand and implement the different national regulatory frameworks.
- This takes us to the lack of potential synergies that could be achieved in a merger. If regulatory frameworks are different, merged banks would still need different teams to deal with the different national frameworks, different products, different procedures to attend customers, different IT systems which are designed to give response to the specific regulatory framework in each country. Mortgages are different, payment commissions are different. There are limited cost synergies and economies of scale.
- We need to improve the resolution framework. It is key to facilitate an acquisition regime for failed entities. Making the acquirer responsible for the conduct of management of the acquired entity before resolution is normally unjustified. This cannot be easily prevented through due diligence, bearing in mind the rapid reaction needed in this process. So, the responsibility regime should be reviewed. Lessons can also be taken from the US. Banks do not acquire the legal entity, but specific assets and liabilities.
- Finally, competitiveness is not only about the Single Market, but about supporting EU business operating in third countries. International companies are crucial to the competitiveness of the European industry and the EU's strategic autonomy. A top priority for the next Commission should be to ensure that the EU regulatory framework does not penalize highly diversified businesses operating in third countries and that duly recognizes banks that operate in those countries.

Figura 5.28: Ventana de respuesta de un artículo.



Figura 5.29: Correo de retroalimentación de validación de respuesta.

## Caso de Uso 16. Validar Petición Fecha

En caso de existir una fecha a validar, se le mostrará al organizador en la ventana de respuesta. El organizador, siguiendo con la descripción textual del caso de uso, puede confirmar o denegar la petición. En caso de denegarla, podrá introducir una nueva fecha. Al acabar la acción, se enviará un correo al orador con las nuevas condiciones.

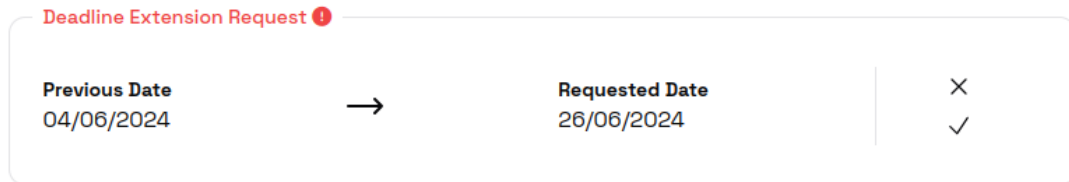


Figura 5.30: Petición de cambio de fecha de un artículo.

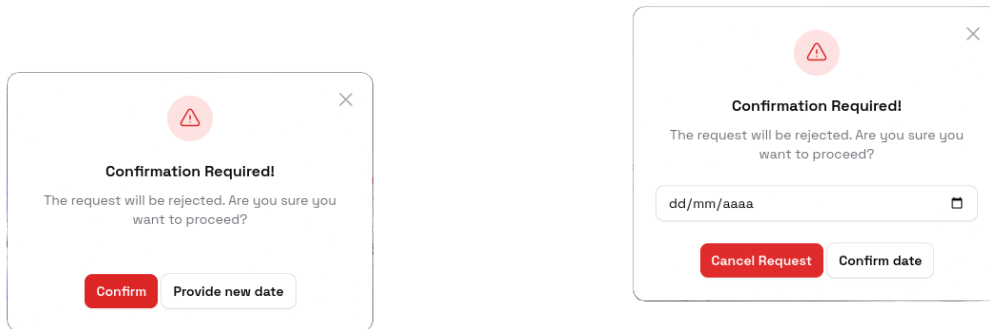


Figura 5.31: Modal de validación de la fecha al denegar una petición y tras activar la acción de dar nueva fecha.

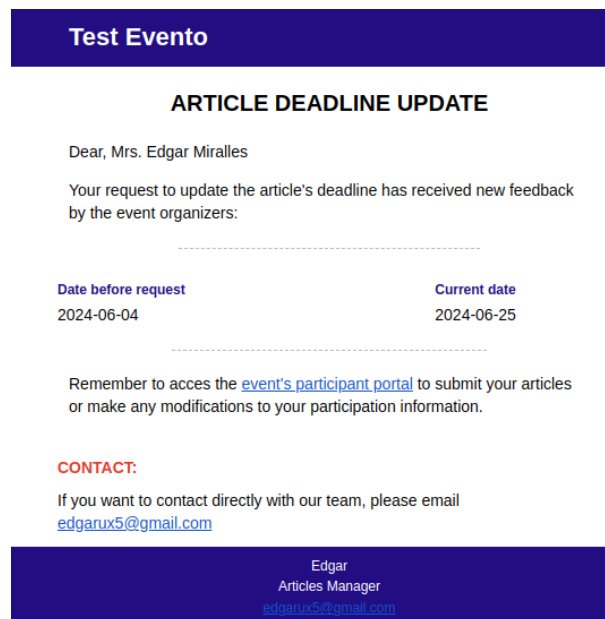


Figura 5.32: Correo de retroalimentación por respuesta a petición de cambio de fecha.

## 5.2.5. Panel de participantes

Pese a ser una aplicación diferente y en un dominio distinto, se ha tratado de utilizar una estética coherente con el panel de administración principal, permitiendo reutilizar una buena cantidad de los componentes de la interfaz.

### Login

El portal de login, cumpliendo con la especificación del Caso de Uso 20. Login, pide un correo electrónico y un código si se encuentra el participante.

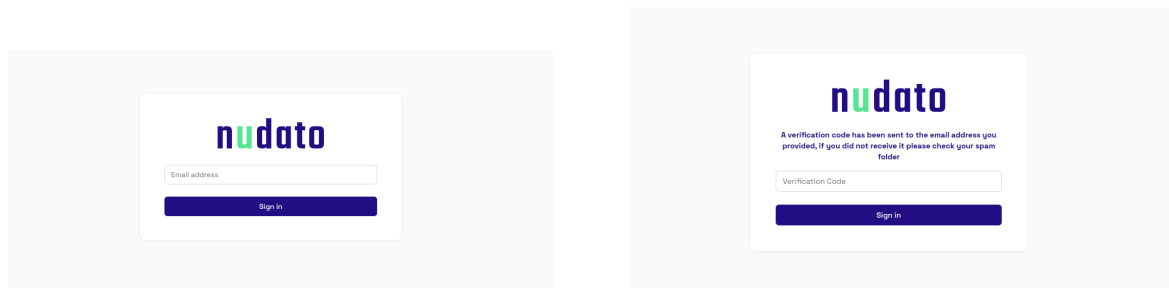


Figura 5.33: Ventana de login.

### Configuración de entorno

El participante, en caso de estar registrado a múltiples organizaciones o eventos, debe seleccionarlos. Esto se puede realizar desde el panel de participación, considerado el principal de la aplicación, o desde un desplegable ubicado en la esquina superior derecha de la pantalla.

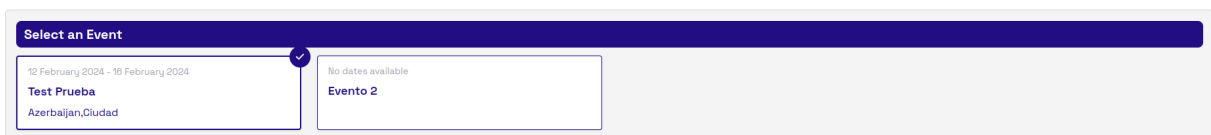


Figura 5.34: Selección de evento desde el panel de participación.

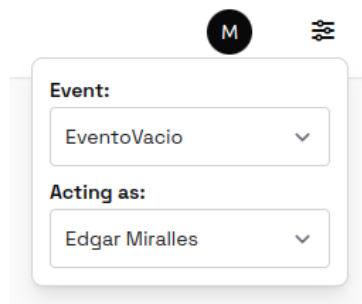


Figura 5.35: Configuración de entorno desde el menú desplegable.

Una vez configurado el entorno del participante, este puede acceder al resto de los paneles.

## Panel de participación

Siguiendo la descripción del Caso de Uso 25. VerParticipación, en el panel se muestran los datos de participación de un orador al evento: estado, actividades en las que participa, vuelos, información del hotel, sesiones a las que atiende y requisitos especiales. Desde este panel se pueden iniciar los casos de uso:

- Caso de uso 26. RechazarActividad: Accesible haciendo clic al icono 'X' en cada una de las actividades.
- Caso de uso 27. ModificarParticipación: Accesible haciendo clic al botón de editar en cualquiera de los campos que dispongan del botón. Las cartas editables funcionan como las que se pueden encontrar en el Caso de Uso 07. ModificarInformación. Ver Figura 5.11

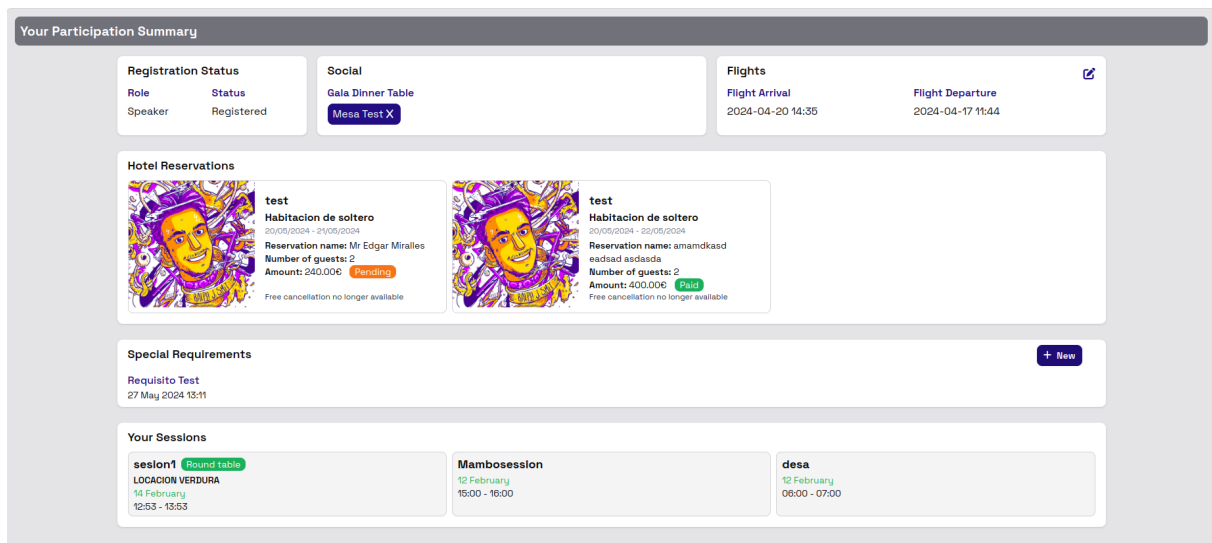


Figura 5.36: Panel de participación.

## Panel de información

Tal y como especifica el Caso de Uso 34. VerInformaciónPersonal, se muestran todos los campos del orador y sus contactos vinculados. Para poder iniciar el Caso de Uso 35. ModificarInformaciónPersonal, el usuario solo necesita dar clic a cualquiera de los campos que quiera editar. Realizar esta acción abrirá un modal desde el que el orador puede generar los cambios que considere, que finalmente enviarán una petición al panel administrativo.

**Personal Information**

<b>Basic Information</b>		<b>Contact Information</b>	
Picture	A profile picture helps to identify you	Email	tester@nudato.com edgar@enviait.com
Title	Mr.	Mobile	dasdas
Name	Edgar Miralles	Phone	fasdas
<b>Biography</b>		<b>Organisation and Role Information</b>	
Biography	adasdasdasdasdaaaaaaadas	Organisation	STAS-21 SL
		Department	Macarronico
		Function	FullStack developer
		Office Address	easdasdasda wd sa sd asdawdsaxcxz cwasd zxcws asdwasd

**Linked Contacts**

Assistants +

Manolo Tester	PortalVacio	Alina
manolito@nudato.com	portalvacio@nudato.com	alina.clorobea@eu.tokeninc.com

Copy Tos +

Figura 5.37: Panel de información.

**Name** ✕

First Name\*

Last Name

**Save**

Figura 5.38: Modal de cambio de información en un campo.

Desde este mismo panel, se puede activar en cualquier momento el *Caso de Uso 36. Modificar Autorización Asistente* accionando el botón de 'X' o 'Check' al lado del nombre del asistente en la carta de contactos vinculados. Esto pedirá una confirmación como describe el caso de uso asociado.

## Panel de sesiones

Siguiendo la especificación del *Caso de Uso 28. VerSesiones*, en este panel se muestran las sesiones y los documentos asociados a estas. El orador solo puede interactuar con sus documentos, añadiendo según los que se encuentren disponibles en su biblioteca de archivos, siguiendo la descripción del *Caso de Uso 29. ModificarDocumentosSesión*.

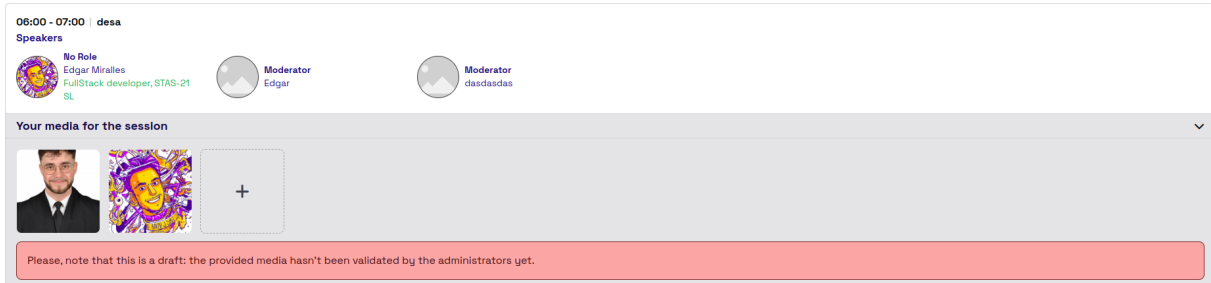


Figura 5.39: Panel de sesiones.

## Panel de documentos

Desde el panel de documentos el orador puede realizar las acciones descritas en el *Caso de Uso 23. VerDocumentos*. Para editarlos según la descripción del *Caso de Uso 29. ModificarDocumentosSesión*, se utiliza el mismo modal ya mostrado en la Figura 5.18.

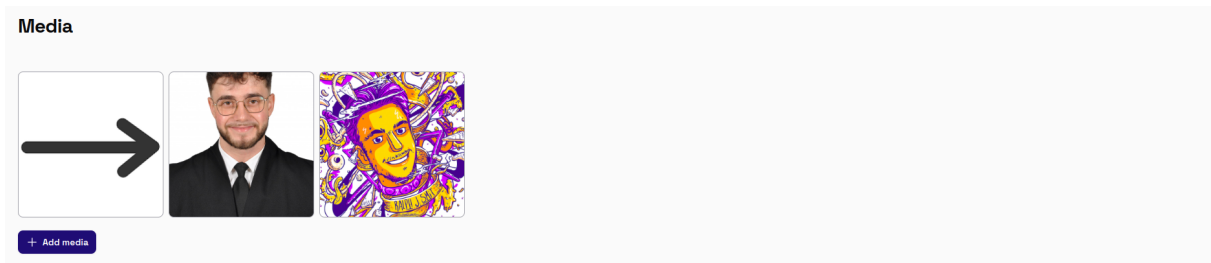


Figura 5.40: Panel de documentos.

## Panel de artículos

Siguiendo con el *Caso de Uso 30. VerArtículosOrador*, se muestran los artículos del orador autenticado ordenados por estado. En este punto el orador puede visualizarlos y activar en cada artículo los casos de uso:

- *Caso de Uso 31. ResponderArtículo*
- *Caso de Uso 32. RechazarArtículo*
- *Caso de Uso 33. PedirAmpliaciónFechaArtículo*

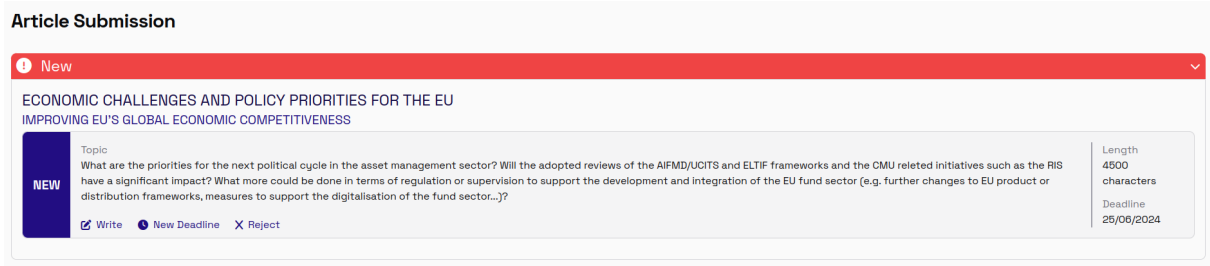


Figura 5.41: Panel de artículos.

En el caso de seleccionar la opción de responder el artículo, el orador deberá rellenar los campos descritos en la *Lista 31.1*. Una vez hecho esto, si el orador envía el artículo recibirá un correo confirmando la acción.

Figura 5.42: Formulario de respuesta de un artículo.

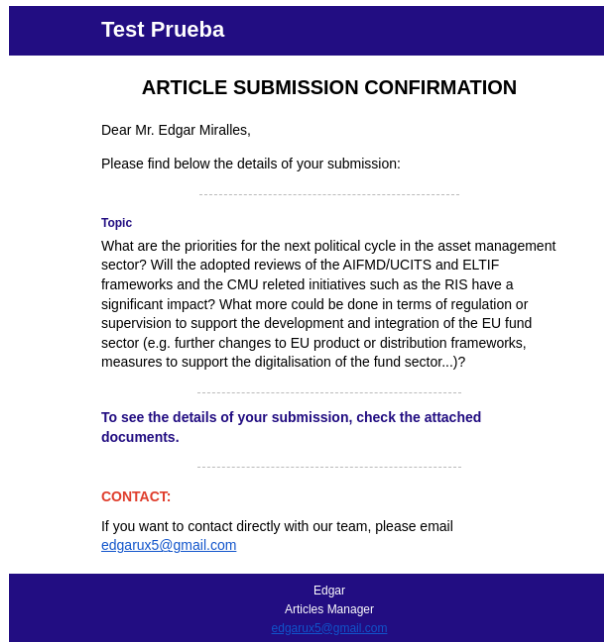


Figura 5.43: Correo de confirmación del envío de un artículo.

Por otro lado, si decide rechazar el artículo, se le presentará un modal pidiendo una razón opcional de rechazo.

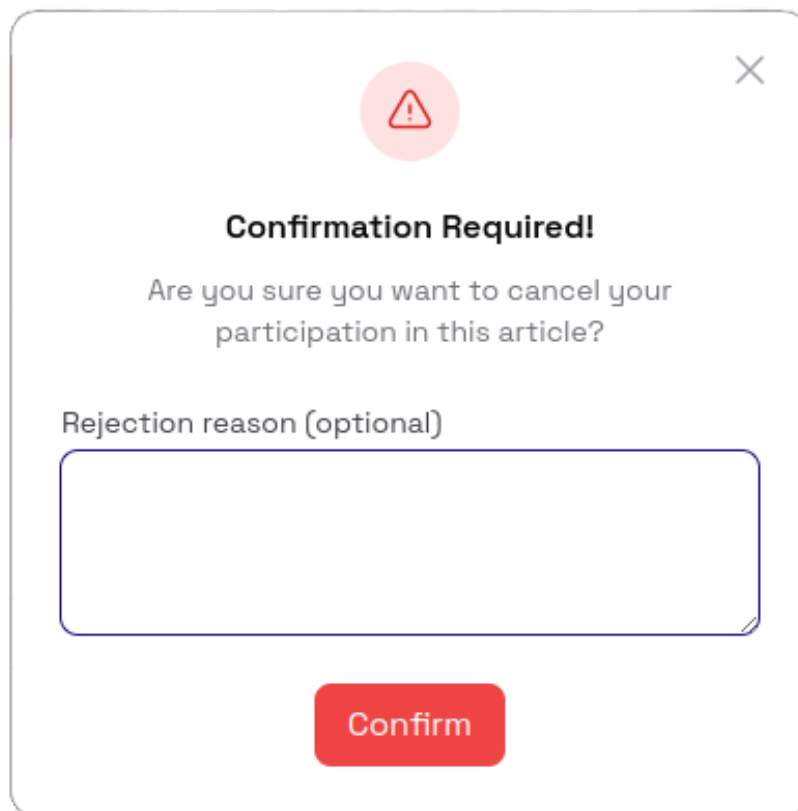
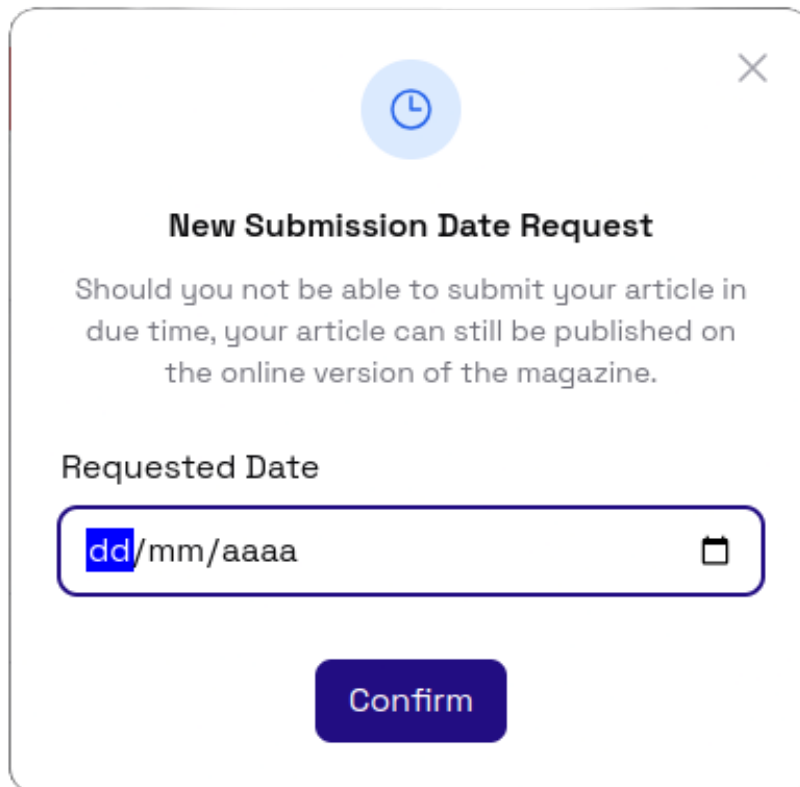


Figura 5.44: Modal de confirmación para el rechazo de un artículo.

Finalmente, si realiza la acción de modificar la fecha de entrega, será provisto de un modal en el que podrá introducir la nueva fecha deseada.



The image shows a modal dialog box with a light blue background and rounded corners. At the top center is a blue circular icon containing a white clock face. In the top right corner is a grey 'X' close button. Below the icon is the title "New Submission Date Request" in bold black text. Underneath is a paragraph of text: "Should you not be able to submit your article in due time, your article can still be published on the online version of the magazine." Below this is the label "Requested Date" in bold black text. Underneath the label is a date input field with a rounded rectangular border. The field contains the text "dd/mm/aaaa" in a dark blue font, with a small calendar icon on the right side. At the bottom center of the modal is a dark blue button with the word "Confirm" in white text.

Figura 5.45: Modal de cambio de fecha de entrega de un artículo.

## 5.3. Diseño de la persistencia

En esta sección se transformarán las entidades descritas en la Figura 4.114 y sus relaciones con tal de poderlas implementar en un sistema gestor de bases de datos. Para ello, se desarrollará un diagrama entidad relación (ERD) en el que se mostrarán las relaciones, sus cardinalidades y las claves foráneas necesarias. Seguidamente, se listarán todas las especificaciones de las tablas a implementar.

### 5.3.1. Diagrama entidad relación

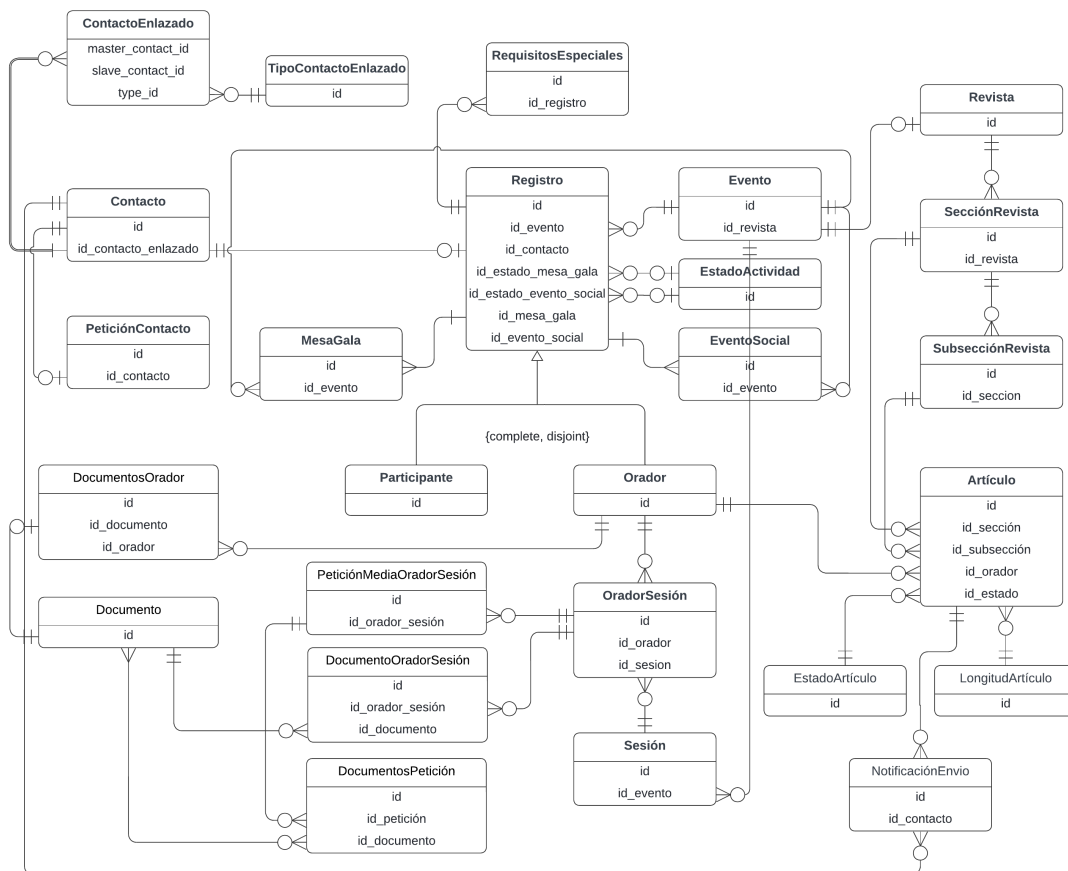


Figura 5.46: Diagrama entidad relación del proyecto.

### 5.3.2. Especificación de las tablas

Las palabras clave de Structured Query Language (SQL) están en mayúscula.

#### Base de datos central

```
CREATE TABLE usuarios
(id NUMERIC(20),
```

```
email VARCHAR(30) NOT NULL,  
login_code VARCHAR(5),  
created_at TIMESTAMP NOT NULL,  
updated_at TIMESTAMP NOT NULL,  
PRIMARY KEY (id)
```

---

## Base de datos de la organización

---

```
CREATE TABLE magazines  
(id NUMERIC(20),  
title VARCHAR(30) NOT NULL,  
PRIMARY KEY (id));  
  
CREATE TABLE magazine_sections  
(id NUMERIC(20),  
name VARCHAR(30) NOT NULL,  
order NUMERIC(3),  
PRIMARY KEY (id));  
  
CREATE TABLE magazine_subsections  
(id NUMERIC(20),  
index VARCHAR(10) NOT NULL,  
name VARCHAR(50) NOT NULL,  
order NUMERIC(3),  
PRIMARY KEY (id));  
  
CREATE TABLE article_statuses  
(id NUMERIC(20),  
name VARCHAR(50) NOT NULL,  
PRIMARY KEY (id));  
  
CREATE TABLE article_notifications  
(id NUMERIC(20),  
date DATETIME NOT NULL,  
FOREIGN KEY (contact_id) REFERENCES contacts(id),  
PRIMARY KEY (id));  
  
CREATE TABLE article_max_lengths  
(id NUMERIC(20),  
length NUMERIC(5),  
PRIMARY KEY (id));  
  
CREATE TABLE articles  
(id NUMERIC(20),  
topic TEXT NOT NULL,  
deadline DATETIME NOT NULL,  
submitted_at DATETIME,  
order NUMERIC(5),  
rejection_reason TEXT,  
key_sentence VARCHAR(200),  
key_sentence_min_length NUMERIC(5) NOT NULL,  
key_sentence_max_length NUMERIC(5) NOT NULL,  
title VARCHAR(200),  
title_min_length NUMERIC(5),  
title_max_length NUMERIC(5) NOT NULL,
```

```

content TEXT,
requested_date DATETIME NOT NULL,
FOREIGN KEY (status_id) REFERENCES article_statuses(id),
FOREIGN KEY (max_length_id) REFERENCES article_max_lengths(id),
FOREIGN KEY (speaker_id) REFERENCES registrations(id),
FOREIGN KEY (section_id) REFERENCES magazine_sections(id),
FOREIGN KEY (subsection_id) REFERENCES magazine_subsections(id),
PRIMARY KEY (id));

CREATE TABLE sessions
(id NUMERIC(20),
name VARCHAR(50) NOT NULL,
start DATETIME NOT NULL,
end DATETIME,
category VARCHAR(50),
FOREIGN KEY (event_id) REFERENCES events(id),
PRIMARY KEY (id));

CREATE TABLE speaker_sessions
(id NUMERIC(20),
FOREIGN KEY (speaker_id) REFERENCES registration(id),
FOREIGN KEY (session_id) REFERENCES sessions(id),
PRIMARY KEY (id));

CREATE TABLE speaker_sessions_media_request
(id NUMERIC(20),
FOREIGN KEY (speaker_sessions_id) REFERENCES speaker_sessions(id),
PRIMARY KEY (id));

CREATE TABLE request_document
(id NUMERIC(20),
FOREIGN KEY (speaker_sessions_media_request_id) REFERENCES
speaker_sessions_media_request(id),
FOREIGN KEY (document_id) REFERENCES document(id),
PRIMARY KEY (id));

CREATE TABLE speaker_sessions_media
(id NUMERIC(20),
FOREIGN KEY (speaker_session_id) REFERENCES speaker_sessions(id),
FOREIGN KEY (document_id) REFERENCES document(id),
PRIMARY KEY (id));

CREATE TABLE contact_request
(id NUMERIC(20),
first_name VARCHAR(50),
last_name VARCHAR(50),
email VARCHAR(30),
email2 VARCHAR(30),
phone VARCHAR(50),
mobile VARCHAR(50),
department VARCHAR(100),
function VARCHAR(100),
address_office VARCHAR(100),
title VARCHAR(6),
biography TEXT,
created_at DATETIME,
update_at DATETIME,
FOREIGN KEY (contact_id) REFERENCES contacts(id),

```

```
FOREIGN KEY (organisation_id) REFERENCES contacts(organisation),
PRIMARY KEY (id));

CREATE TABLE contact_links
(id NUMERIC(20),
authorized BOOLEAN NOT NULL,
role VARCHAR(20) NOT NULL,
FOREIGN KEY (slave_id) REFERENCES contacts(id),
FOREIGN KEY (master_id) REFERENCES contacts(id),
PRIMARY KEY (id));

CREATE TABLE special_requests
(id NUMERIC(20),
request TEXT NOT NULL,
FOREIGN KEY (speaker_id) REFERENCES registrations(id),
PRIMARY KEY (id));

CREATE TABLE speaker_media
(id NUMERIC(20),
FOREIGN KEY (document_id) REFERENCES document(id),
FOREIGN KEY (speaker_id) REFERENCES registrations(id),
PRIMARY KEY (id));

ALTER TABLE events
ADD FOREIGN KEY (magazine_id) REFERENCES magazines(id);

ALTER TABLE registrations
ADD FOREIGN KEY (gala_dinner) REFERENCES gala_dinner_tables(id),
ADD FOREIGN KEY (social_activity) REFERENCES social_activities(id),
ADD FOREIGN KEY (gala_dinner_status_id) REFERENCES
activity_statuses(id),
ADD FOREIGN KEY (social_activity_status_id) REFERENCES
activity_statuses(id);
```

## 6 Implementación

### 6.1. Elección de tecnologías

Un aspecto crucial para implementar una aplicación web es la correcta selección de las tecnologías que permitirán cumplir tanto con los requisitos funcionales como con los no funcionales del proyecto. La elección del framework adecuado puede influir significativamente en el éxito del desarrollo y mantenimiento de la aplicación. Dada la gran cantidad de frameworks disponibles para el desarrollo web, la decisión a menudo se basa en varios factores clave, como la preferencia de lenguaje de programación, la eficiencia y el rendimiento de la tecnología, así como la calidad y la exhaustividad de su documentación. Una selección informada de las herramientas adecuadas no solo facilita el desarrollo eficiente y mantenible, sino que también asegura que la aplicación pueda escalar y adaptarse a las necesidades futuras.

#### 6.1.1. Arquitectura básica

##### Laravel

Laravel [2] es un framework Hypertext Preprocessor (PHP) que sigue el patrón de diseño MVC. Es un framework fuertemente opinado [8] y caracterizado por una sintaxis muy expresiva. Ha sido elegido por varias razones:

- **Ecosistema rico:** Laravel ofrece un ecosistema completo que incluye herramientas integradas para tareas comunes como autenticación, enrutamiento, sesiones, y un sistema de cachés. Esto permite invertir más tiempo de desarrollo en la implementación de la lógica de negocio y cumplir con algunos de los requisitos funcionales de la aplicación.
- **Object-Relational Mapping (ORM) Eloquent:** Laravel utiliza Eloquent, un ORM (Object-Relational Mapping) que proporciona una sintaxis expresiva y similar al resto de Laravel para interactuar con la base de datos. Esto simplifica las operaciones CRUD y asegura que la capa de modelo se integre con la capa de base de datos de forma independiente al SGBD implementado.
- **Motor de plantillas Blade:** Laravel incluye Blade, un motor de plantillas que facilita la creación de vistas dinámicas y reutilizables. Blade permite el uso de plantillas maestras, que se aplican a todas las vistas para dar una estructura consistente.
- **Seguridad:** Laravel proporciona características de seguridad integradas como protección contra CSRF (Cross-Site Request Forgery), encriptación, y saneamiento de entradas. Esto

asegura que la aplicación cumpla con los requisitos de seguridad descritos con anterioridad.

- **Sistema de colas:** Laravel facilita la implementación de colas para tareas que pueden ser procesadas en segundo plano, mejorando el rendimiento y la capacidad de respuesta de la aplicación.
- **Envío de correos:** Laravel ofrece un sistema de correos sencillo y potente que soporta diferentes servicios de correo como SMTP o Amazon SES. Esto permite el envío fácil y eficiente de correos electrónicos. Esto, junto al sistema de colas, crea un entorno perfecto para el envío de correos a los oradores, las confirmaciones de artículos, etc.

## Livewire

Para mejorar la reactividad y la interactividad de la aplicación, se ha decidido utilizar Livewire junto con Laravel. Livewire [9] es un framework que facilita la creación de interfaces dinámicas sin salir del framework de Laravel y su sistema de plantillado, ofreciendo las siguientes ventajas:

- **Reactividad:** Livewire permite crear componentes interactivos y reactivos sin necesidad de escribir código JavaScript.
- **Integración sencilla:** Al estar diseñado específicamente para Laravel, Livewire se integra fácilmente con Blade y otros componentes de Laravel, simplificando el desarrollo y mantenimiento de la aplicación.
- **Manejo del estado:** Livewire facilita el manejo del estado de los componentes, asegurando que la interfaz de usuario se actualice correctamente en respuesta a las acciones del usuario. Toda la reactividad se consigue mediante llamadas al servidor de un tamaño ligero con el objetivo de actualizar solamente aquella parte de la plantilla que debe ser actualizada. Esto mejora la experiencia del usuario al permitir interacciones en tiempo real con la aplicación sin que el desarrollador deba aprender un complejo framework de JavaScript.

## Filament

Filament [3] es una herramienta de administración para Laravel construida mediante componentes de Livewire que facilita la construcción de paneles de administración y interfaces de usuario complejas basadas en componentes. Ha sido elegido para este proyecto por las siguientes razones:

- **Reutilización de componentes:** Filament permite reutilizar componentes entre el panel de administración y el panel de participantes del evento. Esto reduce el tiempo de desarrollo y asegura una experiencia de usuario coherente en ambas interfaces.
- **Interfaz de Usuario amigable:** Filament ofrece una interfaz de usuario moderna y amigable que es fácil de personalizar. Esto asegura que los administradores y los participantes del evento puedan interactuar con la aplicación de manera intuitiva, cumpliendo por defecto muchos de los requisitos de usabilidad.

- **Integración con Laravel:** Este es el punto clave de la elección de Filament. La herramienta se ha construido para ser utilizada junto a Laravel. Esta aprovecha todas las ventajas del framework subyacente, incluyendo la seguridad, el ORM Eloquent, y el sistema de enrutamiento.
- **Uso de Livewire:** Filament utiliza Livewire para manejar la reactividad y la interactividad de los componentes. Esto permite extender los componentes según las necesidades del desarrollo o crear nuevos.

## TailwindCSS

Para el diseño y la estilización de la interfaz de usuario, se ha optado por utilizar TailwindCSS. TailwindCSS [10] es un framework de CSS utility-first que proporciona un conjunto de clases a utilizar directamente en HTML. Ha sido elegido principalmente por la Integración con Laravel y Filament. TailwindCSS se integra perfectamente con Filament, cuyos componentes se construyen a partir de Tailwind y con las plantillas de blade que otorga Laravel.

## Justificación de la elección

La elección de Laravel, Filament, Livewire y Tailwind permiten implementar el patrón MVC de una manera rápida y eficiente gracias a estar fuertemente opinado. Esto permitirá dedicar más tiempo a la implementación de la amplia lógica de negocio planteada en el proyecto. El manejo eficiente de sesiones, la facilidad para implementar sistemas de colas y correos, y sus características de seguridad hacen que sea una opción teniendo en cuenta los casos de uso previstos y los requisitos no funcionales de escalabilidad y seguridad.

### 6.1.2. Implementación de la multitenencia

La multitenencia ha resultado ser uno de los factores más problemáticos en la implementación:

#### El problema de la identificación

El principal problema que deriva de la aplicación planteada es la correcta identificación del tenant. La misma aplicación debe ser capaz de conectarse a múltiples bases de datos sin modificar el código referente a la lógica de negocio.

**Identificación del tenant en el panel administrativo** En el panel de administración, una implementación sencilla de la identificación sería el uso del subdominio de la web. Por ejemplo, el subdominio `test.nudato.com` identifica al tenant llamado "test". Esta estrategia de subdominios permite una clara separación entre los diferentes clientes y facilita la gestión de sus datos específicos.

**Identificación del tenant en el panel de participante** En el caso del panel de participantes se requería una identificación dinámica de los tenants según la organización que el usuario escogiera. Para llevar esto a cabo, se debe plantear una solución alternativa a los subdominios, ya que la acción de cambio de tenant debe ser transparente para el participante del evento que está usando el portal.

**Justificación del uso de Tenancy for Laravel** El paquete Tenancy for Laravel ha sido seleccionado por las siguientes razones:

- **Facilidad de implementación:** Tenancy for Laravel simplifica la configuración y gestión de múltiples bases de datos, permitiendo definir fácilmente los tenants y sus correspondientes bases de datos, ajustándose así al requisito principal de seguridad de la aplicación y la identificación de los organizadores. Por defecto, el paquete ofrece los siguientes métodos de identificación:
  - Identificación por dominio (tenant.com)
  - Identificación por subdominio (tenant.nudato.com)
  - Identificación por dominio O subdominio (los dos previos)
  - Identificación por ruta (yoursaas.com/tenant/dashboard)

Cada uno de estos métodos implica el uso de un middleware diferente para inicializar el tenant.

- **Escalabilidad y mantenimiento:** Este paquete facilita la escalabilidad y el mantenimiento de la aplicación, permitiendo realizar actualizaciones y backups de manera aislada para cada cliente.
- **Flexibilidad:** Tenancy for Laravel ofrece una gran flexibilidad para personalizar el comportamiento de la multitenencia según las necesidades específicas del proyecto, proporcionando una API que permite la inicialización manual de los tenants según su identificador.

## Solución al problema de la identificación

Conociendo el paquete a implementar, se pueden establecer estrategias para identificar al tenant en cada uno de los casos requeridos:

**Identificación del tenant en el panel administrativo** En el panel de administración, se ha decidido usar el middleware del paquete que identifica al tenant según el subdominio. Esto se ajusta perfectamente a la idea inicial.

**Identificación del tenant en el panel de participante** A causa de la identificación dinámica, se ha requerido un método que inicialice el tenant en función de la ID, guardada en sesión una vez el usuario escoge el organizador del que ver los eventos. Debido a la limitada cantidad de métodos disponibles en el paquete de multitenencia, se ha realizado un middleware personalizado para cumplir este propósito. El middleware en cuestión verifica la presencia del ID del tenant en sesión, y en caso de estar ahí se inicializa mediante la API del paquete.

```

class InitializeTenancyBySession
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request):
     *       (\Symfony\Component\HttpFoundation\Response) $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        $tenantId = session('selected_tenant_id');
        if ($tenantId) {
            $tenant = Tenant::find($tenantId);
            /** @var ?Domain $domain */
            $domain = $tenant->domains->first();
            config(['filesystems.disks.nudato-file-manager.prefix' =>
                $domain->domain . "-file-manager"]);
            tenancy()->initialize($tenant);
        }
        return $next($request);
    }
}

```

### 6.1.3. Despliegue *serverless*

Debido a la irregularidad en la demanda del aplicativo, uno de los requisitos más fundamentales era la escalabilidad. Se requería una herramienta que de forma independiente levantara servidores en función de las necesidades, un despliegue *serverless*. Una de las herramientas del ecosistema de Laravel más utilizadas con este propósito es **Laravel Vapor** [11].

Laravel Vapor es una plataforma *serverless* basada en Amazon Web Services (AWS) [12] que maximiza la escalabilidad y ajusta los costos según el uso, ofreciendo una solución FaaS (Function as a Service) que es ideal para aplicaciones con demandas fluctuantes.

#### Escalabilidad automática

Laravel Vapor aprovecha AWS Lambda para servir las solicitudes HTTP de manera *serverless*. AWS Lambda es un servicio FaaS que permite ejecutar código en respuesta a eventos sin tener que aprovisionar o gestionar servidores. Esta capacidad de escalar automáticamente en respuesta a los aumentos de tráfico es crucial para manejar los picos de demanda en los días cercanos a los eventos. AWS Lambda ajusta automáticamente el número de instancias en función del número de solicitudes, asegurando que la aplicación siempre tenga los recursos necesarios sin necesidad de intervención manual, lo que garantiza un rendimiento óptimo.

#### Coste efectivo

Uno de los principales beneficios de utilizar Laravel Vapor es su modelo de pago por uso. Con AWS Lambda, se paga únicamente por el tiempo de ejecución de las funciones y el número de

solicitudes, lo que reduce significativamente los costos durante los períodos de baja actividad. Este modelo es ideal para aplicaciones con patrones de uso irregulares, como una aplicación de gestión de eventos, donde los picos de uso se concentran en días específicos y el resto del tiempo el uso es mínimo.

### **Servicios en la nube: SES, SQS, S3**

En la implementación de la arquitectura serverless, se utilizan varios servicios gestionados en la nube para cumplir con los diferentes requisitos de la aplicación. Estos servicios proporcionan funcionalidades como el envío de correos electrónicos, manejo de colas de procesos y almacenamiento de documentos. A continuación, se describe el uso de cada uno de estos servicios.

**Cloud Email Service (SES)** El servicio de correo electrónico en la nube es fundamental para las acciones de comunicación dentro de la aplicación. Este servicio se utiliza para:

- Realizar la acción de enviar correos manuales a los participantes...
- Enviar confirmaciones de peticiones y artículos, asegurando que los usuarios reciban feedback inmediato sobre sus acciones dentro de la plataforma.
- Manejar correos electrónicos con archivos adjuntos pesados, garantizando que incluso los documentos grandes puedan ser enviados sin problemas. Estos a veces provocan *timeouts* al tardar en procesarse, por lo que debe gestionarse junto a una cola de procesos, capaz de reintentar trabajos.

**Cloud Queue Service (SQS)** El servicio de manejo de colas en la nube se utiliza principalmente para gestionar trabajos relacionados con el envío de correos electrónicos. Este servicio es crucial para:

- Garantizar la fiabilidad en el envío de correos mediante la capacidad de reintentar trabajos fallidos, lo que mejora la resiliencia del sistema.
- Distribuir la carga de trabajo, permitiendo que las tareas intensivas en recursos se procesen sin afectar el rendimiento general de la aplicación.

**Cloud Object Storage (S3)** El servicio de almacenamiento de objetos en la nube es utilizado para almacenar y gestionar los documentos de los oradores. Este servicio proporciona:

- Almacenamiento seguro y escalable para todos los documentos relacionados con los oradores, incluyendo presentaciones, imágenes y otros materiales relevantes.
- Acceso eficiente y rápido a los documentos, facilitando que los oradores y organizadores puedan subir y descargar archivos según sea necesario.
- Alta disponibilidad y durabilidad de los datos, asegurando que los documentos estén siempre accesibles y protegidos contra pérdidas.

## Mantenimiento reducido

Al delegar la gestión de la infraestructura a AWS, Laravel Vapor permite al equipo de desarrollo centrarse en la lógica de negocio en lugar de en la administración de servidores. AWS maneja aspectos críticos como el aprovisionamiento de servidores, la escalabilidad, la tolerancia a fallos y el balanceo de carga. Esto no solo reduce la carga de mantenimiento, sino que también mejora la confiabilidad y disponibilidad del sistema. Con AWS Lambda, no es necesario preocuparse por la gestión del servidor subyacente, lo que simplifica enormemente el despliegue y la operación de la aplicación.

## Integración con el ecosistema de Laravel

Laravel Vapor se integra perfectamente con el ecosistema de Laravel, permitiendo despliegues rápidos y simplificados directamente desde el framework. Esto incluye soporte para todas las características de Laravel, como migraciones de bases de datos, colas, almacenamiento en caché, y más. La integración fluida asegura que las aplicaciones Laravel puedan aprovechar todas las ventajas del despliegue *serverless* sin necesidad de configuraciones complejas adicionales.

## 6.2. Desafíos en el desarrollo

A continuación, se presentan algunos desafíos anecdóticos que ha planteado el proyecto y cómo se han solucionado.

### 6.2.1. Metodologías ágiles y configuración del entorno

Una de las premisas del desarrollo basado en metodologías ágiles es la capacidad de reacción frente a cambios en los requisitos del proyecto o feedback por parte del cliente [13]. Durante el proyecto, la mayoría de los requisitos funcionales se han mantenido estáticos, con cambios principalmente en la interfaz de usuario. El único cambio real en los requisitos ha sido en referencia al *Caso de Uso 21*. Originalmente, se había planteado que el propio usuario escogiera entre los múltiples organizadores en los que tenía algún registro para luego elegir uno de los eventos del organizador. Esta decisión se tomó para evitar mantener el listado de eventos de todos los organizadores, lo que implicaría consultas a múltiples bases de datos. Tras algunas pruebas cerradas con usuarios, se concluyó que esta funcionalidad no podía ser ejecutada de forma clara, ya que muchas veces no conocen a los organizadores del evento al que asisten. Esto llevó a la necesidad de mantener la lista de eventos previamente mencionada, permitiendo que el usuario solo escogiera entre todos los eventos (el proceso de conexión a la base de datos de la organización se realizaría de manera transparente). Uno de los componentes de la interfaz es el configurador de entornos, ubicado en la barra de navegación 5.35. En circunstancias normales, para cumplir con el nuevo requisito, se debería generar la consulta a todas las bases de datos con cada carga de una nueva ventana. Por la complejidad de la consulta, se optó por realizarla en el momento de autenticación del usuario durante el login y seguidamente guardar el resultado en una caché de Redis. Así, el componente puede verificar si los datos requeridos ya se encuentran en esta estructura para no realizar la petición. Los datos que se necesitan guardar en la caché

son una lista de eventos por cada usuario que se conecta al portal de participantes. Cada objeto de la lista contiene la información que se debe mostrar del evento, el ID del *tenant* para poder conectarse a su base de datos y el rol que toma el usuario en el evento.

### **Problemas derivados del uso de la caché**

Esta solución a la repetición de consultas extremadamente pesadas lleva a su vez un nuevo desafío. La consulta se guarda en caché 1 día desde que se introduce. Se ha utilizado un tiempo bastante alto, ya que las circunstancias en las que debería cambiar son muy reducidas y se producen con poca frecuencia:

- Creación de un nuevo registro con el correo del usuario.
- Eliminación de un registro con el correo del usuario.
- Autorización por parte de un participante a un asistente con el correo del usuario.

Pese a ser situaciones que ocurren con poca frecuencia, podrían producir una falta de sincronización entre el estado de la caché y el de las bases de datos de las organizaciones. Con el objetivo en mente de paliar este problema, se ha desarrollado un sistema de eventos mediante el **patrón Observer** para actualizar la caché en cada una de las operaciones listadas.

Al existir el sistema de roles, un mismo usuario puede ser de 3 tipos en un mismo evento:

1. Participante (*participant*)
2. Asistente (*assistant*)
3. Participante y asistente (*participant\_assistant*)

Si no existieran los roles, el patrón observer solo tendría que eliminar o añadir datos en la caché en función de si se detecta una operación de creado o de borrado de un registro. Al haber roles, la creación o borrado de un registro también debe iterar los asistentes de ese registro para poder actualizar el rol cacheado del usuario del asistente en el evento. Si estos casos no se tuvieran en cuenta, se obtendría como resultado una desincronización, que duraría hasta la expiración de la caché, generando así un modelo de consistencia eventual entre lo que hacen los organizadores en el panel de administrador y lo que ve reflejado el participante.

Por los requisitos del proyecto, este comportamiento no era aceptable porque en las propias funciones del observer se ha implementado la actualización de los datos de la caché.

### **Problemas derivados de la persistencia de datos en sesión**

Tras configurar el entorno, el usuario tiene en sesión el tenant, el evento y el orador que ha seleccionado (en caso de ser un asistente). Esta persistencia es necesaria para mantener el estado entre las distintas ventanas del aplicativo y asegurar la coherencia de los datos sin que el usuario tenga que seleccionar el evento en cada una de estas. Esta situación puede generar un problema parecido a la desincronización con la caché: si un organizador borra un registro de un evento y el usuario ya estaba conectado a ese evento, daría un error al intentar cargar el registro inexistente. Este caso es más grave que la problemática anterior debido a que se genera un error que impide el funcionamiento normal de la aplicación. Para solucionarlo, cada vez que

se recupera un dato de sesión, se comprueba primero que siga existiendo el modelo en la base de datos

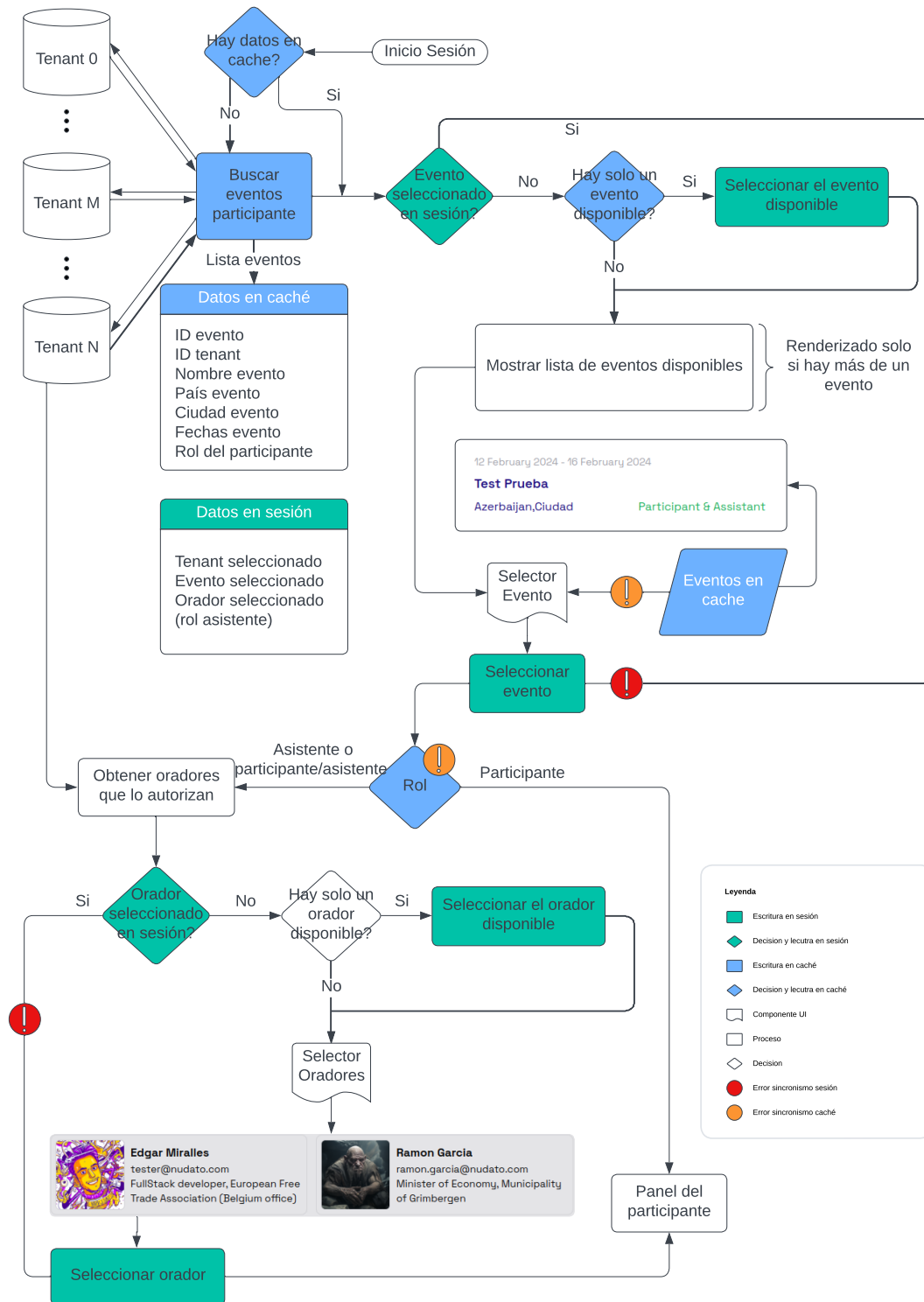


Figura 6.1: Diagrama de flujo de la configuración de entorno con los posibles errores representados.

En la figura se puede observar el flujo de configuración del entorno, incluyendo el cacheo de

datos. Las exclamaciones naranjas representan los problemas derivados de la desincronización con la caché y los rojos los derivados del problema con los datos en sesión.

### 6.2.2. Eficiencia de peticiones en múltiples bases de datos

Un problema recurrente en el desarrollo del aplicativo para participantes ha sido la necesidad de ejecutar consultas en diferentes bases de datos. Para minimizar los tiempos de espera, se han aplicado índices sobre las columnas más usadas en este contexto. Esto indica a la base de datos la necesidad de mantener una estructura de datos optimizada según las columnas especificadas en el índice.

Además, se ha implementado un plan de monitorización sobre estas peticiones. En caso de encontrar un cuello de botella causado por esto, se replicarían los datos de registro en la base de datos central para eliminar la necesidad de iterar sobre múltiples bases de datos.

### 6.2.3. Gestión de los documentos de los oradores

Para evitar mantener datos duplicados y facilitar la conexión entre las dos aplicaciones, se decidió integrar la misma librería de documentos utilizada en el módulo administrativo. Esto generaba el riesgo de que los oradores pudieran acceder a documentos de otras partes de la librería e incluso a los documentos privados de otros oradores mediante los elementos de navegación que ofrecía la interfaz. Para restringir este comportamiento, se extendió la funcionalidad del plug-in original como solución temporal y se contactó con el creador de la librería para facilitar la implementación de la solución en el repositorio original.

Otro problema relacionado con la gestión de los documentos ha sido la necesidad de crear automáticamente carpetas para los oradores del evento para mantener un orden y aislar estos documentos. Para abordar esta necesidad, se recurrió a un patrón de diseño ampliamente utilizado, el **patrón Observer**. Mediante este patrón, cada vez que se creaba un nuevo registro con el rol de speaker o se modificaba uno existente, se creaba también su carpeta en la librería.

## 7 Evaluación

Para asegurar el correcto funcionamiento del código al integrarlo con la rama principal y en futuras actualizaciones, se ha desarrollado un conjunto de pruebas automatizadas, las cuales se han integrado con el flujo de integración y despliegue de la aplicación. Estas pruebas se han organizado según los componentes de la interfaz y los casos de uso descritos. En particular, se han realizado pruebas de renderizado de los componentes para garantizar que se muestren en pantalla sin errores con cualquier tipo de dato de entrada, así como pruebas de funcionalidad para verificar su correcto desempeño. En los tests de renderizado se ha hecho especial énfasis en aquellas propiedades que se muestran en pantalla y que pueden generar errores en caso de estar vacías. Por ejemplo, si se quiere mostrar el nombre de la organización de un orador, pero el orador no tiene organización, un mal acceso a la propiedad en el código podría generar un error. Para asegurar estos casos, se han creado modelos de prueba con solo los datos obligatorios.

### 7.1. Listado de oradores

<b>Tests renderizado</b>		
Oradores con todos los atributos	La tabla se renderiza sin ningún error y ningún orden ni filtro aplicados por defecto	Pasa
Oradores con solo atributos obligatorios	La tabla se renderiza sin ningún error y ningún orden ni filtro aplicados por defecto	Pasa
Hay petición de validación	Se muestra en el contacto la etiqueta de validación pendiente	Pasa

Cuadro 7.1: Tests de renderizado sobre el componente de listado de oradores.

<b>Caso de uso 01. VerOradores</b>		
Se acciona el botón de crear de un orador	Se muestra el modal de creación del orador (Inicio del caso de uso 02)	Pasa
Se acciona el botón de editar de un orador	Se muestra el modal de edición con los datos del orador seleccionado (Inicio del caso de uso 03)	Pasa
Se acciona el botón de borrar de un orador	Se muestra el modal de confirmación (Inicio del caso de uso 04)	Pasa
Se inicia la acción en masa de enviar correo	Se muestra el modal de envío de correo (Inicio del caso de uso 19)	Pasa
Se inicia la acción en masa de borrar oradores	Se muestra el modal de confirmación (Inicio del caso de uso 04)	Pasa

Cuadro 7.2: Tests del caso de uso 01. VerOradores.

<b>Caso de uso 02. CrearOrador</b>		
No se introduce alguno de los campos obligatorios (contacto, estado)	Se muestra un mensaje de error en el campo que falta	Pasa
Se crea un orador sin campos no relacionales	Se crea una fila en la tabla de <i>registrations</i> con los datos introducidos	Pasa
Se añade un requisito especial vacío	Se muestra un mensaje de error indicando que no puede haber requisitos vacíos	Pasa
Se crea un orador con requisitos especiales	Se crea un orador en la tabla de registros con los datos introducidos y un requisito especial relacionado con el orador en la tabla de <i>special_requests</i>	Pasa
Se crea un orador con sesiones añadidas	Se crea un orador en la tabla de registros con los datos introducidos y un requisito especial relacionado con el orador en la tabla de <i>special_requests</i>	Pasa

Cuadro 7.3: Tests del caso de uso 02. CrearOrador.

<b>Caso de uso 03. ModificarOrador</b>		
Se edita algún campo no relacional del orador	Se modifica la fila del orador en la base de datos y se actualiza la vista	Pasa
Se añade alguna sesión en el orador	Se crea una fila en la tabla intermedia <i>speaker_sessions</i>	Pasa
Se crea algún requisito especial para el orador	Se crea una fila en la tabla <i>special_requests</i> y se relaciona con el orador	Pasa
Se borra algún campo obligatorio (estado, contacto)	Se muestra un mensaje de error en el campo que falta	Pasa

Cuadro 7.4: Tests del caso de uso 03. ModificarOrador.

<b>Caso de uso 04. BorrarOrador</b>		
Se borra un orador sin sesiones ni requisitos especiales	Se borra la fila del orador en la tabla de registro	Pasa
Se borra un orador asignado a sesiones	Se borran las filas que estén relacionadas con el orador en la tabla de <i>speaker_sessions</i> en cascada.	Pasa
Se borra un orador con requisitos especiales	Se borran las filas relacionadas con el orador en la tabla <i>special_requests</i>	Pasa

Cuadro 7.5: Tests del caso de uso 04. BorrarOrador.

<b>Caso de uso 19. EnviarCorreo</b>		
Se envía un correo con plantilla	Se envía el correo a los oradores seleccionados	Pasa
Se envía un correo sin plantilla	Se envía el correo a los oradores seleccionados	Pasa
Se envía un correo con cc o bcc	Se envía el correo a los oradores seleccionados y a correos especificados en los campos de bcc y cc	Pasa

Cuadro 7.6: Tests del caso de uso 19. BorrarOrador.

## 7.2. Detalles del orador

<b>Tests renderizado</b>		
Orador con todos los atributos	Se renderizan todos los componentes de la página exitosamente	Pasa
Orador con solo atributos obligatorios	Se renderizan todos los componentes de la página exitosamente	Pasa
Hay petición de validación de un orador con todos los atributos	Se inicia el caso de uso 06. ValidarPeticones exitosamente	Pasa
Hay petición de validación de un orador con solo los atributos obligatorios	Se inicia el caso de uso 06. ValidarPeticones exitosamente	Pasa

Cuadro 7.7: Tests de renderizado sobre la página de detalles del orador.

## 7.2.1. Ventana de información

<b>Caso de uso 05. VerDetallesOrador</b>		
Se acciona el botón de editar en cualquier carta de información	La carta cambia al modo edición (inicio del caso de uso 07)	Pasa
Se acciona el botón de añadir un contacto vinculado	La carta cambia al modo edición (inicio del caso de uso 08)	Pasa

Cuadro 7.8: Tests del caso de uso 05. VerDetallesOrador.

<b>Caso de uso 06. ValidarPeticiones</b>		
Acepta un cambio de información de contacto	La fila del contacto se modifica en la tabla <i>contacts</i> y se borra la petición de la tabla <i>contact_request</i>	Pasa
Acepta un cambio en los documentos de una sesión	Se modifica la tabla de <i>speaker_session_documents</i> creando las filas necesarias. Se borra la petición de la tabla <i>speaker_sessions_media_request</i> . Se borran en cascada las filas relacionadas en la tabla <i>request_media</i>	Pasa

Cuadro 7.9: Tests del caso de uso 06. ValidarPeticiones.

<b>Caso de uso 07. ModificarInformación</b>		
Se intenta modificar la información del orador dejando un campo obligatorio vacío (nombre orador, correo, estado)	Se muestra un mensaje de error en el campo correspondiente	Pasa
Se modifica la información de contacto	Se modifica la fila del contacto al que se asocia el orador en la tabla <i>contacts</i>	Pasa
Se modifica la información de participación	Se modifica la fila del orador en la tabla <i>registrations</i>	Pasa
Se intenta añadir un requisito especial vacío	Se muestra un mensaje de error comunicando que el requisito no puede ser vacío	Pasa
Se modifica la información de participación y se añade un requisito especial	Se modifica la fila del orador en la tabla <i>registrations</i> y se crea una nueva fila en la tabla de <i>special_requests</i>	Pasa

Cuadro 7.10: Tests del caso de uso 07. ModificarInformación.

<b>Caso de uso 08. AñadirContactoVinculado</b>		
Se busca un contacto inexistente	Se muestra un mensaje de contacto inexistente	Pasa
Se busca un contacto ya vinculado	No se muestra ningún contacto	Pasa
Se añade un asistente autorizado	Se crea una nueva fila en la tabla <i>contact_links</i> con el rol de asistente, el contacto del orador y el contacto seleccionado y con la columna de autorizado en cierto	Pasa
Se añade un asistente no autorizado	Se crea una nueva fila en la tabla <i>contact_links</i> con el rol de asistente, el contacto del orador y el contacto seleccionado y con la columna de autorizado en falso	Pasa
Se añade un vínculo del tipo <i>copia</i>	Se crea una nueva fila en la tabla <i>contact_links</i> con el rol de copia, el contacto del orador y el contacto seleccionado	Pasa

Cuadro 7.11: Tests del caso de uso 08. AñadirContactoVinculado.

## 7.2.2. Ventana de documentos y sesiones

<b>Tests renderizado</b>		
Sesiones con todos los atributos	Se renderiza la página correctamente	Pasa
Sesiones con solo los atributos obligatorios	Se renderiza la página correctamente	Pasa
Sesiones con oradores que tienen todos los atributos completos	Se renderiza la página correctamente	Pasa
Sesiones con oradores que tienen solo los atributos obligatorios	Se renderiza la página correctamente	Pasa
Sesiones con documentos	Se renderiza la página correctamente	Pasa
Sesiones sin documentos	Se renderiza la página correctamente	Pasa

Cuadro 7.12: Tests de renderizado sobre la página de detalles del orador, ventana de sesiones y documentos.

<b>Caso de uso 09. BorrarSesion</b>		
Sesión sin documentos	Se elimina al orador de la sesión, borrando la fila correspondiente en la tabla <i>speaker_sessions</i>	Pasa
Sesión con documentos	Se elimina al orador de la sesión, borrando la fila correspondiente en la tabla <i>speaker_sessions</i> y se eliminan los documentos de la sesión, eliminando las filas correspondientes en la tabla <i>speaker_sessions_media</i>	Pasa

Cuadro 7.13: Tests del caso de uso 09. BorrarSesion.

<b>Caso de uso 10. AñadirSesion</b>		
<b>Caso añadir existente</b>		
Se busca una sesión incompatible con el orador (horario ya ocupado por otra sesión)	No se muestra la sesión en el buscador	Pasa
Se busca una sesión disponible	Se muestra la sesión buscada	Pasa
Se busca una sesión inexistente	Se muestra un mensaje de sesión inexistente	Pasa
Se busca y añade una sesión	Se modifica la tabla de <i>speaker_sessions</i> , añadiendo una nueva fila que representa la relación	Pasa
<b>Caso crear</b>		
Se intenta crear una sesión con campos incompletos	Se muestra un mensaje de error en los campos vacíos	Pasa
Se intenta crear una sesión con un horario incompatible con un orador asignado	Se muestra un mensaje de error indicando la razón	Pasa
Se crea una sesión con oradores	crea una nueva fila en la tabla <i>sessions</i> y tantas filas como oradores en la tabla <i>speaker_sessions</i>	Pasa

Cuadro 7.14: Tests del caso de uso 10. AñadirSesion.

<b>Caso de uso 11. ModificarDocumentos</b>		
Se añade un documento a la carpeta general del orador	Se crea una nueva fila en la tabla <i>speaker_media</i> para representar la relación	Pasa
Se borra un documento de la carpeta general del orador	Se borra la fila correspondiente en la tabla <i>speaker_media</i> que representaba la relación	Pasa
Se añade un documento en una sesión	Se crea una nueva fila en la tabla <i>speaker_sessions_media</i> que representa la relación	Pasa
Se borra un documento de una sesión	Se borra la fila en la tabla <i>speaker_sessions_media</i> que representaba la relación	Pasa

Cuadro 7.15: Tests del caso de uso 11. ModificarDocumentos.

### 7.3. Revista

<b>Creación de Revista</b>		
No se añade un título	No se muestra ningún campo para empezar a crear secciones	Pasa
Se intenta crear revista con campo de nombre de sección vacío	Se muestra un mensaje de error en el campo afectado	Pasa
Se intenta crear revista con el nombre o índice de una subsección vacíos	Se muestra un mensaje de error en el/los campos afectado	Pasa
Se crea una revista con los campos completos	Se crea la fila de la tabla <i>magazines</i> con el nombre de la revista y las filas correspondientes de secciones y subsecciones en las tablas <i>magazine_sections</i> y <i>magazine_subsections</i>	Pasa

Cuadro 7.16: Tests de la creación de la revista.

### 7.4. Listado de artículos

<b>Tests renderizado</b>		
Artículos con todos los atributos	La tabla se renderiza sin ningún error y ningún orden ni filtro aplicados por defecto	Pasa
Artículos con solo atributos obligatorios	La tabla se renderiza sin ningún error y ningún orden ni filtro aplicados por defecto	Pasa
Hay petición de validación	Se muestra en el contacto la etiqueta de validación pendiente	Pasa

Cuadro 7.17: Tests de renderizado sobre el componente de listado de artículos.

<b>Caso de uso 12. VerArtículos</b>		
Se acciona el botón de crear de un artículo	Se muestra el modal de creación del artículo (Inicio del caso de uso 13)	Pasa
Se acciona el botón de editar artículo	Se muestra el modal de edición con los datos del artículo seleccionado (Inicio del caso de uso 18)	Pasa
Se acciona el botón de borrar de un artículo	Se muestra el modal de confirmación (Inicio del caso de uso 17)	Pasa
Se inicia la acción en masa de enviar correo	Se muestra el modal de envío de correo (Inicio del caso de uso 19)	Pasa
Se inicia la acción en masa de borrar artículos	Se muestra el modal de confirmación (Inicio del caso de uso 17)	Pasa
Se interactúa con cualquier lugar vacío de la fila de la tabla	Se redirige a la ventana de detalles del artículo (Inicio del caso de uso 14)	Pasa

Cuadro 7.18: Tests del caso de uso 12. VerArtículos.

<b>Caso de uso 18. ModificarPetición</b>		
Se edita algún campo del artículo	Se modifica la fila del artículo en la base de datos y se actualiza la vista	Pasa
Se borra algún campo obligatorio (orador, sección, subsección, tema, fecha límite, tamaños máximos y mínimos del título y la frase clave)	Se muestra un mensaje de error en el campo que falta	Pasa

Cuadro 7.19: Tests del caso de uso 18. ModificarPetición.

<b>Caso de uso 17. BorrarArtículo</b>		
Se borra un artículo	Se borra la fila correspondiente en la tabla <i>articles</i>	Pasa

Cuadro 7.20: Tests del caso de uso 17. BorrarArtículo.

<b>Caso de uso 19. EnviarCorreo</b>		
Se envía un correo con plantilla	Se envía el correo a los oradores de los artículos seleccionados	Pasa
Se envía un correo sin plantilla	Se envía el correo a los oradores de los artículos seleccionados	Pasa
Se envía un correo con cc o bcc	Se envía el correo a los oradores de los artículos seleccionados y a correos especificados en los campos de bcc y cc	Pasa

Cuadro 7.21: Tests del caso de uso 19. EnviarCoreo.

## 7.5. Detalles artículo (Caso de uso 14)

### Petición

Tests renderizado		
Artículo con todos los atributos	Se renderiza el componente sin ningún error	Pasa
Artículo solo con atributos obligatorios	Se renderiza el componente sin ningún error	Pasa
Hay petición de validación	Se muestra una exclamación en la pestaña de respuesta	Pasa

Cuadro 7.22: Tests de renderizado sobre el componente de la página de detalles del artículo.

Caso de uso 14. VerDetallesArtículo		
Se acciona cambia a la pestaña de respuesta cuando hay una validación pendiente	Se renderiza el componente de la respuesta del artículo (Inicio del caso de uso 15)	Pasa
Se borra algún campo obligatorio de la petición	Se muestra un error en el campo correspondiente	Pasa
Se inicia la acción en masa de enviar correo	Se muestra el modal de envío de correo (Inicio del caso de uso 19)	Pasa

Cuadro 7.23: Tests del caso de uso 14. VerDetallesArtículo.

### Respuesta

Tests renderizado		
Artículo sin respuesta	Muestra un mensaje informativo y permite iniciar el <i>caso de uso 19. EnviarCorreo</i>	Pasa
Artículo con respuesta	Se muestra la respuesta y permite iniciar el <i>caso de uso 15. ValidarRespuesta</i>	Pasa
Artículo rechazado con razón	Muestra la razón del rechazo y permite iniciar el <i>caso de uso 19. EnviarCorreo</i>	Pasa
Artículo rechazado sin razón	Permite iniciar el <i>caso de uso 19. EnviarCorreo</i>	Pasa
Hay una petición de cambio de fecha	Se muestra la petición y permite interactuar con ella para iniciar el <i>caso de uso 16. ValidarPeticiónFecha</i>	Pasa

Cuadro 7.24: Tests de renderizado sobre el componente de respuesta del artículo.

<b>Caso de uso 14. VerDetallesArtículo</b>		
Se acciona rechazar o aceptar una petición de cambio de fecha	Se muestra el modal de confirmación correspondiente (inicio del caso de uso 16)	Pasa
Se acciona el botón de validación de la respuesta	Se despliegan los estados, iniciando el caso de uso 15	Pasa
Se pulsa sobre el botón de enviar correo	Se muestra el modal de envío de correo (Inicio del caso de uso 19)	Pasa

Cuadro 7.25: Tests del caso de uso 14. VerDetallesArtículo parte 2.

<b>Caso de uso 15. ValidarRespuesta</b>		
Se valida el estado de una nueva respuesta	Se cambia el estado del artículo, modificando la fila en la tabla de <i>articles</i> y se encola el correo de confirmación	Pasa

Cuadro 7.26: Tests del caso de uso 15. ValidarRespuesta.

<b>Caso de uso 16. ValidarPeticiónFecha</b>		
Se acepta la petición	Se cambia la fila del artículo en base de datos y se envía un correo. Se borra la petición	Pasa
Se rechaza la petición y no se da nueva fecha	Se borra la petición de base de datos	Pasa
Se rechaza la petición y se da nueva fecha	Se cambia la fila del artículo en base de datos y se envía un correo. Se borra la petición	Pasa

Cuadro 7.27: Tests del caso de uso 16. ValidarPeticiónFecha.

## 7.6. Portal de participante

### Login

<b>Caso de uso 20. Login</b>		
Se introduce un correo de registro inexistente	Se muestra un mensaje de error indicando que no existe el registro	Pasa
Se introduce un correo existente	Se genera un nuevo código para el participante en la base de datos y aparece el campo para introducirlo en la vista	Pasa
Se introduce un código inválido	Se vuelve a la ventana de introducción del correo	Pasa
Se introduce un código válido	Se autentica al usuario creando una sesión, se redirige a la página principal y se borra el código de la base de datos.	Pasa

Cuadro 7.28: Tests del caso de uso 20. Login.

## Configuración de entorno

<b>Tests renderizado</b>		
<i>Componente de configuración</i>		
Solo un organizador disponible	Se renderiza el componente con los campos de organizador y evento	Pasa
Múltiples organizadores disponibles	Se renderiza el componente con los campos de organizador y evento	Pasa
Solo un evento disponible	Se renderiza el componente con los campos de organizador y evento	Pasa
Múltiples eventos disponibles	Se renderiza el componente con los campos de organizador y evento	Pasa
Usuario en modo asistente, solo un orador disponible	Se renderiza el componente con los campos de organizador, evento y orador	Pasa
Usuario en modo asistente, múltiples oradores disponibles	Se renderiza el componente con los campos de organizador, evento y orador	Pasa
<i>Componente en página de resumen de participación</i>		
Solo un organizador disponible	Al seleccionarse automáticamente el orador al solo haber uno, no se muestra la carta de selección	Pasa
Múltiples organizadores disponibles	Se renderiza la carta de selección de organizador	Pasa
Solo un evento disponible y organizador seleccionado	Se renderiza la carta de selección de evento	Pasa
Múltiples eventos disponibles y organizador seleccionado	Se renderiza la carta de selección de evento	Pasa
Usuario en modo asistente, con evento seleccionado y solo un orador disponible	Se renderiza la carta de selección de orador	Pasa
Usuario en modo asistente, con evento seleccionado y múltiples oradores disponibles	Se renderiza la carta de selección de orador	Pasa

Cuadro 7.29: Tests de renderizado de la configuración de entorno. Casos de uso 21 y 22.

<b>Tests renderizado</b>		
<i>Componente de configuración</i>		
Solo un organizador disponible	Se selecciona el organizador automáticamente y se guarda en sesión. Se selecciona en la interfaz.	Pasa
Solo un evento disponible	Se selecciona el evento automáticamente y se guarda en sesión. Se selecciona en la interfaz. Se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
Usuario en modo asistente, un orador disponible	Se selecciona el orador automáticamente y se guarda en sesión. Se selecciona en la interfaz.	Pasa
Organizador seleccionado en sesión	Se selecciona en la interfaz.	Pasa
Evento seleccionado en sesión	Se selecciona en la interfaz. Si no es un orador, se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
Usuario en modo asistente, orador seleccionado en sesión	Se selecciona en la interfaz. Se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
Se selecciona un organizador	Se guarda en sesión el organizador seleccionado y se recarga la página. Se selecciona en la interfaz	Pasa
Se selecciona un evento	Se guarda en sesión el evento seleccionado y se recarga la página. Se selecciona en la interfaz. Si es un orador, se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
Usuario en modo asistente, se selecciona un orador	Se guarda en sesión el orador seleccionado y se recarga la página. Se selecciona en la interfaz. Se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
<i>Componente en página de resumen de participación</i>		
Solo un organizador disponible	Se selecciona el organizador automáticamente y se guarda en sesión. Se selecciona en la interfaz. Se muestran los eventos	Pasa
Solo un evento disponible	Se selecciona el evento automáticamente y se guarda en sesión. Se selecciona en la interfaz. Se muestran los oradores en caso de ser asistente	Pasa
Usuario en modo asistente, múltiples oradores disponibles	Se selecciona el orador automáticamente y se guarda en sesión. Se selecciona en la interfaz. Se muestra la participación al evento. (Inicio caso de uso 25. VerParticipación)	Pasa
Organizador seleccionado en sesión	Se selecciona en la interfaz. Se muestran los eventos	Pasa

Cuadro 7.30: Tests de configuración del entorno. Casos de uso 21 y 22.

<i>Componente en página de resumen de participación</i>		
Evento seleccionado en sesión	Se selecciona en la interfaz. Se muestran los oradores en caso de ser asistente. En caso de no serlo, se muestra la participación (Inicio caso de uso 25. VerParticipación)	Pasa
Usuario en modo asistente, orador seleccionado en sesión.	Se selecciona en la interfaz. Se muestra la participación	Pasa
Se selecciona un organizador	Se guarda en sesión el organizador seleccionado y se recarga la página. Se selecciona en la interfaz. Se muestran los eventos	Pasa
Se selecciona un evento	Se guarda en sesión el evento seleccionado y se recarga la página. Se selecciona en la interfaz. En caso de no ser asistente, se muestra la participación (Inicio caso de uso 25. VerParticipación)	Pasa
Usuario en modo asistente, se selecciona un orador	Se guarda en sesión el orador seleccionado y se recarga la página. Se selecciona en la interfaz. Se muestra la participación (Inicio caso de uso 25. VerParticipación)	Pasa

Cuadro 7.31: Tests de configuración del entorno. Casos de uso 21 y 22. Parte 2.

### Resumen de participación

<b>Tests renderizado</b>		
Participación con todos los campos	Se renderiza exitosamente	Pasa
Participación con solo los campos obligatorios	Se renderiza exitosamente	Pasa

Cuadro 7.32: Tests renderizado caso de uso 25

<b>25. VerParticipación</b>		
Se pulsa cualquier botón de editar o añadir datos	Se cambia al modo edición el dato correspondiente, iniciando el caso de uso 27	Pasa
Se pulsa el botón de rechazar una actividad	Se abre el modal de confirmación referente al caso de uso 26	Pasa

Cuadro 7.33: Tests caso de uso 25

<b>26. RechazarActividad</b>		
Se confirma la acción	Se modifica el estado de la actividad a rechazado, modificando la fila del orador en la tabla de registros	Pasa
Se cancela la acción	Se cierra el modal	Pasa

Cuadro 7.34: Tests caso de uso 26. RechazarActividad

<b>27. ModificarParticipación</b>		
Se intenta añadir un requisito vacío	Se muestra un mensaje de error indicando que no puede ser vacío	Pasa
Se añade un requisito	Se crea una fila en la tabla <i>special_requests</i>	Pasa
Se modifican los vuelos	Se modifica el orador en la tabla de registros	Pasa

Cuadro 7.35: Tests caso de uso 27. ModificarParticipación

### Panel de información

<b>Tests renderizado</b>		
Orador con todos los campos de información	Se renderiza exitosamente	Pasa
Orador con solo los campos obligatorios	Se renderiza exitosamente	Pasa
Orador sin contactos vinculados	Se renderiza exitosamente	Pasa
Orador sin contactos con contactos vinculados (contacto con todos los atributos)	Se renderiza exitosamente	Pasa
Orador sin contactos con contactos vinculados (contacto con solo los atributos obligatorios)	Se renderiza exitosamente	Pasa

Cuadro 7.36: Tests renderizado del panel de información

<b>34. VerInformaciónPersonal</b>		
Se pulsa sobre cualquier campo	Se muestra el modal del caso de uso 35	Pasa
Se pulsa añadir un contacto vinculado	Se muestra el modal del caso de uso 35	Pasa
Se pulsa sobre el icono de verificación de un asistente	Se muestra el modal de confirmación del caso de uso 36	Pasa

Cuadro 7.37: Tests caso de uso 34. VerInformaciónPersonal

<b>35. ModificarInformaciónPersonal</b>		
Se cancela la acción	Se cierra el modal	Pasa
Se introduce un valor equivalente al ya existente en el orador	No se guarda como petición o se cancela la que había en ese campo	Pasa
Se introduce un cambio en algún campo	Se crea o modifica la petición de contacto en la tabla <i>contact_requests</i>	Pasa

Cuadro 7.38: Tests caso de uso 35. ModificarInformaciónPersonal

<b>36. ModificarAutorizaciónAsistente</b>		
Se cancela la acción	Se cierra el modal	Pasa
Se confirma la acción	Se modifica la fila correspondiente en la tabla <i>contact_links</i>	Pasa

Cuadro 7.39: Tests caso de uso 36. ModificarAutorizaciónAsistente

### Panel de sesiones

<b>Tests renderizado</b>		
Sesión con todos los atributos	Se renderiza exitosamente	Pasa
Sesión con solo los atributos obligatorios	Se renderiza exitosamente	Pasa
Sesión sin documentación	Se renderiza exitosamente	Pasa
Sesión con documentación	Se renderiza exitosamente	Pasa
Oradores de sesión con todos los atributos	Se renderiza exitosamente	Pasa
Oradores de sesión con solo los atributos obligatorios	Se renderiza exitosamente	Pasa

Cuadro 7.40: Tests renderizado del panel de sesiones

<b>29. ModificarDocumentosSesión</b>		
Se borra un documento	Se elimina el documento de la tabla <i>speaker_media</i>	Pasa
Se acciona añadir un documento	Se renderiza un modal mostrando los documentos del orador	Pasa
Se añade un documento sin una petición pendiente	Se crea una fila en la tabla <i>speaker_sessions_media_request</i> . Se añaden tantas filas como documentos nuevos haya en la tabla <i>request_document</i>	Pasa
Se añade un documento con una petición pendiente	Se añaden tantas filas como documentos nuevos haya en la tabla <i>request_document</i>	Pasa

Cuadro 7.41: Tests renderizado del caso de uso 29. ModificarDocumentosSesión

### Panel de documentos

<b>Tests renderizado</b>		
Orador con documentos	Se renderiza exitosamente	Pasa
Orador sin documentos	Se renderiza exitosamente	Pasa

Cuadro 7.42: Tests renderizado del panel de documentos

<b>24. Modificar Documentos</b>		
Se acciona añadir documentos	Se abre la biblioteca de documentos restringida a la carpeta del orador	Pasa
Se añade un documento	Se crean tantas filas en la tabla <i>speaker_media</i> como documentos haya	Pasa
Se elimina un documento	Se eliminan las filas correspondientes en la tabla <i>speaker_media</i>	Pasa

Cuadro 7.43: Tests renderizado del caso de uso 24. Modificar Documentos

### Panel de artículos

<b>Tests renderizado</b>		
Orador con artículos	Se renderiza exitosamente	Pasa
Orador sin artículos	Se renderiza exitosamente	Pasa
Artículos con todos los atributos	Se renderiza exitosamente	Pasa
Artículos con solo los atributos necesarios	Se renderiza exitosamente	Pasa

Cuadro 7.44: Tests renderizado del panel de artículos

<b>30. VerArtículosOrador</b>		
Se inicia la acción de cambiar la fecha de entrega	Se abre el modal correspondiente del caso 33	Pasa
Se selecciona la opción de escribir el artículo	Se abre el formulario de respuesta del artículo seleccionado, dando paso al caso 31	Pasa
Se selecciona la opción de rechazar el artículo	Se abre un modal de confirmación, iniciando el caso 32	Pasa

Cuadro 7.45: Tests del caso de uso 30. VerArtículosOrador

<b>31. ResponderArtículo</b>		
Se guarda la respuesta sin cumplir con los requisitos de longitud	Se guarda exitosamente en fila correspondiente en la tabla <i>articles</i>	Pasa
Se intenta enviar la respuesta sin cumplir con los requisitos de longitud	Muestra un error en los campos que no cumplen con las validaciones	Pasa
Envía la respuesta sin contactos en copia	Se modifica la fila de la tabla de artículos cambiando el estado y guardando la respuesta. Se envía el correo de confirmación del artículo	Pasa
Envía la respuesta con contactos en copia	Se modifica la fila de la tabla de artículos cambiando el estado y guardando la respuesta. Se envía el correo de confirmación del artículo al orador y los contactos especificados	Pasa

Cuadro 7.46: Tests del caso de uso 31. ResponderArtículo

<b>32. RechazarArtículo</b>		
Se cancela la acción	Se cierra el modal	Pasa
Se confirma la acción dando una razón	Se cambia el estado en la fila del artículo y se guarda la razón en esta misma	Pasa
Se confirma la acción sin dar una razón	Se cambia el estado en la fila del artículo	Pasa

Cuadro 7.47: Tests del caso de uso 32. RechazarArtículo

<b>33. PedirAmpliaciónFechaArtículo</b>		
Se cancela la acción	Se cierra el modal	Pasa
Se confirma la acción sin dar una fecha	Se muestra un error en el campo de la fecha	Pasa
Se da una fecha y confirma la acción	Se guarda en la fila del artículo la petición de fecha	Pasa

Cuadro 7.48: Tests del caso de uso 33. PedirAmpliaciónFechaArtículo

## 7.7. Conclusión de las pruebas

En resumen, todos los tests realizados, tanto unitarios como de integración, han tenido éxito, confirmando la robustez y funcionalidad del sistema desarrollado. Estos tests forman parte de una metodología de trabajo rigurosa y correcta, subrayando la importancia de la verificación y validación en el desarrollo de software. Las pruebas no solo aseguran que cada componente funcione correctamente de manera individual, sino que también garantizan que los diferentes módulos del sistema se integren de manera armoniosa y eficiente. La implementación de una

batería completa de tests es crucial para detectar y corregir errores a tiempo, mejorar la calidad del código y asegurar un despliegue sin problemas. Además, gracias a los tests implementados, en futuras actualizaciones se podrán detectar regresiones importantes gracias al sistema de CI/CD, que ejecuta todos los tests en cada despliegue. Así, se refuerza la confianza en el sistema y se proporciona una base sólida para su mantenimiento y escalabilidad futuros.



## Comentario del Cronograma

El cronograma está dividido en 3 fases principales:

- **Módulo administrativo de Oradores (07/02/2024 - 27/02/2024):** En esta fase se ha desarrollado el panel administrativo en la aplicación principal de Nudato de manera independiente sin tener en cuenta las funciones de validación. En esta fase se ha realizado el desarrollo de las funciones CRUD de los oradores y los artículos, su vista de listada y las acciones básicas sobre estos.
- **Portal de participantes (27/02/2024 - 25/03/2024):** Durante esta fase, se ha implementado el panel de participantes de manera autónoma del panel de organizadores. El sistema de login, las ventanas para ver información y la configuración del entorno.
- **Sistema de peticiones entre entornos (25/03/2024 - 29/03/2024):** En esta fase se integran ambos módulos independientes mediante la implementación de peticiones desde el portal de participantes y la validación de estas en el panel administrativo.
- **Desarrollo de los tests (29/03/2024 - 02/03/2024):** En esta fase final se han implementado tests unitarios y de integración para asegurar la calidad del código.

## 8.2. Evaluación de costes

La evaluación de costes se basa en el tiempo total dedicado a cada fase del proyecto. La suma de las 4 fases del proyecto da un total de **321 horas**.

A partir de este número de horas y los requisitos del proyecto, se ha realizado un análisis basado en las tablas salariales proporcionadas por la UGT en el XVIII Convenio Colectivo Estatal de Empresas de Consultoría, Tecnologías de la Información, y Estudios de Mercado y de la Opinión Pública (2020-2024)[14].

El trabajo desempeñado podría clasificarse dentro del Grupo C según el documento de referencia, ya que se han realizado tareas de análisis y desarrollo con un nivel de autonomía alto. Con un salario base anual de 26.534,12 €, repartidos entre 1820 horas, se ha generado un coste total de **4679,92 €**.

## 9 Legislación y protección de datos

En este apartado se analizará la legislación aplicable a la gestión de datos personales en la aplicación CRM para eventos. Se describirán principalmente las medidas que toma Nudato para cumplir con la Ley Orgánica de Protección de Datos (LOPD o LOPDGDD) [15]. Dentro de la LOPD se establecen distintos niveles de protección de datos en función del tipo de información manejada: básico, medio y alto. Dado que la aplicación maneja información personal no crítica, pero importante (datos sensibles, incluyendo información personal como nombre, correo, participación al evento), se posiciona en el grupo de nivel medio.

### 9.1. Medidas de protección

A continuación se describen las medidas de protección tomadas en nudato y los módulos desarrollados en este proyecto para asegurar que se cumplen con los requisitos listados por la LOPDGDD:

- **Control de acceso:** El panel administrativo demanda el uso de contraseñas seguras y permite el uso de 2FA. Por otro lado, el panel de participantes utiliza un método de control de acceso en función a un código de un solo uso que el participante recibe en el correo con cada intento de acceso.
- **Registro de actividades:** El panel administrativo mantiene un sistema de registros de todos los cambios realizados a los modelos de datos.
- **Copias de seguridad:** Se realizan copias de seguridad diarias tanto de las bases de datos de los *tenants* como de la central.
- **Política de privacidad:** La aplicación mantiene una política de privacidad [16] que informa a los usuarios sobre el manejo de sus datos y los derechos que estos tienen sobre ellos.
- **Gestión de consentimiento:** La vía principal de registro de nuevos participantes es el rellenado de un formulario que incluye una casilla de aceptación de la política de privacidad. Sin el marcado de este campo, no se puede proceder con el registro.
- **Derecho de acceso y rectificación:** Gracias a la implementación del panel de participantes, se ofrece una manera prácticamente automática de abordar estos derechos fundamentales del usuario sobre sus datos.

## 10 Conclusiones

El desarrollo de esta aplicación web para la gestión de eventos ha sido una experiencia enfocada al mundo real y desafiante que me ha permitido poner en práctica una gran cantidad de conocimientos adquiridos durante el grado. Cada decisión tecnológica ha sido fundamentada basándose en los requisitos del proyecto y el conocimiento que extraído de las asignaturas que cursadas en el marco de la carrera.

El mayor desafío ha sido trabajar con tecnologías con un nivel de abstracción muy alto sobre los patrones de diseño, estructuras de datos y otros conceptos aprendidos durante el grado. Esta abstracción ha requerido en varias ocasiones un adentramiento en el código fuente de los paquetes con tal de adaptar su comportamiento a los requisitos del aplicativo. Esta experiencia ha sido de mucho valor, dándome una visión clara de la importancia de una comprensión sólida de los fundamentos de la ingeniería de software y de como se aplican estos conocimientos en un contexto práctico.

Otro punto fundamental ha sido el desarrollo de interfaces de usuario intuitivas y eficientes, así como la reutilización de componentes entre diferentes paneles. Esto ha optimizado el tiempo de desarrollo (pasado y futuro) y ha garantizado una experiencia de usuario coherente y fluida.

Finalmente, este proyecto ha reafirmado la importancia de seguir metodologías ágiles en el desarrollo de software. El desarrollo iterativo e incremental de las funcionalidades del aplicativo, junto a las revisiones regulares con el cliente, han sido factores clave para el desarrollo de una aplicación útil y coherente en el mercado del software de gestión de eventos.

En resumen, la aplicación desarrollada no solo cumple con los objetivos planteados al inicio del proyecto, sino que también establece una base sólida para la creación de nuevas funcionalidades sobre el portal de participantes o el módulo del panel administrativo. La combinación de tecnologías elegidas, el enfoque metodológico adoptado, y la gran cantidad de tests, aseguran que la aplicación pueda seguir evolucionando y adaptándose a las necesidades cambiantes del mundo de la gestión de eventos.

## Bibliografía y recursos utilizados

- [1] Paul Greenberg. CRM at the Speed of Light. Social CRM Strategies, Tools, and Techniques for Engaging Your Customers. McGraw Hill Professional, 2010 (páginas 1, 2).
- [2] Código Abierto. Laravel v11 Documentación Oficial. URL: <https://laravel.com/docs/11.x> (páginas 1, 2, 150).
- [3] Código Abierto. Filament v3 Documentación Oficial. URL: <https://filamentphp.com/docs> (páginas 1, 2, 7, 151).
- [4] Ralph J. Smit. Filament Media Library, Documentación Oficial. URL: <https://filamentphp.com/plugins/ralphjsmit-media-library-manager#configuring-the-plugin-per-panel-filament-v3> (página 7).
- [5] Dempsey Chang, Laurence S. Dooley y Juhani E Tuovinen. «Gestalt theory in visual screen design: a new look at an old subject». en. En: Gestalt Theory in Visual Screen Design (jul. de 2002), págs. 5-12. URL: <http://crpit.com/confpapers/CRPITV8Chang.pdf> (página 32).
- [6] Martin Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley, 2002 (página 116).
- [7] Fred Chong y Gianpaolo Carraro. Architecting Multi-Tenant Applications for SaaS. Microsoft Corporation. 2006. URL: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/aa479086\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/aa479086(v=msdn.10)) (página 117).
- [8] Mehmet Baris Kalkar. Que es un framework opinionado. es. URL: <https://baris.io/blog/opinionated-frameworks> (página 150).
- [9] Código Abierto. Livewire v3 Documentación Oficial. URL: <https://livewire.laravel.com/> (página 151).
- [10] Código Abierto. TailwindCSS v3 Documentación Oficial. URL: <https://tailwindcss.com/> (página 152).
- [11] Código Abierto. Laravel Vapor Documentación Oficial. URL: <https://docs.vapor.build/introduction.html> (página 154).
- [12] Código Abierto. Amazon Web Services Documentación Oficial. URL: <https://docs.aws.amazon.com/> (página 154).
- [13] Kent Beck y Cynthia Andres. Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, 2004 (página 156).
- [14] Movilidad y Consumo de la Unión General de Trabajadores Federación Estatal de Servicios. BOE-A-2023-17238 Resolución de 13 de julio de 2023. es. Dirección General de Trabajo, por la que se registra y publica el XVIII Convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública. URL: <https://www.boe.es/buscar/doc.php?id=BOE-A-2023-17238> (página 179).

- [15] Agencia española de protección de datos (AEPD). Guía de Seguridad de Datos. es. URL: [https://www.ehu.eus/documents/2458096/2528821/guia\\_seguridad\\_datos\\_2008.pdf](https://www.ehu.eus/documents/2458096/2528821/guia_seguridad_datos_2008.pdf) (página 180).
- [16] EnvialT. Política de privacidad. es. URL: <https://enviait.com/politca-de-privacidad/> (página 180).