

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

SISTEMA DE GESTIÓN DE EVENTOS PARA LA CORPORACIÓN PYTHONCHILE

Trabajo de titulación para optar al título profesional de Ingeniero de Ejecución en Software.

Alumno:

Pablo Lira Gómez

Profesor Guía:

Óscar Francisco Carrasco Vera



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: SISTEMA DE GESTIÓN DE EVENTOS PARA LA CORPORACIÓN PYTHONCHILE

Nombre del candidato(a): Pablo Patricio Lira Gómez

Carrera / Grado: Ingeniero de Ejecución en Software

Campus: Viña del Mar ; **Departamento:** ELECTROTECNIA E INFORMÁTICA

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Óscar Carrasco Vera, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 08/08/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 08/08/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradezco a mis padres, hermanos y a todos lo que me apoyaron.

RESUMEN

KEYWORDS: EVENTOS, PYTHON, ONLINE, AUTOMATIZACIÓN DE PROCESOS, SISTEMA DE GESTIÓN EVENTOS.

El objetivo del proyecto presente en este escrito es el estudio de viabilidad de distintas alternativas de solución a la problemática de "gestión de eventos" con el propósito de mejorar la ejecución de los eventos, seleccionando la alternativa que mejor cumple a completitud las necesidades de la Corporación Python Chile.

La Corporación Python Chile es una organización sin fines de lucro, dedicada a difundir el lenguaje de programación Python en el país, en consecuencia, una de las actividades es la organización de distintos eventos durante el año.

Los eventos se han organizados con servicios de terceros y distintas herramientas han puesto a la palestra distintos inconvenientes como una experiencia usuario deficiente al tener que ingresar en distintos sitios y con otros idiomas, cambios de políticas de servicios, inconsistencia en los datos etc.

La alternativa seleccionada es "Desarrollo a la medida de las necesidades del cliente", por lo cual está en acorde a los requerimientos se desarrolla un proyecto de software que incluye desarrollo de plan de personal, planificación temporal y de recursos, administración del riesgo y costos. Y su posterior Análisis y Diseño de la solución.

Capítulo 1

Se ven aspectos relevantes del proyecto y su gestión. En donde se analizan el cliente y la situación actual, se detectan los problemas, se levantan los objetivos que se quieren lograr, se toman los requerimientos del usuario, se proponen tres alternativas de solución, se analizan y se escoge bajo un criterio la más adecuada.

Capítulo 2

Se observan aspectos relevantes del análisis para la alternativa seleccionada, donde se revisa la descripción de la solución propuesta. Se especifican los requerimientos funcionales y no funcionales, casos de uso, así como la matriz de trazabilidad que muestra la completitud de los requisitos funcionales satisfechos en los casos de uso.

El caso de uso general junto con el modelo conceptual nos permite obtener una perspectiva holística del sistema. Por último, tenemos los modelos de casos de uso narrativos con sus correspondientes diagramas de secuencia y contratos.

Capítulo 3

Se realizan aspectos relevantes del diseño para la alternativa seleccionada como la arquitectura de software, dando cumplimiento a los requerimientos no funcionales, organizando la estructura de la aplicación, teniendo una perspectiva holística del diseño del

software en base a los diagramas de componentes, modelo de clases, modelo relacional, diccionario de datos. El contexto de ejecución se ha seleccionado el sistema operativo Ubuntu/Linux, con el lenguaje de programación Python 3 y otras tecnologías. Se detalla más el comportamiento del sistema con diagramas de secuencias extendidos y colaboración. Y diseñan prototipos de la aplicación.

ÍNDICE

INTRODUCCIÓN	1
1. ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTION.....	4
1.1. Descripción de la organización	4
1.2. Descripción de la situación sin proyecto	6
1.2.1. Descripción de la situación	6
1.3. Problemas detectados.....	10
1.4. Objetivo general y específicos	11
1.5. Requerimientos del usuario	11
1.5.1. Requerimiento del usuario	11
1.6. Alternativas globales de solución	13
1.6.1. Alternativa 1. Optimizar procesos que ya se tienen mediante scripts	13
1.6.2. Alternativa 2. Contratar servicios administración de eventos de terceros	14
1.6.3. Alternativa 3. Desarrollo a la medida a las necesidades del cliente	15
1.7. Criterios para evaluar las alternativas.....	16
1.7.1. Ponderación de criterios y escala de evaluación	16
1.7.2. Evaluación	17
1.8. Alternativa Seleccionada	18
1.8.1. Descripción y beneficios de la alternativa seleccionada	18
1.8.2. Desarrollo del plan de personal	19
1.8.3. Planificación temporal y asignación de recursos por actividad	20
1.8.4. Estimación de costo	21
1.8.5. Administración del riesgo	22
2. ASPECTOS RELEVANTES DEL ANÁLISIS	26
2.1. Descripción de la solución propuesta	26
2.2. Actores del Sistema	27
2.3. Diagrama de actividades general UML	28
2.4. Requerimientos del sistema	29
2.4.1. Requerimientos funcionales.....	29
2.4.2. Requerimientos no funcionales.....	30
2.5. Estructura función del sistema	31
2.6. Matriz de trazabilidad	32
2.7. Modelo conceptual	33

2.8.	Modelo de casos de uso	34
2.8.1.	Caso de Uso: CU-01 Mantener Evento	34
2.8.2.	Caso de Uso: CU-02 Administrar entradas	36
2.8.3.	Caso de Uso: CU-03 Enviar Notificación	37
2.8.4.	Caso de Uso: CU-04 Obtener reporte de asistencia y encuesta	39
2.8.5.	Caso de Uso: CU-05 Mantener encuesta	41
2.8.6.	Caso de Uso: CU-06 Administrar postulación de ponencias	43
2.8.7.	Caso de Uso: CU-07 Asignar rol	45
2.8.8.	Caso de Uso: CU-08 Administrar agenda	46
2.8.9.	Caso de Uso: CU-09 Mantener nivel de patrocinio	47
2.8.10.	Caso de Uso: CU-10 Asignar cantidad de notificaciones	48
2.8.11.	Caso de Uso: CU-11 Mantener Perfil	50
2.8.12.	Caso de Uso: CU-12 Monitorear inscripciones al evento.....	51
2.8.13.	Caso de Uso: CU-13 Obtener Entrada	53
2.8.14.	Caso de Uso: CU-14 Cancelar entrada.....	55
2.8.15.	Caso de Uso: CU-15 Mantener cuenta	57
2.8.16.	Caso de Uso: CU-16 Registrar ingreso.....	59
2.8.17.	Caso de Uso: CU-17 Responder encuesta	60
2.8.18.	Caso de Uso: CU-18 Administrar ponencia	62
2.8.19.	Caso de Uso: CU-19 Cancelar ponencia aceptada.....	63
3.	ASPECTO RELEVANTES DEL DISEÑO PARA LA ALTERNATIVA SELECCIONADA.	66
3.1.	Arquitectura de software.....	66
3.2.	Diseño de datos.....	68
3.2.1.	Modelo de Clases.....	68
3.2.2.	Modelo Dato Relacional	72
3.2.3.	Diccionario de datos.....	74
3.3.	Diagrama de secuencia extendido.....	81
3.4.	Diseño de interfaz.....	85
	CONCLUSIONES Y RECOMENDACIONES	90
	BIBLIOGRAFÍA	91

ÍNDICE DE FIGURAS

Figura 1-1. Organigrama del Directorio de la Corporación Python Chile.	4
Figura 1-2. Organigrama de la Corporación de Python Chile	4
Figura 1-3. Conferencia PyCon Chile 2023 Valparaíso – Chile.....	5
Figura 1-4. Sitios web www.pycon.cl y www.pyday.cl.....	6
Figura 1-5. Eventbrite	7
Figura 1-6. Sessionize	7
Figura 1-7. Formulario de votación	8
Figura 1-8. Formularios de patrocinios y de socios colaboradores	8
Figura 1-9. Formulario de retroalimentación	9
Figura 1-10. Imagen del Servicio Hubilo	15
Figura 2-1. Diagrama de actividades	28
Figura 2-2. Caso de uso general	31
Figura 2-3. Modelo conceptual Fuente: Elaboración Propia	33
Figura 2-4. Diagrama Secuencia: CU-01 Gestionar Evento	35
Figura 2-5. Diagrama Secuencia: CU-02 Administrar entradas	36
Figura 2-6. Diagrama Secuencia: CU-03 Enviar Notificación	38
Figura 2-7. Diagrama Secuencia: CU-04 Obtener reporte de asistencia y encuesta.....	40
Figura 2-8. Diagrama Secuencia: CU-05 Mantener encuesta.....	42
Figura 2-9. Diagrama Secuencia: CU-06 Administrar Postulación.....	44
Figura 2-10. Diagrama Secuencia: CU-07 Asignar Rol	45
Figura 2-11. Diagrama Secuencia: CU-08 Administrar agenda	46
Figura 2-12. Diagrama Secuencia: CU-09 Gestionar categoría de patrocinio	47
Figura 2-13. Diagrama Secuencia: CU-10 Asignar cantidad de notificación.....	49
Figura 2-14. Diagrama Secuencia: CU-11 Mantener Perfil Fuente: Elaboración Propia	50
Figura 2-15. Diagrama Secuencia: CU-12 Monitorear inscripciones al evento.....	52
Figura 2-16. Diagrama Secuencia: CU-13 Obtener Entrada	54
Figura 2-17. Diagrama Secuencia CU-14 Cancelar Entrada.....	56
Figura 2-18. Diagrama Secuencia: CU-15 Mantener Cuenta	58
Figura 2-19. Diagrama Secuencia: CU-16 Registrar ingreso.....	59
Figura 2-20. Diagrama Secuencia: CU-17 Responder encuesta	61
Figura 2-21. Diagrama Secuencia: CU-18 Administrar postulación	62
Figura 2-22. Diagrama Secuencia: CU-19 Cancelar Ponencia	64
Figura 3-1. Diagrama de despliegue.....	66
Figura 3-2 Modelo Clases General.....	69
Figura 3-3. Modelo Clases de Roles y Notificación	70
Figura 3-4. Modelo Clases de encuesta y entradas.....	71
Figura 3-5. Modelo de clases postulación, ponencia y patrocinio.....	72
Figura 3-6. Modelo relacional	73
Figura 3-7. DSE: CU-01 Mantener evento	81
Figura 3-8 DSE: CU-02 Administrar entradas.....	81
Figura 3-9 DSE: CU-03 Enviar notificación	82
Figura 3-10 DSE: CU-04 Obtener reporte asistencia y encuesta	82

Figura 3-11 DSE: CU-05 Mantener Encuesta.....	82
Figura 3-12 DSE: CU-06 Administrar postulación de ponencias	83
Figura 3-13 DSE: CU-07 Asignar rol	83
Figura 3-14 DSE: CU-08 Administrar agenda	84
Figura 3-15 DSE: CU-09 Mantener nivel de patrocinio	84
Figura 3-16. 10 leyes de usabilidad de Jacob Nielse	85
Figura 3-17. Pantalla principal	87
Figura 3-18. Pantalla Keynotes	87
Figura 3-19. Pantalla Cronograma	88
Figura 3-20. Pantalla Auspiciadores	89

ÍNDICE DE TABLAS

Tabla 1.1. Ponderación de los criterios.....	16
Tabla 1.2. Escala de evaluación de los criterios.	16
Tabla 1.3. Alternativa 1. Calificación de Optimizar procesos mediante scripts	17
Tabla 1.4. Alternativa 2. Calificación de Contratar servicios de terceros.....	17
Tabla 1.5. Alternativa 3. Calificación del Desarrollo a la medida de las necesidades del cliente	17
Tabla 1.6. Resumen de Calificación de Alternativas.....	18
Tabla 1.7. Resumen de la Programación de Actividades.	20
Tabla 1.8. Costo empresa mensual	21
Tabla 1.9. Costo proyecto	22
Tabla 1.10. Tabla Administración del Riesgo	23
Tabla 1.11. Plan de Mitigación.....	24
Tabla 2.1. Actores del sistema.	27
Tabla 2.2. Requisitos no funcionales	30
Tabla 2.3 matriz de trazabilidad	32
Tabla 2.4. Caso de Uso Narrativo: CU-01 Mantener Evento	34
Tabla 2.5. Contrato: CU-01 crear_evento.....	35
Tabla 2.6. Caso de uso narrativo: CU-02 Administrar entradas	36
Tabla 2.7. Contratos: CU-02 guardar_entradas	36
Tabla 2.8. Caso de uso narrativo: CU-03 Enviar Notificación.....	37
Tabla 2.9. Contratos: CU-03.1 enviar_notificación	38
Tabla 2.10. Contrato: CU-03.2 restar_notificación	38
Tabla 2.11. Caso de uso narrativo: Obtener reporte de asistencia y encuesta	39
Tabla 2.12. Contrato: CU-04.1 generar_reporte	40
Tabla 2.13. Contrato: CU-04.2 desplegar_parametros.....	40
Tabla 2.14. Caso de uso narrativo: CU-05 Mantener encuesta	41
Tabla 2.15. Contrato: CU-05.1 crear_encuesta	42
Tabla 2.16. Contrato: CU-05.2 validar_datos	42
Tabla 2.17. Caso de uso narrativo: Administrar postulación de ponencias	43
Tabla 2.18. Contrato: CU-06 administrar_postulaciones	44
Tabla 2.19. Caso de uso narrativo: CU-07 Asignar rol	45

Tabla 2.20. Contrato: CU-07 asignar_rol.....	46
Tabla 2.21. Caso de uso narrativo CU-08 Administrar agenda.....	46
Tabla 2.22. Contratos: CO-08 crear_agenda.....	47
Tabla 2.23. Caso de uso narrativo: CU-09 Mantener nivel de patrocinio.....	47
Tabla 2.24. Contratos: CO-09 Gestionar categoría de patrocinio.....	48
Tabla 2.25 Caso de uso narrativo CU-10 Asignar cantidad de notificaciones	48
Tabla 2.26. Contrato: CU-10 asignar_notificaciones	49
Tabla 2.27. Caso de uso narrativo: CU-11 Mantener perfil.....	50
Tabla 2.28. Contratos: CU-11 crear_perfil	50
Tabla 2.29. Caso de uso narrativo: CU-12 Monitorear inscripciones al evento	51
Tabla 2.30. Contrato: CU-12 monitorear_incripciones	52
Tabla 2.31. Contrato: CU-12.2 procesar_opciones	52
Tabla 2.32. Caso de uso narrativo: CU-13 Obtener Entrada.....	53
Tabla 2.33. Contratos: CO-13 asignar_entrada	54
Tabla 2.34. Caso de uso narrativo: CU-14 Cancelar entrada	55
Tabla 2.35. Contratos: CO-14 cancelar_entrada	56
Tabla 2.36 Caso de uso narrativo: CU-15 Mantener cuenta	57
Tabla 2.37. Contrato: CU-15 modificar	58
Tabla 2.38. Caso de uso narrativo: CU-16 registrar ingreso.....	59
Tabla 2.39. Contratos: CO-16 comprobar_entrada.....	60
Tabla 2.40. Caso de uso narrativo: responder encuesta.....	60
Tabla 2.42. Contrato: CU-17 validar_repuestas	61
Tabla 2.41. Contrato: CU-17 guardar_respuestas	61
Tabla 2.43. Caso de uso narrativo: CU-18 Administrar postulación	62
Tabla 2.44. Contrato: CU-18 Administrar Ponencia	63
Tabla 2.45. Caso de uso narrativo: CU-19 Cancelar ponencia aceptada.....	63
Tabla 2.46. Contrato CU-19 Cancelar Ponencia	64
Tabla 3.1. Diccionario de Datos: usuarios	74
Tabla 3.2. Diccionario de Datos: usuario_rol_evento.....	74
Tabla 3.3. Diccionario de Datos: tipo_rols.....	74
Tabla 3.4. Diccionario de Datos: perfil_publico.....	75
Tabla 3.5. Diccionario de Datos: eventos	75
Tabla 3.6. Diccionario de Datos: usuario_notificacion	75
Tabla 3.7. Diccionario de Datos: notificacion.....	76
Tabla 3.8. Diccionario de Datos: entradas	76
Tabla 3.9. Diccionario de Datos: usuario_entrada	76
Tabla 3.10. Diccionario de Datos: patrocinio.....	77
Tabla 3.11. Diccionario de Datos: patrocinadores_evento	77
Tabla 3.12. Diccionario de Datos: agenda_evento	77
Tabla 3.13. Diccionario de Datos: bloque_horario	78
Tabla 3.14. Diccionario de Datos: postulación_ponencias.....	78
Tabla 3.15. Diccionario de Datos: llamado_postulaciones.....	78
Tabla 3.16. Diccionario de Datos: ponencias.....	79
Tabla 3.17. Diccionario de Datos: encuestas.....	79

Tabla 3.18. Diccionario de Datos: pregunta	79
Tabla 3.19. Diccionario de Datos: encuesta_pregunta	80
Tabla 3.20. Diccionario de Datos: repuesta_encuesta	80

INTRODUCCIÓN

Las reuniones entre individuos se han realizado desde el comienzo de la civilización humana. Es así como las personas se han reunido en torno a diferentes motivos, como celebraciones, recitales, tradiciones, recolecciones, cacería, etcétera.

Las reuniones son un acto consciente de un grupo de personas que requiere un punto o lugar de encuentro, una planificación, un motivo o tema, un objetivo, una organización, una fecha y duración. Las reuniones se realizan para compartir o intercambiar bienes, conocimientos, experiencias o por el simple hecho fomentar relaciones.

Un evento es todo tipo de reunión con un propósito definido, por ejemplo, conferencias, congresos, seminarios, talleres, capacitaciones, torneos, conmemoraciones, presentación de nuevos productos etc.

Una planificación deficiente de un evento implicará un aumento del riesgo de fracaso e incidentes inesperados que pongan entredicho la calidad del evento y la seguridad de los participantes y su dificultad está dada por su estructura, cantidad de participantes, temática, grado de formalidad etcétera. Así también implica las posibles repercusiones económicas y responsabilidades legales de los organizadores.

La corporación sin fines de lucro Python Chile dedicada a difundir el lenguaje de programación Python ubicada en Santiago de Chile, realiza durante el año diferentes tipos de eventos, con el objetivo de congrega profesionales, entusiastas, integrantes del sector de la tecnología de la información y las comunicaciones, académicos con intereses en el lenguaje de programación Python y temáticas relacionadas. En estas instancias benefician y generan un impacto positivo al sector, los participantes pueden realizar networking, aprender nuevas habilidades, presentar trabajos investigativos, compartir experiencias, adquirir nuevos conocimientos etcétera.

Con la experiencia adquirida por la realización de eventos, la Corporación Python Chile ha identificado necesidad de mejora en la gestión de los eventos buscando centralizar la experiencia de administrar el evento en una sola aplicación de software.

El objetivo principal del proyecto es "Mejorar la gestión de la organización de eventos para la Corporación Python Chile a través de una solución de software". Con esto se quiere llevar a cabo exitosamente la ejecución de la conferencia y obteniendo información relevante para futura tomas de decisiones.

Los objetivos específicos del proyecto que ayudan a alcanzar el objetivo principal son: Consolidar la información; Facilitar la inscripción y seguimientos de los participantes; Analizar los datos para la toma de decisiones, y la Gestión de espacios físicos.

CAPÍTULO 1. ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTIÓN

1. ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTION

En este capítulo se presenta la institución y sus antecedentes, también se exponen la información relevante del proyecto.

1.1. Descripción de la organización

El cliente es la Corporación Python Chile la cual es organización sin fines de lucro constituida en el año 2022, con domicilio en la comuna de Las Condes, Región Metropolitana de Santiago de Chile. Su nombre es gracia al lenguaje de programación Python. La corporación es de reciente creación realizada por los miembros de la comunidad de Python Chile que llevan cuatro años realizando actividades.

La misión de la Corporación Python Chile es promover el uso de lenguaje de programación Python para generar un impacto positivó en el ecosistema educativo, industrial, investigativo etcétera del país.

La Visión de Python Chile es ser un punto de encuentro de todos los entusiastas del lenguaje de programación en el país.

El Directorio de la Corporación Python Chile se estructura de la siguiente forma.

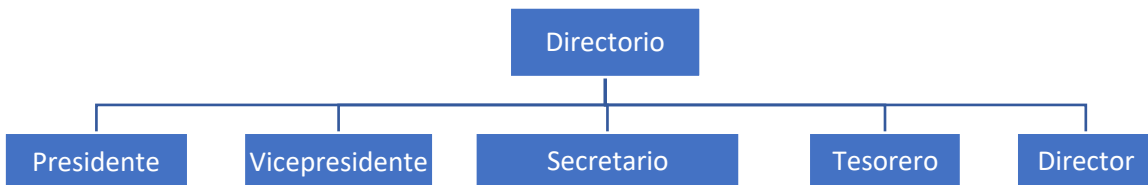


Figura 1-1. Organigrama del Directorio de la Corporación Python Chile.

Fuente: Elaboración Propia

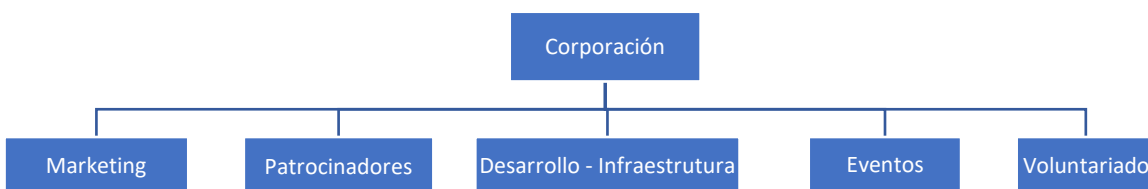


Figura 1-2. Organigrama de la Corporación de Python Chile

Fuente: Elaboración Propia

La corporación de Python Chile tiene como objetivo ser un lugar de encuentro de todos los entusiastas del lenguaje de programación Python a lo largo del país, fortaleciendo el ecosistema y así generar un impacto positivo en los actores de la industria del desarrollo de software.

Python Chile realiza los siguientes tipos eventos:

Conferencia: De índole académico formal y divulgativo, puede tener una duración de un o más días, en donde los conferencistas expertos tienen una presentación oral sobre un tema de interés para el público, puede combinarse con otras actividades como talleres y hackáthones o espacios de tiempo y áreas físicas para realizar networking.

Eventos satélites: Es un evento que acompaña a una conferencia principal. Tiene la característica que se realiza en otra ubicación geográfica.

Hackathon: Es un evento con el objetivo de realizar colaborativamente una solución tecnológica sobre un problema en específico, en donde los participantes se agrupan para realizar el desafío, se cuenta con un límite de tiempo fijado para completar el desafío y cabe destacar que los participantes son profesionales, estudiantes y entusiastas con conocimientos sobre desarrollo de soluciones.

Workshop (Talleres): Es un evento en la cual se realizan talleres prácticos sobre un tema en específico, por ejemplo, introducción a la programación de Python, en donde los participantes aprenden a programar con Python.

Meetup: Es un evento organizado a través de la plataforma meetup.com en la cual se planifican una o dos presentaciones que tienen máximo de duración en conjunto de una hora para exponer un tema, pueden ser virtuales, presenciales o híbridos. Los participantes al terminar tienen un espacio para hacer networking.



Figura 1-3. Conferencia PyCon Chile 2023 Valparaíso – Chile
Fuente: Corporación Python Chile.

1.2. Descripción de la situación sin proyecto

En esta sección se exponen la situación actual de los procesos que se llevan a cabo para realizar las tareas necesarias para completar el objetivo de concretar un evento.

1.2.1. Descripción de la situación

La corporación Python Chile organiza dos eventos de tipo conferencias al año, PyCon con duración de uno a tres días y la PyDay de duración de un día con eventos satélites.

Después que el comité de organización define la planificación del evento, este se plasma en un sitio web estático, el cual cumple la función de informar a todos los interesados las fechas importantes por ejemplo cuando se realizara la conferencia, postulación de ponentes, cierre de auspicios etcétera. Así también el cronograma, quienes auspician, los organizadores, socios colaboradores etcétera. Toda esta información se realiza de manera manual y directamente en la plantilla HTML.

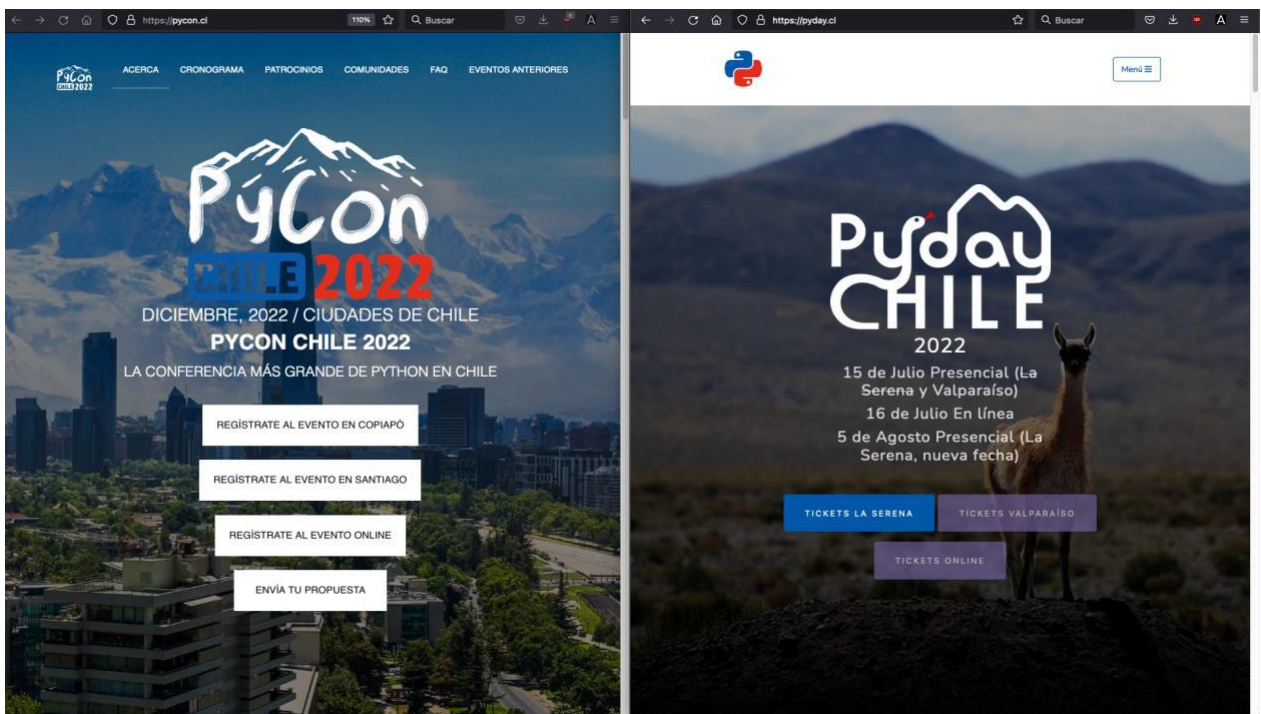


Figura 1-4. Sitios web www.pycon.cl y www.pyday.cl

Fuente: Elaboración propia

Una vez concretado el sitio web de la conferencia se crea un evento en la plataforma Eventbrite.

La plataforma permite a los organizadores gestionar los participantes de la conferencia, cada participante se inscribe en el evento con sus datos de contacto y la plataforma le asigna un boleto único de entrada. El formulario de inscripción es modificable y no tiene la funcionalidad de realizar encuestas posteriores al evento.

También provee de la funcionalidad de enviar una notificación vía mail a los participantes de novedades de la conferencia etc., pero para participantes de eventos anteriores no se pueden enviar por políticas de la plataforma.

La siguiente tarea es crear una convocatoria de presentación en la plataforma Sessionize que solo se encuentra en Inglés.

Los ponentes pueden enviar sus propuestas de presentaciones través de la plataforma. Una vez terminado el plazo de postulación un comité de selección escoge las presentaciones que se expondrán en la conferencia.

Se crea un cronograma con las presentaciones seleccionadas y se exporta para la utilización en el sitio de la conferencia.

La plataforma Sessionize permite la votación de las presentaciones de la conferencia solamente a los organizadores e invitados.

Por lo cual se debe de llevar a cabo una votación a través de un formulario creado en Google Forms para que los participantes puedan votar por las presentaciones que más interesen.

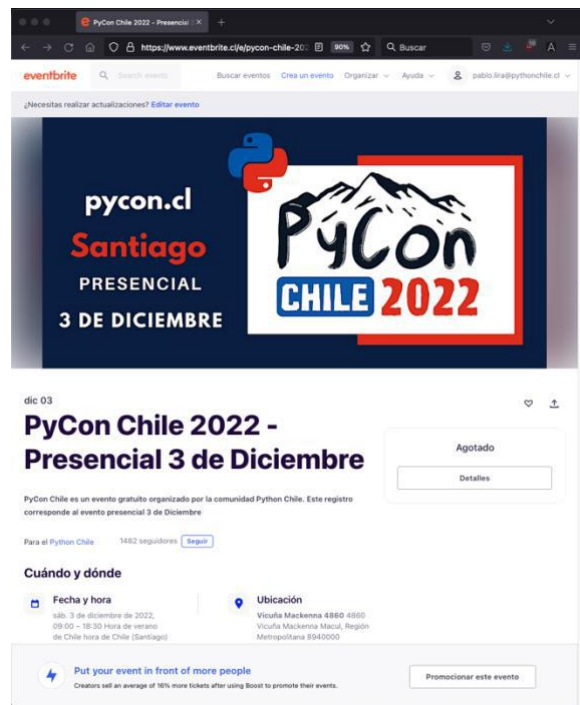


Figura 1-5. Eventbrite
Fuente: Elaboración Propia

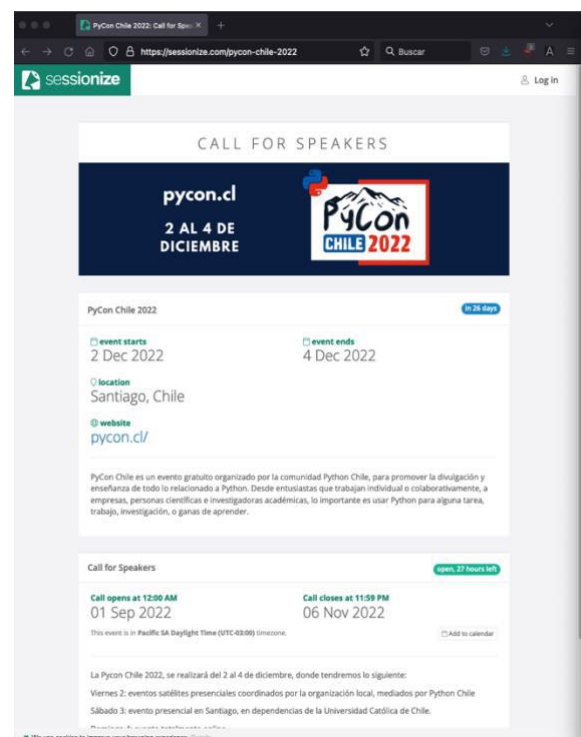


Figura 1-6. Sessionize
Fuente: Elaboración Propia

El comité de selección también realiza una encuesta utilizando la herramienta de creación de formularios Google Forms, la cual permite recabar la información de las preferencias de los participantes y exportar los datos a una hoja de calculo

The screenshot shows a Google Forms interface for a voting survey. The title is 'Votación Charlas PyCon Chile 2021'. The main text asks participants to help decide on talks for the first PyCon Chile. It lists several topics, including 'Exploit Development with Python', and provides three radio button options for each: 'No me gustaría', 'No tengo opinión', and 'Me gustaría'. The form is displayed in a browser window with the URL 'https://docs.google.com/forms/d/1waer1cQQn5D0...'. The interface includes a navigation bar with 'Preguntas', 'Respuestas' (400), and 'Configuración' tabs. A sidebar on the right contains icons for sharing, printing, and other actions.

Figura 1-7. Formulario de votación

Fuente: Elaboración Propia

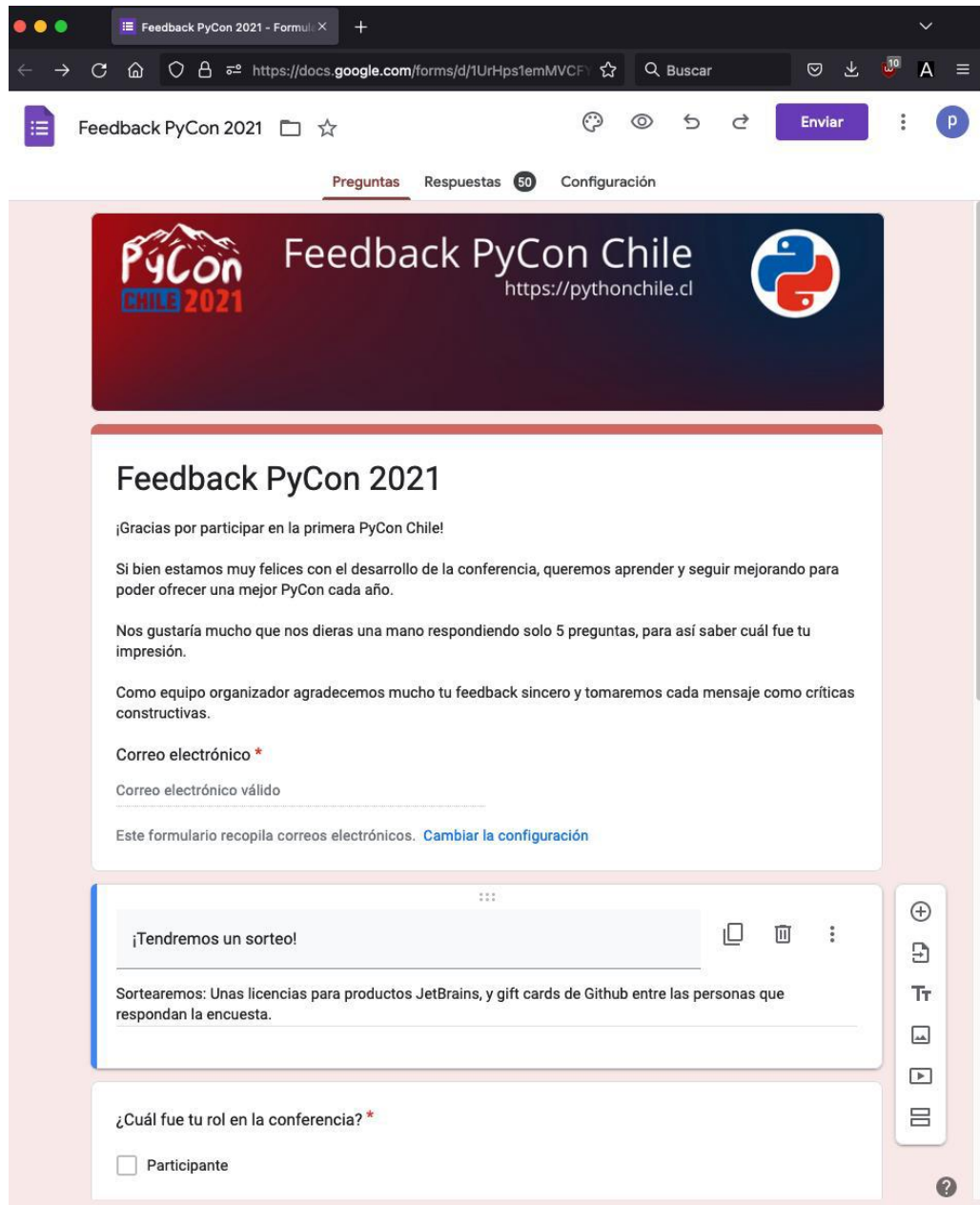
Se confeccionan dos formularios más para obtener los datos de contactos de los patrocinadores y de los socios colaboradores sin ánimo de lucro. Posteriormente se pondrán en el sitio de la conferencia de manera manual los logos y la dirección web de los patrocinantes y de los colaboradores.

The image shows two side-by-side screenshots of Google Forms. The left form is titled 'Postulación de patrocinios para la PyCon Chile 2021' and includes the Python Chile logo. It provides information about the event (November 5, 6, and 7, 2021) and asks for contact details such as name, email, and organization name. The right form is titled 'Colaboración PyCon Chile 2021' and also features the Python Chile logo. It explains the collaboration process, including the event dates and the goal of increasing participation. It lists requirements for collaborators, such as sharing social media posts and providing a logo and URL. Both forms are displayed in browser windows with their respective URLs.

Figura 1-8. Formularios de patrocinios y de socios colaboradores

Fuente: Elaboración Propia

Al finalizar la conferencia, nuevamente se realiza un formulario para recabar información en búsqueda de retroalimentación para mejoras en próximas conferencias. Este formulario se envía por correo electrónico y se comparte por las cuentas de las redes sociales oficiales, cualquier persona con el enlace completar el formulario.



The image shows a browser window displaying a Google Forms survey. The browser's address bar shows the URL: <https://docs.google.com/forms/d/1UrHps1emMVCF/>. The form title is "Feedback PyCon 2021". The header features the PyCon Chile 2021 logo and the text "Feedback PyCon Chile" with the website URL <https://pythonchile.cl>. The form content includes a thank-you message, a request for feedback, and a contest announcement. The contest details are: "¡Tendremos un sorteo!" followed by "Sortearemos: Unas licencias para productos JetBrains, y gift cards de Github entre las personas que respondan la encuesta." Below this, there is a question: "¿Cuál fue tu rol en la conferencia?*" with a radio button option for "Participante".

Feedback PyCon 2021

¡Gracias por participar en la primera PyCon Chile!

Si bien estamos muy felices con el desarrollo de la conferencia, queremos aprender y seguir mejorando para poder ofrecer una mejor PyCon cada año.

Nos gustaría mucho que nos dieras una mano respondiendo solo 5 preguntas, para así saber cuál fue tu impresión.

Como equipo organizador agradecemos mucho tu feedback sincero y tomaremos cada mensaje como críticas constructivas.

Correo electrónico *

Correo electrónico válido

Este formulario recopila correos electrónicos. [Cambiar la configuración](#)

¡Tendremos un sorteo!

Sortearemos: Unas licencias para productos JetBrains, y gift cards de Github entre las personas que respondan la encuesta.

¿Cuál fue tu rol en la conferencia? *

Participante

Figura 1-9. Formulario de retroalimentación
Fuente: Elaboración propia

1.3. Problemas detectados

En base a lo descrito anteriormente se han encontrado los siguientes problemas:

1. Deficiente proceso actualización de información: La actualización de la información es ingresada manualmente en el documento HTML del sitio de la conferencia el cual posteriormente debe de subirse al servidor web.
2. La plataforma se encuentra en inglés: Sessionize no se encuentra en español, lo cual puede traer confusión en ponentes que no se manejen en el idioma causando confusión en él envió de sus propuestas de presentación.
3. Cambios en la política de los servicios: Los servicios web que se usan sufren cambios en sus plataformas y políticas los cuales afectan las condiciones del uso de los servicios, restringiendo opciones en la toma de decisiones.
4. Sobre carga de autenticación: El usuario debe de autenticarse en distintos servicios causando frustración por el sistema poco amigable y cambio de contextos.
5. Defectos en Procesos manuales: La manipulación de gran cantidad de datos puede ocasionar inconsistencias como datos erróneos, desactualizados etc.
6. Sin automatización en la creación de formulario de votos: El servicio de Sessionize no permite la votación masiva de los participantes, por lo cual realizan un formulario en Google Forms, en el cual no se lleva a cabo ningún proceso de verificación.
7. Poca veracidad en retroalimentación: En el formulario de retroalimentación de la conferencia, cualquier persona con el enlace puede responder el formulario, no existe un proceso de verificación o cruce de datos.
8. Nula verificación de votantes: Debido al uso de distintos servicios no se verifica la autenticidad de las personas que votan las presentaciones, no hay un proceso de cruce de datos, por lo cual puede darse el caso de votaciones falsas en favor de un ponente.

1.4. Objetivo general y específicos

El objetivo principal del proyecto es mejorar la gestión de la organización de eventos para la Corporación Python Chile mediante una solución de software que facilite la ejecución exitosa de conferencias y proporcione información relevante para futuras decisiones.

Los objetivos específicos del proyecto que ayudan a alcanzar el objetivo principal son:

1. **Optimizar la consolidación de la información:** Consolidar toda la información del evento para acceso sencillo y rápido.
2. **Simplificar tareas del personal organizador:** Mejorar la experiencia de ejecución o configuración de aspectos relevantes a la creación de un evento, también trae como beneficio una optimización del tiempo que lleva hacer la tarea.
3. **Facilitar la inscripción de los participantes:** Mejorar la experiencia del usuario al consolidar los eventos en un solo sitio.
4. **Mejorar la experiencia con los expositores:** Simplificar los procesos de postulación de sus ponencias y demás tareas, así como las actualizaciones de fotografías y descripciones de las ponencias.
5. **Mejorar la relación con los patrocinadores y organizadores:** Aumentar la colaboración para aumentar la visibilidad de su participación en los eventos.

1.5. Requerimientos del usuario

Los requerimientos del usuario son enunciados expresados en lenguaje natural y claro que describen las condiciones, capacidades o servicios que los usuarios esperan de un sistema para resolver un problema o alcanzar un objetivo. Estos requerimientos reflejan los deseos, necesidades y restricciones desde la perspectiva del usuario, y sirven como base para el desarrollo de la aplicación o sistema de software.

El proceso de obtención de los requerimientos de usuarios se ha realizado a través de talleres grupales con los stakeholders, identificando las necesidades sobre las características principales que debe de tener para cumplir necesidades de mejora en la gestión de administración de eventos.

1.5.1. Requerimiento del usuario

Las funcionalidades que los usuarios necesitan, así como sus necesidades sobre el proyecto:

RU-01: El organizador debe poder manipular el evento desde el panel de administración para tener el control sobre el proceso del evento.

- RU-02:** El organizador debe poder manipular la administración de entradas al evento desde el panel de administración para asegurar una experiencia exitosa.
- RU-03:** El organizador debe poder enviar una notificación vía e-mail a los participantes inscritos al evento para mantenerlos informados.
- RU-04:** El organizador debe poder obtener un reporte de asistencia y una encuesta de satisfacción del evento para evaluar el impacto después del evento.
- RU-05:** El organizador debe poder obtener un listado de participantes para el control de acceso.
- RU-06:** El organizador debe poder manipular la postulación de ponencias, sean charlas o talleres, para tener un control sobre el proceso.
- RU-07:** El organizador debe poder aceptar o rechazar ponencias al evento desde el panel de administración para seleccionar las ponencias adecuadas.
- RU-08:** El organizador debe poder notificar vía e-mail a los usuarios que sus charlas han sido aceptadas o rechazadas.
- RU-09:** El organizador debe poder asignar el rol de expositor a los usuarios que tengan charlas aceptadas.
- RU-10:** El organizador debe poder manipular la agenda del evento para planificar el tiempo y los espacios disponibles durante la duración del evento.
- RU-11:** El organizador debe poder manipular distintos niveles de patrocinio con su debida descripción.
- RU-12:** El organizador debe poder asignar una cantidad de notificaciones por e-mail a un nivel de patrocinio como beneficio a los patrocinadores.
- RU-13:** El organizador debe poder asignar el rol de patrocinador a un usuario que represente al patrocinador.
- RU-14:** El organizador debe poder asignar el rol de coorganizador del evento a un representante usuario de la institución.
- RU-15:** El organizador debe poder asignar una cantidad de notificaciones por e-mail al coorganizador.
- RU-16:** El organizador debe poder manipular una encuesta para obtener feedback de los participantes del evento.
- RU-17:** El organizador debe poder manipular su perfil.
- RU-18:** El organizador debe poder ver la cantidad de inscritos para asistir al evento y así planificar decisiones estratégicas.
- RU-19:** El coorganizador debe poder enviar notificaciones vía e-mail a los participantes inscritos para mantenerlos informados.
- RU-20:** El coorganizador debe poder obtener un reporte de asistencia y una encuesta de satisfacción del evento para evaluar el impacto después del evento.
- RU-21:** El coorganizador debe poder manipular su perfil.
- RU-22:** El coorganizador debe poder ver la cantidad de inscritos para asistir al evento y así planificar decisiones estratégicas.
- RU-23:** El usuario debe poder obtener o anular una entrada al evento.
- RU-24:** El participante debe poder recibir notificaciones por e-mail respecto al evento.
- RU-25:** El usuario debe poder manipular su cuenta por motivos de privacidad.

RU-26: El organizador y coorganizador deben de poder hacer un registro de los participantes cuando llegan al lugar del evento.

RU-27: El participante debe poder contestar encuestas para dejar su opinión sobre el evento.

RU-28: El expositor debe poder manipular borradores de charlas o talleres para postular al evento.

RU-29: El expositor debe poder manipular su perfil.

RU-30: El expositor debe poder cancelar su charla por motivos personales.

RU-31: El patrocinador debe poder manipular su perfil.

RU-32: El patrocinador debe poder enviar notificaciones por e-mail para difundir información de su marca.

1.6. Alternativas globales de solución

Analizan y evalúan tres alternativas de solución para el proyecto con el objetivo de seleccionar la mejor.

1.6.1. Alternativa 1. Optimizar procesos que ya se tienen mediante scripts

La situación actual son procedimientos ejecutados manualmente por el equipo organizador del evento, cuyos elementos es un sitio web estático en el cual se agrega la información manualmente obtenida del servicio Sessionize para la convocatoria de envió de presentaciones, Sessionize también se utiliza para las votaciones de las presentaciones y Eventbrite para la gestión de entradas. Secciones como patrocinadores son actualizadas manualmente en el HTML del sitio web estático.

Por medios de scripts se automatizarán los procesos manuales, obteniendo como beneficios confiabilidad en los datos publicados, aumento de la productividad en el ahorro de tiempo al ya no realizar un procedimiento manual.

Script de automatización de votación de presentaciones: Sessionize permite enviar una invitación cada vez, por lo cual un script tomaría los datos de contacto de los participantes inscritos desde un documento descargado de Eventbrite y enviaría las invitaciones a través de Sessionize solucionando la automatización del formulario de votos y asegurando que los votos

son reales, pero afectando la usabilidad de los participantes al tener que estar inscrito en dos sitios diferentes con idiomas diferentes.

Script de automatización de cronograma del evento: Sessionize permite bajar un archivo con la lista de los expositores aprobados, por lo cual el script tomaría los datos de los expositores y generaría el cronograma automáticamente integrando la información en la plantilla HTML.

Script de optimización de publicación secciones: Los datos serán escrito en un archivo aparte y mediante un script de Python se actualizarán en la sección correspondiente en el sitio.

La mantención de los scripts es de una complejidad baja, pero con cambios que pueden suceder en las actualizaciones de librerías que se utilicen al desarrollarlos, también deben de ser mantenidos por cambios de en el api de los servicios.

La escalabilidad de los scripts depende del diseño y del alcance de estos, por lo cual se pueden optimizar y agregar pequeñas mejoras al código.

Esta alternativa no contempla todos los requerimientos, solo es una mejora de los procesos actuales por lo cual no cumple con las expectativas de los usuarios. Tiene un costo estimable de un analista programador con un costo monetario estimado por experiencia de 500 dólares.

1.6.2. Alternativa 2. Contratar servicios administración de eventos de terceros

La segunda propuesta es la contratación de un servicio que se encuentre en la nube que cumpla con la mayor parte de los requerimientos de la organización.

Hubilo es una plataforma web para la gestión completa de eventos de cualquier tipo como conferencia, talleres, seminarios, hackathon etc. sean de manera virtual o presencial, cubre la funcionalidad completa de un evento. Los participantes pueden obtener su boleto de entrada, se pueden realizar cronogramas de las presentaciones, hacer reportes automatizados etcétera. Es una de las plataformas en línea que más funcionalidades tiene.

Si bien tiene muchas funcionalidades no hay una escalabilidad personalizada de la aplicación y sobre la mantención solo se pueden reportar incidencias esto es debido por la naturaleza privada de la solución.

Al ser un servicio de tercero brindado por una empresa se espera grandes estándares de seguridad, pero sin embargo no hay información pública sobre la misma y no hay un control sobre la información.

Tiene un costo aproximadamente de \$1400 USD por evento para 200 participantes, teniendo en consideración que se utilizaría en reiteradas ocasiones. www.hubilo.com

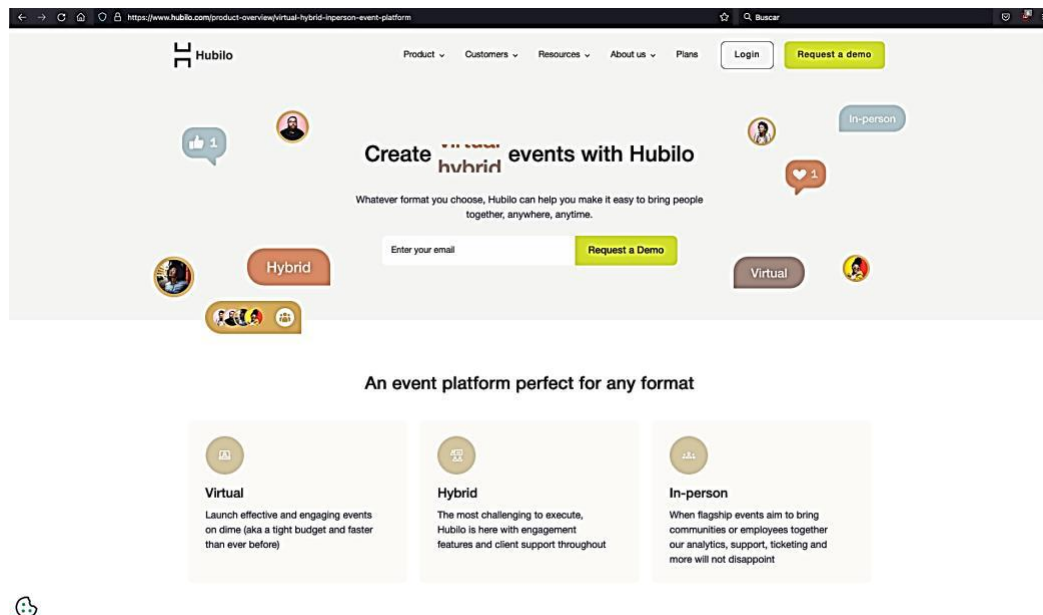


Figura 1-10. Imagen del Servicio Hubilo
Fuente: Elaboración Propia

1.6.3. Alternativa 3. Desarrollo a la medida a las necesidades del cliente

La siguiente alternativa es una solución es el diseño, desarrollo e implementación de un software de gestión de eventos, sin embargo, solo se llevará a cabo el diseño en este documento y la etapa de desarrollo e implementación son nombradas para esclarecer la alternativa.

La gran ventaja de la realización de un software de gestión de eventos a medida es que está diseñado para cumplir con los requerimientos de los usuarios, lo que implica que tendrá todas las funcionalidades y características necesarias para cubrir las necesidades de gestión de la organización.

Al ser una opción una opción diseñada para el cliente la escalabilidad y mantención del código y con el correcto diseño del software se puede ampliar en funcionalidades y en mantención del código.

La usabilidad es unos de los requerimientos que toda aplicación debe de tener hoy en día como meta, ya que la experiencia que tenga el usuario sobre la aplicación será la opinión que se forme sobre ella, al ser un desarrollo un especialista de UX puede diseñar interacciones que logren esta meta.

Desde la perspectiva de la seguridad de los datos se seguirán las estipulaciones de la legislación vigente y se consideran las buenas prácticas técnicas para garantizar que sea seguro y que solo las personas indicadas puedan acceder a los datos.

Se estima que el diseño, desarrollo e implementación tendrá un costo aproximado de \$550 UF a \$650 UF.

1.7. Criterios para evaluar las alternativas

A continuación, se define los criterios de evaluación de las alternativas para evaluar cual es la más atractiva de ser seleccionada.

- a. **Funcionalidad:** Hace referencia competencia de llevar a cabo un conjunto de funciones que cumplen con las necesidades de los usuarios.
- b. **Usabilidad:** Facilidad de uso que tienen los usuarios al interactuar, dependiendo del nivel de conocimiento del uso del usuario.
- c. **Mantenición:** Refiere a la facilidad de que el producto de software pueda mantenerse después de su implementación, por ejemplo, en la corrección de errores.
- d. **Escalabilidad:** Es la capacidad del software de añadir nuevas funcionalidades y características para ampliar su funcionamiento según la necesidades futuras de los usuarios.
- e. **Seguridad de los datos:** Es la capacidad de salvaguardar los datos de destrucción, de robo de información, filtraciones, disponibilidad y veracidad de los datos.
- f. **Costo:** Es la suma de los valores monetarios que incluye los gastos en diseño, desarrollo e implementación si es necesario de la alternativa.

1.7.1. Ponderación de criterios y escala de evaluación

Se utiliza una ponderación según el nivel de importancia ponderación de criterio, teniendo en consideración

Tabla 1.1. Ponderación de los criterios.

CRITERIO	PONDERACIÓN
Funcionalidad	25%
Usabilidad	15%
Mantenición	10%
Escalabilidad	15%
Seguridad de los datos	15%
Costo	20%

Fuente: Elaboración Propia

Se establece la siguiente escala de evaluación de los criterios.

Tabla 1.2. Escala de evaluación de los criterios.

CALIFICACIÓN	CONCEPTO
0	No Aplica
1	Muy Malo
2	Malo
3	Regular
4	Bueno
5	Muy Bueno

Fuente: Elaboración Propia

1.7.2. Evaluación

Se presentan las siguientes evaluaciones de cada una de las alternativas globales de solución, según el criterio y escala específica.

Tabla 1.3. Alternativa 1. Calificación de Optimizar procesos mediante scripts

CRITERIO	NOTA	DESCRIPCIÓN
Funcionalidad	2	Cumple con un aproximado del 20% de las funcionalidades
Usabilidad	3	Mejora algunos de los procesos usabilidad
Mantenición	4	Los scripts se pueden modificar y adaptar.
Escalabilidad	2	Los scripts se pueden mejorar para procesar más rápido los datos. Por funcionalidad no hay mucho más que se pueda agregar por scripts.
Seguridad de los datos	2	Los datos no se cifran para ser utilizados en los scripts
Costo	5	Los costos de desarrollo e implementación son menos

Fuente: Elaboración Propia

Tabla 1.4. Alternativa 2. Calificación de Contratar servicios de terceros

CRITERIO	NOTA	DESCRIPCIÓN
Funcionalidad	4	Cumple con un aproximado del 70% de las funcionalidades
Usabilidad	4	Solo está en inglés, pero es de fácil uso
Mantenición	0	No es necesaria la mantención.
Escalabilidad	1	No hay personalizaciones del producto para clientes específicos.
Seguridad de los datos	4	Los datos son almacenados por en la aplicación
Costo	1	El costo se vuelve demasiado elevado con el uso recurrente

Fuente: Elaboración Propia

Tabla 1.5. Alternativa 3. Calificación del Desarrollo a la medida de las necesidades del cliente

CRITERIO	NOTA	DESCRIPCIÓN
Funcionalidad	5	Cumple con un aproximado del 100% de las funcionalidades
Usabilidad	4	Cumple las expectativas de facilidad de uso
Mantenición	4	Se deben de realizar mantenciones y mejoras
Escalabilidad	4	Es abierto a nuevas funcionalidades
Seguridad de los datos	5	Los datos están protegidos
Costo	3	Tiene un costo alto al principio, pero a largo plazo es viable

Fuente: Elaboración Propia

Tabla 1.6. Resumen de Calificación de Alternativas

Criterio	Ponderación	ALTERNATIVAS					
		Calificación			Calificación Ponderada		
		1	2	3	1	2	3
Funcionalidad	25%	2	4	5	0.5	1	1.25
Usabilidad	15%	3	4	4	0.45	0.6	0.6
Mantenimiento	10%	4	0	4	0.4	0	0.4
Escalabilidad	15%	3	2	4	0.45	0.3	0.6
Seguridad de los datos	15%	2	4	5	0.3	0.6	0.75
Costo	20%	5	1	3	1.0	0.2	0.6
Total	100%	19	15	25	3.1	2.7	4,2

Fuente: Elaboración Propia

Según los datos obtenidos al crear la tabla, la primera alternativa, 'Optimizar procesos mediante scripts', es una opción de mejora que se ajusta bien a presupuestos bajos y es fácil de mantener. La segunda alternativa, 'Contratar servicios de terceros', no es ideal debido a su alto costo y baja personalización. La tercera alternativa, 'Desarrollo a la medida de las necesidades del cliente', ha sido seleccionada por su costo moderado, gran escalabilidad y mayor funcionalidad en comparación con las otras opciones.

1.8. Alternativa Seleccionada

La selección de la alternativa es la que obtuvo mejor puntaje análisis.

1.8.1. Descripción y beneficios de la alternativa seleccionada

La alternativa número uno de mejorar los procesos actuales no contempla todas las funcionalidades necesarias que piden los usuarios según los requerimientos, es solo la mejora de lo que ya está implementado, no cambia el hecho de que los participantes se deben de inscribir en sitios de terceros, si se cuenta con un presupuesto limitado sería viable su elección, no se selecciona debido a que no cumple con los requerimientos funcionales.

La segunda alternativa de contratación de servicio de tercero cuenta con muchas funcionalidades y una buena usabilidad, pero los costos a ser contratada recurrentemente encarecen su uso, además la solución no es escalable es decir que no se puede agregar ni cambiar sus funcionalidades, al igual que la alternativa anterior la información de los participantes queda en un sitio de tercero que no pertenece a la Corporación Python Chile, por lo cual esta alternativa también es descartada.

La tercera alternativa "Desarrollo a la medida de las necesidades del cliente" con la mayor puntuación es seleccionada, ya que, al estar diseñada con los requerimientos y requisitos de los usuarios y organización, permite que solución sea escalable en términos de usuarios y de producto, considerando también que los datos estarán seguros ya que la base de datos queda a cargo de la Corporación Python Chile. Unos de los criterios más importante es la

estimación de costo del valor monetario, aunque es alto para los presupuestos de la corporación Python Chile es asumible ya que representa un ahorro monetario a largo plazo.

El propósito de la alternativa seleccionada es "optimizar la gestión de eventos mediante la implementación de un sistema web". Con esto se quiere llevar a cabo exitosamente la ejecución de la conferencia y obteniendo información relevante para futura tomas de decisiones

Beneficios:

1. Autonomía de uso
2. Mayor seguridad de los datos
3. Al utilizar una sola solución se simplifican los procesos.
4. Generación automática de información
5. Los procesos están mejor definidos
6. Mayor automatización de los procesos y menor error humano.
7. Capitalización de los usuarios inscritos.

1.8.2. Desarrollo del plan de personal

Se contará con un equipo de trabajo conformados por profesionales del área del desarrollo de productos de software, para la realización de este proyecto:

- **Un jefe de proyecto (JP):** Es el profesional encargado de la gestión del equipo y de la gestión del proyecto.
- **Un analista Programador (AP):** Es el profesional encargado de analizar los requisitos y requerimientos del usuario y de la organización y de desarrollar la lógica y las funcionalidades.
- **Un desarrollador Frontend (DF):** Es el profesional encargado de desarrollar las interfaces graficas con los cuales los usuarios interactúan.
- **Un analista de QA (QA):** Es el profesional encargado del control de calidad del producto del software asegurándose que funcione correctamente.
- **Un ING de Software (IS):** Profesional encargado de desarrollar y mantener la calidad del software.
- **Un especialista UX/UI:** Profesional encargado de la experiencia de usuario

1.8.3. Planificación temporal y asignación de recursos por actividad

Abarca las estrategias de gestión de tiempo y los recursos de manera eficiente, con una metodología de desarrollo en cascada.

Tabla 1.7. Resumen de la Programación de Actividades.

Fase	Tarea	Duración (días)	Recursos
Proyecto sistema gestión de eventos		121	
Fase 0	Iniciación	5	
	Reunión de lanzamiento	1	JP
	Reunión con Stakeholders	2	AP, IS
	Definición del alcance	1	JP, AP, IS
	Identificación de objetivos del proyecto	1	JP, AP, IS
Fase 1	Planificación	10	
	Elaboración de cronograma	4	JP, AP-25%
	Plan de gestión de riesgos	2	JP, AP
	Asignación de recursos	2	JP
	Plan de control de calidad	2	JP, QA
Fase 2	Especificación de Requerimiento	14	
	Recopilación de requerimientos	3	AP
	Análisis de requerimientos	3	AP, IS
	Modelado del sistema	5	AP
	Validación de requerimientos	2	AP
	Revisión y aprobación final	1	JP, AP
Fase 3	Diseño	18	
	Diseño de arquitectura	3	IS
	Diseño de datos	10	AP
	Diseño de UI/UX	4	UX, DF-25%
	Revisión y aprobación final	1	JP
Fase 4	Implementación	44	
	Desarrollo del Backend	20	AP, IS
	Desarrollo del Frontend	20	DF, AP-5%, UX-10%
	Pruebas funcionales preliminares	3	QA
	Revisión y aprobación final	1	JP, AP, IS
Fase 5	Pruebas	22	
	Pruebas funcionales	3	QA
	Pruebas no Funcionales	2	QA
	Pruebas de aceptación	5	JP-40%, QA
	Ajustes y corrección de errores	10	AP, DF, QA-10%
	Documentación Final	2	JP, AP, UX, QA
Fase 6	Despliegue	6	
	Despliegue a producción	2	AP
	Test y Monitoreo del sistema	2	AP, QA
	Solución de incidencias	2	AP, IS
Fase 7	Cierre	2	
	Revisión final del proyecto	1	JP
	Entrega final	1	JP

Fuente: Elaboración propia.

1.8.4. Estimación de costo

La estimación de costo del proyecto se realiza con el método de estimación por esfuerzo, esta técnica se utiliza los costos mensuales de la empresa y de la planificación, distribuyendo los recursos para realizar la estimación del costo del proyecto.

La tabla 1-8 "Costo de empresa mensual" muestra los costos fijos (CF), los costos variables (CV) y los gastos de administración y venta (GAV). La sumatoria del total de los costos fijos y el total de gasto de administración y venta, da como resultado el "Costo empresa (CE)" de 270UF. Los costos variables se le asignan al proyecto y son costos para el cliente.

Tabla 1.8. Costo empresa mensual

Costo Fijo (CF)		GAV		CE	Costo Variable (CV)	
Ítem	Precio (UF)	Ítem	Precio (UF)	UF	Ítem	Precio (UF)
JP	53	Vendedor	30	-	QA	32
AP	39			-	UX	27
DF	35			-		
IS	50			-		
Otros	63			-		
Total	240	Total	30	270		

Fuente: Elaboración propia

En la Tabla 1-9 "costo proyecto" La primera columna "Recursos" informan que recursos trabajan en el proyecto por tipo y rol. La segunda columna es la sumatoria de los "costos" fijos de los recursos que se utilizan en el proyecto (177 UF), los recursos variables no se suman, la tercera columna calcula el "porcentaje" con respecto al total calculado previamente de los recursos fijos. La cuarta columna "Distribución costo por recurso" reparte el valor de costo empresa calculado anterior entre los recursos fijos.

La quinta columna es "Valor día" calcula el costo del recurso por día dividiendo costo ya distribuido del recurso calculado anteriormente por el promedio de los días hábiles que se trabaja en el mes, en este caso es de 20 días. Si el recurso es variable se divide el costo variable.

La sexta columna "Días trabajado en el proyecto" es la sumatoria de los días trabajados en el proyecto para cada recurso, esta información se obtiene de la sección "Planificación temporal y asignación de recurso por actividad" específicamente de la Tabla 1-7 "Carta Gantt Planificación de actividades".

En la séptima columna "Costo para empresa (CPE)" es la sumatoria de la multiplicación del "valor día" por "Días trabajado en el proyecto" de todos los recursos fijo.

Tabla 1.9. Costo proyecto

Recurso		Costo (UF)		%	Reparto costo por recurso (UF)	Valor día (20 días)	Días trabajado en el proyecto	CPE	CV	Total a cobrar (UF)	
Tipo	Rol	Fijo	Var							CPC	Riesgo
Fijo	JP	53	-	29,9%	80,8	4,0	22	88,9	-	CPC	556,9
Fijo	AP	39	-	22,0%	59,5	3,0	71	211,2	-		
Fijo	DF	35	-	19,8%	53,4	2,7	33	88,1	-	Utilidad 30%	167,07
Fijo	IS	50	-	28,2%	76,3	3,8	33	125,8	-		
Var	QA	-	32	-	-	1,6	20	-	32	Riesgo 5%	27,84
Var	UX	-	27	-	-	1,35	8	-	10,8		
Total		177	-	100%	(CE) 270	-	-	514,1	42,8	Total	751,8

Fuente: Elaboración propia

La última columna "total a cobrar", la fila CPC (costo para el cliente) es la suma de CPE y CV, a este resultado de 556,9 UF se calcula el porcentaje de ganancia de 30% y un riesgo de 5% y se suman a CPC para obtener el valor de venta mínimo para el cliente 751,8 UF.

1.8.5. Administración del riesgo

El riesgo es todo lo que pudiera pasar en el transcurso del proyecto que afecte negativa y significativamente al proyecto alterando el calendario, calidad del software y en el peor de los casos al fracaso del proyecto. Por lo cual la administración del riesgo adquiere importancia estratégica en la gestión del proyecto, bajando el nivel de incertidumbre siendo proactivos antes que sucedan los problemas, evitándolos o planificando contingencias.

La administración de riesgo requiere cuatro etapas, la identificación del riesgo, análisis del riesgo, planeación del riesgo y monitorización del riesgo.

En la tabla 1-10 "administración del riesgo" se identifican los riesgos que podrían suceder en el transcurso del proyecto. Se realiza un análisis valorando la probabilidad de que suceda, el impacto de retiro y el costo de retiro, del análisis podemos concluir una priorización de los riesgos para ser retirado, dada la siguiente ecuación. La prioridad de retiro es de menor a mayor.

Los valores de posibilidad, impacto y costo de retiro van en la escala de valores 1 a 10.

$$\text{Cálculo de prioridad} = (11-\text{posibilidad}) * (11-\text{impacto}) * \text{Costo Retiro}$$

En la tabla 1-11 "Plan de mitigación", los riesgos se reordenan según la prioridad de retiro, se identifica la señal de alerta y se desarrollan estrategias de mitigación proactiva y reactiva.

En la etapa de monitorización de riesgo es una revisión continua durante cada etapa del proyecto, en la cual de manera proactiva se va revisando y actualizando la lista de riesgo.

Tabla 1.10. Tabla Administración del Riesgo

Ítems	ID	Riesgo	Posibilidad	Impacto	Costo de retiro	Prioridad	Señal de alerta	Responsable
Producto Software	0	Aumento del alcance del proyecto	4	4	3	147	El cliente solicita más características durante el desarrollo.	Jefe de Proyecto
Producto software	1	Librerías de terceros podrían quedar sin soporte	3	5	8	384	Aviso de finalización de soporte.	Analista Programador
Tecnología	2	Poca experiencia en el uso de librerías	4	4	5	245	El desarrollador indica que no tiene experiencia con la librería.	Jefe de Proyecto
Tecnología	3	Interfaz poco intuitiva para los usuarios	2	2	3	243	Se ha detectado que la interfaz actual dificulta el uso, afectando la experiencia del usuario y la eficiencia.	Especialista UX/UI
Proyecto	4	Pérdida de personal crítico	4	8	7	147	Un ingeniero avisa que cambiará de organización.	Jefe de proyecto
Proyecto	5	Mala planificación estratégica	2	8	3	81	Los hitos del proyecto no se están cumpliendo a tiempo.	Jefe de Proyecto
Proyecto	6	Incidentes que impidan el traslado del personal operativo del proyecto	1	8	9	270	Posibles interrupciones operativas inesperadas que afectan el normal traslado del personal	Empresa
Organización	7	Generación de Ambientes laborales hostiles	4	7	4	112	Se han detectado factores que podrían afectar el clima organizacional y el bienestar del equipo.	Jefe de Proyecto
Organización	8	La empresa quiebra debido a la falta de pagos.	2	10	2	18	La falta de pagos sostenida podría llevar a la empresa a una situación de quiebra.	Empresa

Fuente: Elaboración propia

Tabla 1.11. Plan de Mitigación

Riesgo	ID	Prioridad	Señal de alerta	Mitigación preventiva	Mitigación reactiva
La empresa quiebra debido a la falta de pagos.	8	18	La falta de pagos sostenida podría llevar a la empresa a una situación de quiebra.	Reunión con el cliente sobre los términos, las condiciones del pago y garantía de pago.	Iniciar el proceso de cobranza de garantía.
Mala planificación estratégica	5	81	Los hitos del proyecto no se están cumpliendo a tiempo.	Analizar la situación en la que están atrasados y por qué; realizar una planificación para cumplir los plazos.	Priorizar características y aumentar las horas laborales
Generación de Ambientes laborales hostiles	7	112	Se han detectado factores que podrían afectar el clima organizacional y el bienestar del equipo.	Realizar reuniones periódicas con el personal y llevar a cabo ejercicios de resolución de conflictos	Implementar mediaciones especializadas con terceros y proceder con la reasignación del personal a nuevos proyectos.
Aumento del alcance del proyecto	0	147	El cliente solicita más características durante el desarrollo.	Realizar reuniones periódicas con el cliente para explicar que, una vez finalizado el proyecto, este podrá ser actualizado según sus necesidades	Reasignar al personal de desarrollo a los proyectos correspondientes, y llevar a cabo una replanificación y priorización de las funcionalidades
Pérdida de personal crítico	4	147	El desarrollador indica que no tiene experiencia con la librería.	Establecer una negociación con el ingeniero con el objetivo de asegurar su continuidad en el equipo.	Incorporar un nuevo ingeniero al equipo y redistribuir las responsabilidades entre los miembros para optimizar el desempeño del proyecto.
Interfaz poco intuitiva para los usuarios	3	243	Se ha detectado que la interfaz actual dificulta el uso, afectando la experiencia del usuario y la eficiencia.	Incluir prototipos interactivos y validaciones tempranas con stakeholders	Publicar una guía rápida de uso o tutoriales para facilitar la navegación mientras se rediseña la interfaz.
Poca experiencia en la utilización de librerías	2	245	El desarrollador indica que no tiene experiencia con la librería.	Brindar apoyo al desarrollador mediante cursos o libros especializados sobre la librería en uso	Contratar a un consultor externo para apoyar el proceso de aprendizaje del equipo
Incidentes que impidan el traslado del personal operativo del proyecto	6	270	Posibles interrupciones operativas inesperadas que afectan el normal traslado del personal	Implementar el teletrabajo como modalidad de apoyo para el equipo.	Implementar extensión de horario de trabajo para recuperar el tiempo perdido.
Librerías de terceros podrían quedar sin soporte	1	384	Aviso de finalización de soporte.	Verificar calendarios de soporte de librerías y, si está en uso, planificar un cambio.	Buscar un reemplazo o fork de la librería.

Fuente: Elaboración propia

**CAPÍTULO 2. ASPECTOS RELEVANTES DEL ANÁLISIS PARA LA ALTERNATIVA
SELECCIONADA**

2. ASPECTOS RELEVANTES DEL ANÁLISIS

En este capítulo se presenta aspectos relevantes del análisis para la alternativa seleccionada.

2.1. Descripción de la solución propuesta

El proyecto consiste en el desarrollo de un software de gestión de eventos a medida, diseñado específicamente para satisfacer las necesidades de la Corporación Python Chile. La alternativa seleccionada, "Desarrollo a la medida de las necesidades del cliente", ha sido elegida debido a su alta puntuación en términos de adaptación a los requerimientos de usuarios y necesidades de la organización.

Esta solución permite una escalabilidad tanto en el número de usuarios como en las funcionalidades del producto, garantizando a su vez la seguridad de los datos al estar la base de datos gestionada por la Corporación Python Chile. Aunque el costo inicial es elevado, se justifica por el ahorro monetario a mediano plazo.

El objetivo principal de esta solución es optimizar la gestión de eventos mediante la implementación de un sistema web. Este sistema permitirá llevar a cabo la ejecución de eventos de manera eficiente y recoger información relevante para futuras decisiones estratégicas.

Beneficios de la Solución:

1. **Autonomía de Uso:** Los usuarios podrán operar el sistema de manera independiente, sin necesidad de asistencia constante.
2. **Mayor Seguridad de los Datos:** La administración de la base de datos por la Corporación Python Chile asegura la protección de la información.
3. **Simplificación de Procesos:** Utilizar una única solución integrada simplifica los procesos administrativos y operativos.
4. **Generación Automática de Información:** El sistema generará informes y datos relevantes de manera automática, facilitando el análisis y la toma de decisiones.
5. **Procesos Mejor Definidos:** Los flujos de trabajo estarán claramente delineados, lo que mejorará la eficiencia operativa.
6. **Mayor Automatización y Menor Error Humano:** La automatización de tareas reduce la posibilidad de errores humanos y aumenta la precisión.
7. **Capitalización de Usuarios Inscritos:** El sistema permitirá gestionar y aprovechar mejor la base de datos de usuarios inscritos.

El sistema de gestión de eventos incluirá las siguientes características:

- Crear un evento.
- Crear un cronograma del evento.
- Definir categorías de patrocinios y sus beneficios.
- Generar convocatorias para postulaciones de presentaciones.
- Enviar notificaciones a los participantes y organizadores.
- Crear las entradas para el evento.
- Permitir a los Coorganizadores crear y gestionar sus perfiles.
- Enviar notificaciones a los participantes.
- Permitir a los usuarios inscritos postular presentaciones para el evento.
- Crear perfiles de presentadores.
- Obtener y cancelar entradas para el evento.
- Permitir a los patrocinadores crear y gestionar sus perfiles.

2.2. Actores del Sistema

Tabla 2.1. Actores del sistema.

Actores	Descripción	Tipo De actor	Usuarios
Organizador	Es el encargado de la gestión del evento.	Primario del sistema	N
Coorganizador	Es el colaborador del evento, tiene menos responsabilidad.	Primario del sistema	N
Patrocinador	Aporta apoyo financiero al evento a cambio de reconocimiento y publicidad.	Primario del sistema	N
Expositor	Son las personas que presentarán, ponencias, charlas, talleres etc.	Primario del sistema	N
Participante	Son las personas que asisten al evento.	Primario del sistema	N
Usuario	Son las personas que tienen una cuenta en el sistema.	Primario del sistema	N
Tiempo	Es el sistema que a su debido tiempo ejecuta casos de uso	Secundario del sistema	1

Fuente: de elaboración propia

2.3. Diagrama de actividades general UML

Diagrama de actividad general del proceso administrativo de creación de evento.

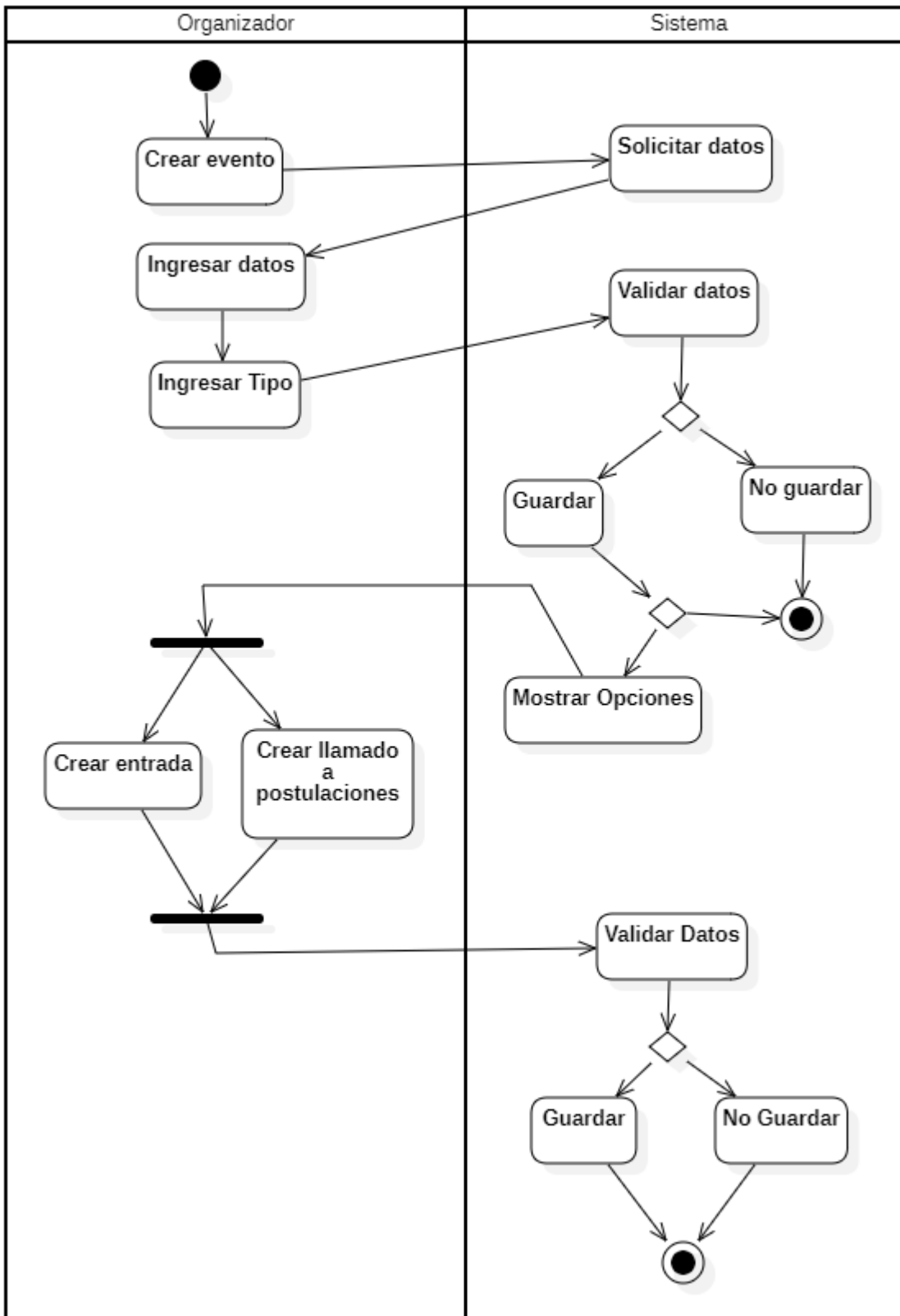


Figura 2-1. Diagrama de actividades
Fuente: Elaboración Propia

2.4. Requerimientos del sistema

Los requerimientos del sistema en esta etapa son importante debido que es en donde se analizan y especifican los requerimientos funcionales y no funcionales.

2.4.1. Requerimientos funcionales

A continuación, se detallan los requerimientos funcionales asociados al sistema de gestión de eventos.

- RF-01:** El sistema debe de permitir al organizador mantener un evento con los siguientes datos; titulo, descripción, fecha inicio, fecha final, duración, imagen y tipo de evento.
- RF-02:** El sistema debe de permitir al organizador mantener entradas para el evento con una fecha de inicio y fin para obtenerlas.
- RF-03:** El sistema debe de permitir al organizador, coorganizador, patrocinador enviar notificación al mail de los participantes del evento con información relevante en una fecha y hora determinada.
- RF-04:** El sistema debe permitir a los organizadores y coorganizadores generar reportes detallados del evento, que incluyan tanto los datos de asistencia como las estadísticas de las encuestas realizadas.
- RF-05:** El sistema debe de permitir al organizador mantener una encuesta con la siguiente información título, descripción y preguntas.
- RF-06:** El sistema debe de permitir al organizador mantener una postulación ponencia con los datos fecha de inicio y fin de postulación, tipo de ponencia, titulo y resumen.
- RF-07:** El sistema debe de permitir al organizador asignar el rol de expositor, coorganizador, o patrocinante al usuario.
- RF-08:** El sistema debe de permitir al organizador mantener una agenda con la fecha, horario y estado.
- RF-09:** El sistema debe de permitir al organizador mantener nivel patrocinio con su título, descripción y beneficio cantidad de notificaciones a enviar vía correo electrónico.
- RF-10:** El sistema debe de permitir al organizador asignar una cantidad de notificaciones por mail a un coorganizador y patrocinador.
- RF-11:** El sistema debe de permitir al organizador, coorganizador, patrocinador y expositor mantener su perfil con los siguientes datos, nombre del organizador, descripción y logo.

- RF-12:** El sistema debe de permitir al organizador y coorganizador ver la cantidad de entradas adquiridas.
- RF-13:** El sistema debe de permitir al usuario obtener una entrada con los datos del evento, con código QR, numero etc.
- RF-14:** El sistema debe de permitir al participante cancelar una entrada.
- RF-15:** El sistema debe de permitir a los usuarios manipular su cuenta con los datos de su contraseña como confirmación.
- RF-16:** El sistema debe de permitir al organizador y coorganizador registrar ingresos participantes comprobando con su Rut, nombre o entrada.
- RF-17:** El sistema debe de permitir al participante contestar las preguntas de la encuesta.
- RF-18:** El sistema debe de permitir al usuario mantener una ponencia con los siguiente datos tipo, titulo, descripción, resumen.
- RF-19:** El sistema debe de permitir al expositor cancelar su ponencia aceptada.

2.4.2. Requerimientos no funcionales

A continuación, se detallan los requerimientos no funcionales asociados al sistema de gestión de eventos.

Tabla 2.2. Requisitos no funcionales

ID	Descripción	Forma de verificación
RNF-001	El sistema debe de soportar una cantidad de usuarios de al menos de 100 a la vez.	Se verificará a través de pruebas de carga del sistema a través de Apache Jmeter. Midiendo el tiempo de respuesta que no debe de ser mayor de 3000 ms.
RNF-002	El sistema debe ser visualmente atractivo, alcanzando un puntaje promedio superior a 1.5 en las dimensiones de Atractivo y Estimulación del cuestionario UEQ.	Se aplicará el cuestionario UEQ a un grupo de prueba, evaluando que Atractivo y Estimulación superen el valor de 1.5 en la escala de -3 a +3.
RNF-003	El sistema debe de ser mínimo 60 puntos en la métrica de usabilidad de sistema (SUS)	Después de una capacitación introductoria de una hora, los usuarios responderán una encuesta y el cálculo SUS debe de ser mayor de 60.
RNF-004	El sistema debe de estar disponible un 95% antes y durante el evento.	Se usará un sistema de monitorización en la nube UptimeRobot, para verificar la disponibilidad del sistema.

Fuente de elaboración propia

2.5. Estructura función del sistema

Se presenta caso de uso general del sistema de gestión de eventos. No se consideran los casos de uso de autenticación de usuarios, ni los casos de uso relacionados al actor Administrador, aunque están considerado dentro del sistema. Por temas de espacios los actores del lado derecho son actores primarios.



Figura 2-2. Caso de uso general
Fuente: Elaboración Propia

2.6. Matriz de trazabilidad

La siguiente matriz muestra la completitud de los requerimientos funcionales con respecto a qué caso de uso les da su cumplimiento.

Tabla 2.3 matriz de trazabilidad

ID	Casos de uso																		
	0 1	0 2	0 3	0 4	0 5	0 6	0 7	0 8	0 9	1 0	1 1	1 2	1 3	1 4	1 5	1 6	1 7	1 8	1 9
RF-01	x																		
RF-02		x																	
RF-03			x																
RF-04				x															
RF-05					x														
RF-06						x													
RF-07							x												
RF-08								x											
RF-09									x										
RF-10										x									
RF-11											x								
RF-12												x							
RF-13													x						
RF-14														x					
RF-15															x				
RF-16																x			
RF-17																	x		
RF-18																		x	
RF-19																			x

Fuente: elaboración propia

2.7. Modelo conceptual

Se presenta el siguiente diagrama conceptual del sistema de gestión de eventos que representa un punto de vista del análisis.

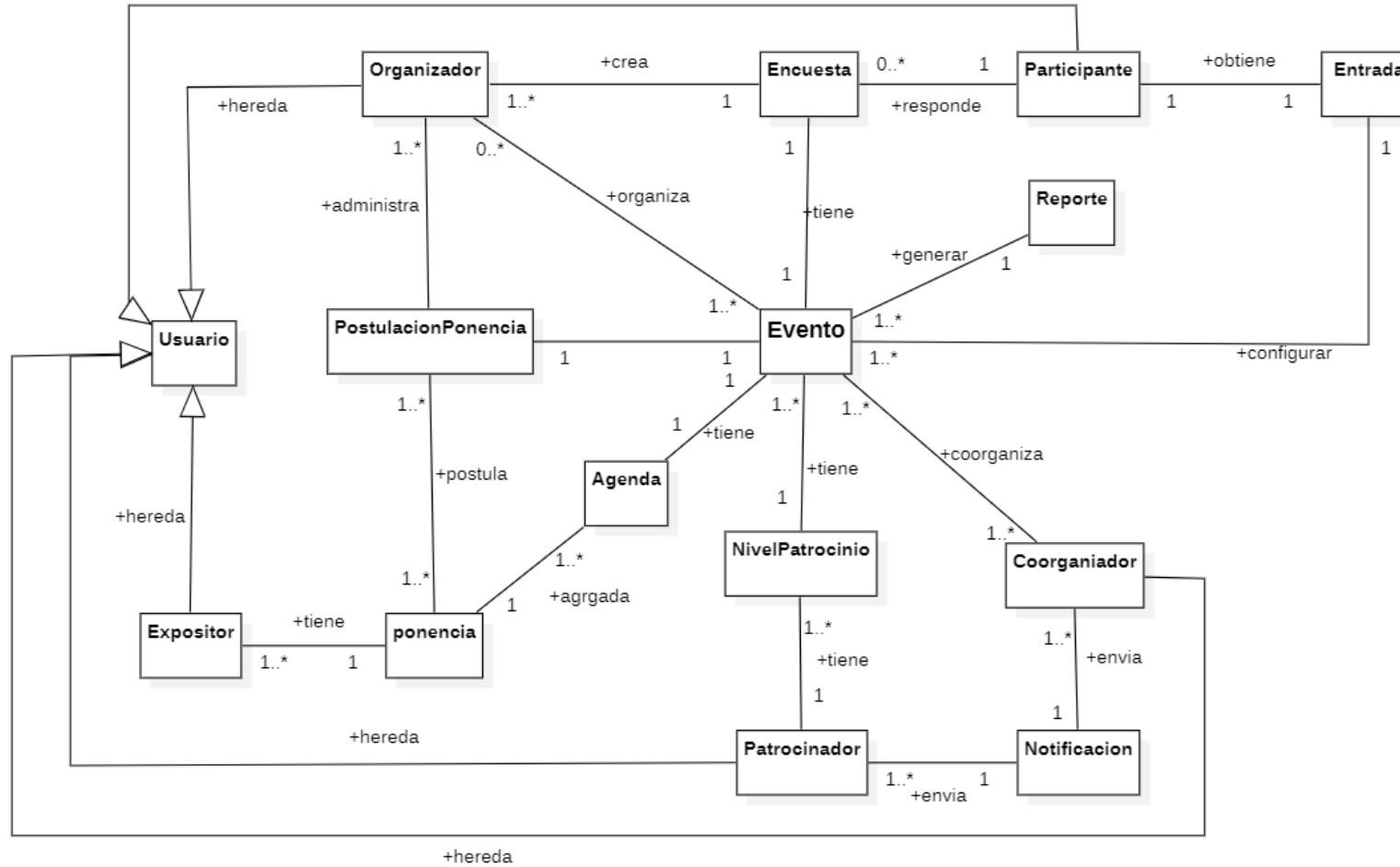


Figura 2-3. Modelo conceptual
Fuente: Elaboración Propia

2.8. Modelo de casos de uso

2.8.1. Caso de Uso: CU-01 Mantener Evento

Tabla 2.4. Caso de Uso Narrativo: CU-01 Mantener Evento

Caso de uso:	Mantener Evento
Código:	CU-01
Resumen:	Se crea un evento con los datos relevantes.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Organizador
Precondiciones:	El Actor se encuentra autenticado.
Flujo Normal	
Actor	Sistema
1.- El organizador selecciona crear evento.	2.- El sistema despliega un formulario con los siguientes campos: Nombre, Fecha y hora, imagen.
3.- El organizador introduce los datos.	4.- El sistema despliega formulario de tipo de evento: presencial, online e híbrido.
5.- El organizador selecciona presencial.	6.- El sistema despliega formulario de ubicación.
7.- El organizador ingresa los datos sobre la ubicación del evento.	8.- El sistema guarda los datos.
	9.- muestra mensaje de éxito.
Postcondición: El sistema creó exitosamente un Evento.	
Flujos Alternativos: 5.a selecciona online	
Actor	Sistema
5.a.1.- El organizador selecciona online.	5.a.2.- El sistema despliega formulario de la dirección online del evento.
5.a.3.- El organizador ingresa la dirección online.	5.a.4.- El sistema guarda los datos.
	5.a.5.- El sistema muestra mensaje de éxito.
Flujos Alternativos: 5.b selecciona híbrido	
Actor	Sistema
5.b.1.- El organizador selecciona híbrido.	5.b.2.- El sistema despliega formulario solicitando datos de ubicación y dirección online del evento.
5.b.3.- El organizador ingresa los datos de ubicación y dirección online.	5.b.4.- El sistema guarda los datos. 5.b.5.- El sistema muestra mensaje de éxito.

Fuente: Elaboración propia

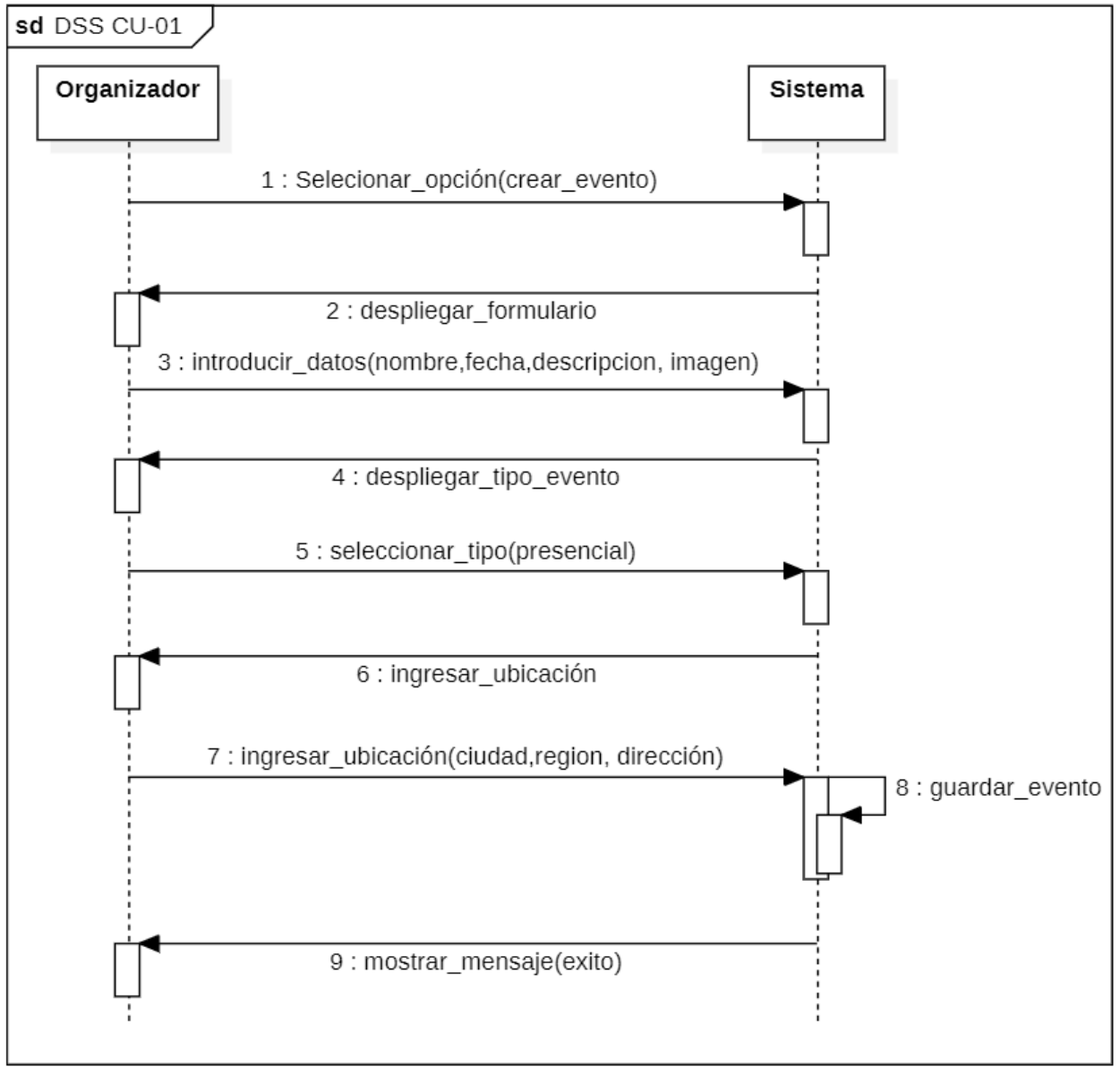


Figura 2-4. Diagrama Secuencia: CU-01 Gestionar Evento
Fuente: Elaboración Propia

Tabla 2.5. Contrato: CU-01 crear_evento

Nombre de la operación	crear_evento (nombre, fecha, lugar, tipo)
Referencia Cruzada	CU-01
Precondición	No debe de haber un evento con el mismo nombre. Que la fecha del evento solo puede ser futura.
Postcondición	Se creó un evento.

Fuente: Elaboración Propia

2.8.2. Caso de Uso: CU-02 Administrar entradas

Tabla 2.6. Caso de uso narrativo: CU-02 Administrar entradas

Caso de uso:	Administrar entradas
Código:	CU-05
Resumen:	Se crea las entradas que estarán disponible para el evento.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Organizador
Precondiciones:	- El actor se encuentra autenticado. - El evento ya ha sido creado y seleccionado.
Flujo Normal	
Actor	Sistema
1.- El organizador selecciona crear entradas.	2.- El sistema despliega formulario de creación de entradas.
3.- El organizador ingresa la cantidad de entrada disponible.	4.- El sistema guarda los datos. 5.- muestra mensaje de éxito.
Postcondición:	Se crearon entradas.

Fuente de elaboración propia

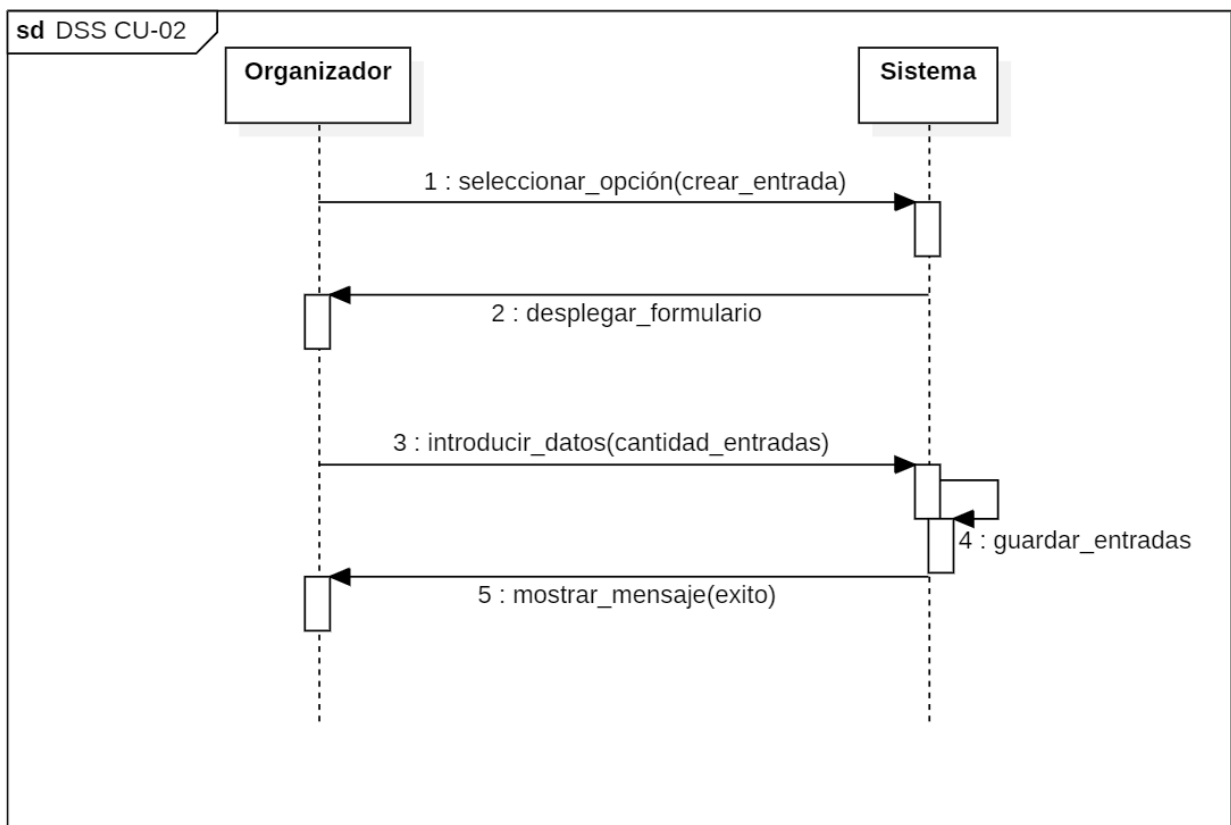


Figura 2-5. Diagrama Secuencia: CU-02 Administrar entradas

Fuente: Elaboración Propia

Tabla 2.7. Contratos: CU-02 guardar_entradas

Nombre de la operación	guardar_entradas(cantidad)
Referencia Cruzada	CU-02
Precondición	Debe de existir un evento activo.
Postcondición	Se guardó la cantidad de entradas.

Fuente: Elaboración Propia

2.8.3. Caso de Uso: CU-03 Enviar Notificación

Tabla 2.8. Caso de uso narrativo: CU-03 Enviar Notificación

Caso de uso:	Enviar notificación	
Código:	CU-03	
Resumen:	Los actores envían notificación a los participantes del evento.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador, Coorganizador, Patrocinador, Tiempo	
Precondiciones:	<ul style="list-style-type: none"> - El actor se encuentra autenticado. - El evento esta seleccionado. 	
Flujo Normal		
Actor	Sistema	
1.-El actor selecciona enviar notificación.		
	2.- El sistema comprueba que el actor tiene mensajes disponibles para enviar.	
	3.- El sistema muestra el número de mensajes que le quedan disponible y despliega formulario de mensaje.	
4.- El actor redacta el mensaje a selecciona enviar.		
	5.- El sistema despliega formulario de fecha y hora o enviar ahora.	
6.- El actor selecciona enviar ahora.		
	7.- El sistema envía el mensaje a los participantes que están inscritos en el evento.	
	8.- Se resta uno a la cantidad de mensaje disponible.	
	9.- El sistema despliega un mensaje de operación exitosa.	
Postcondición:	Se envió un mensaje a todos los participantes del evento y se resta una notificación al actor.	
Flujos Alternativos: El actor no tiene para enviar mensajes		
Actor	Sistema	
	3.a.1.- El sistema comprueba y despliega que no le quedan mensajes para enviar.	
	3.a.2.- Fin de caso de uso.	

Fuente de elaboración propia

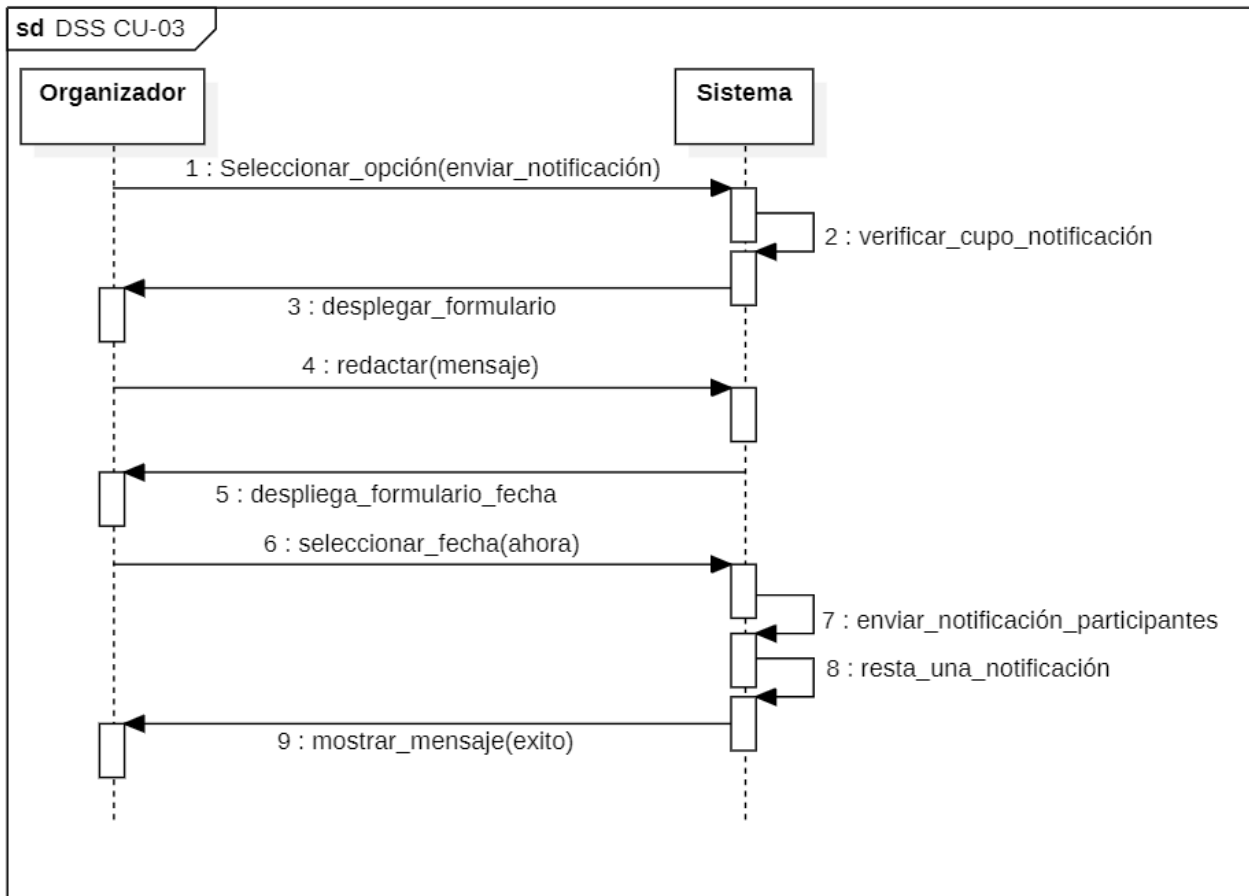


Figura 2-6. Diagrama Secuencia: CU-03 Enviar Notificación
Fuente: Elaboración Propia

Tabla 2.9. Contratos: CU-03.1 enviar_notificación

Nombre de la operación	enviar_notificación(texto)
Referencia Cruzada	CU-03
Precondición	El usuario debe estar autenticado, tener el rol de Organizador y debe existir un evento con participantes registrado
Postcondición	Se envió el mensaje

Fuente: Elaboración Propia

Tabla 2.10. Contrato: CU-03.2 restar_notificación

Nombre de la operación	restar_notificacion()
Referencia Cruzada	CU-03
Precondición	El mensaje ha sido enviado correctamente a los participantes. El actor tiene al menos 1 notificación disponible para enviar
Postcondición	Se restó en uno la cantidad de notificaciones disponibles El sistema actualizó el saldo de notificaciones disponibles.

2.8.4. Caso de Uso: CU-04 Obtener reporte de asistencia y encuesta

Tabla 2.11. Caso de uso narrativo: Obtener reporte de asistencia y encuesta

Caso de uso:	Obtener reporte asistencia y encuesta	
Código:	CU-04	
Resumen:	Los actores obtienen un reporte de la asistencia del evento y los resultados de la encuesta respondida por los participantes.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador, Coorganizador	
Precondiciones:	<ul style="list-style-type: none"> - El evento a terminado. - Se registraron los participantes que asistieron al evento. - Se ha terminado el periodo de contestación de encuesta. - El evento ha sido seleccionado. 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona la opción de reporte.		
	2.- El sistema despliega formulario de selección.	
3.- Selecciona la opción "Reporte de asistencia y encuesta".		
	4.- Solicita parámetros de la selección (evento, fecha, tipo, formato, etc.).	
5.- Selecciona los parámetros requeridos.		
	6. Genera el reporte combinando datos de asistencia y encuestas.	
	7.- Da un mensaje de éxito y el botón de descarga.	
Postcondición:	El actor obtuvo el reporte generado con los datos solicitados.	
Flujo Alternativo: : Parámetros inválidos o sin datos		
5.a El actor ingresa parámetros inválidos o incompletos.		
	6.a El sistema muestra un mensaje de error indicando que los parámetros son inválidos o faltantes.	
7.a El actor corrige los parámetros.		
	8.a.- El sistema vuelve al flujo normal en el paso 6.	

Fuente: Elaboración propia

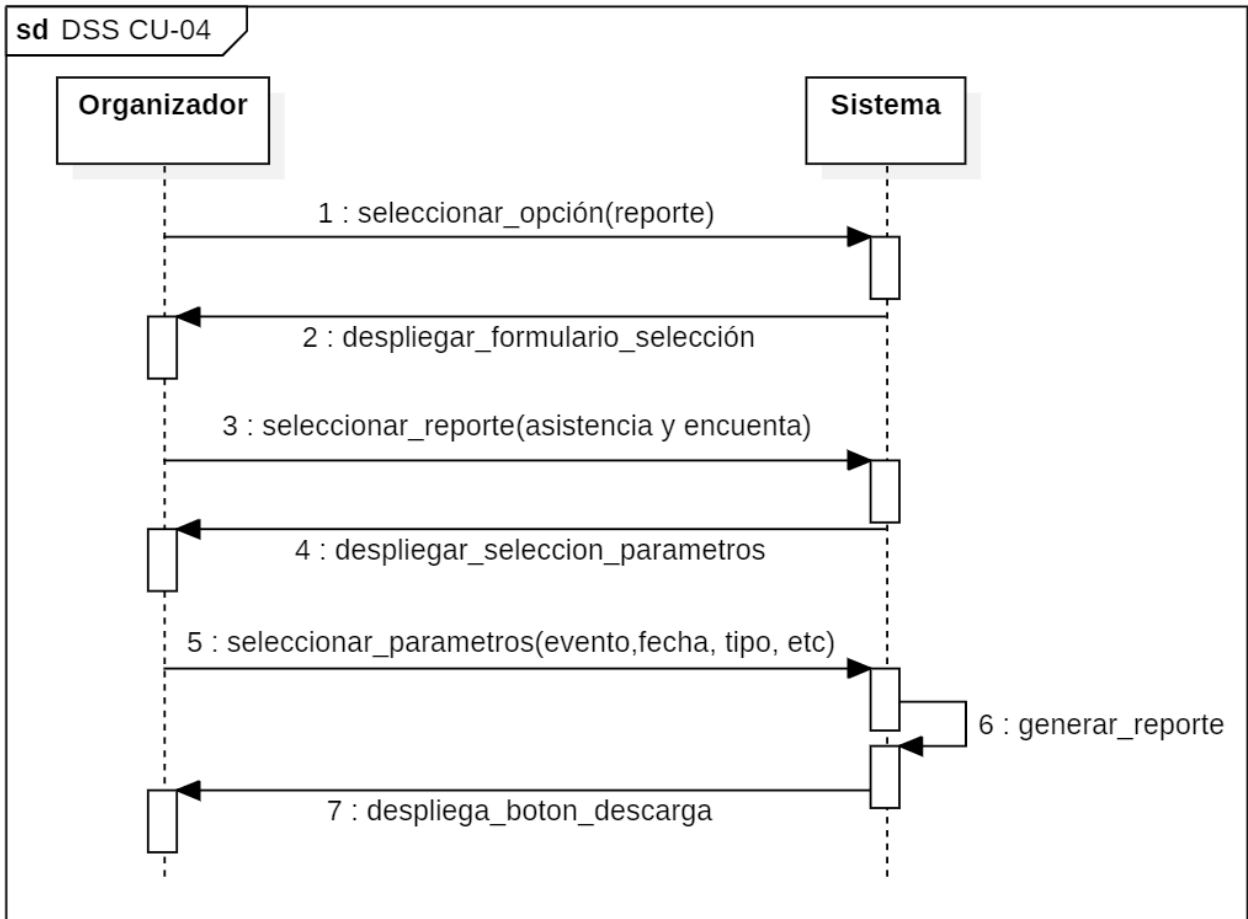


Figura 2-7. Diagrama Secuencia: CU-04 Obtener reporte de asistencia y encuesta
Fuente: Elaboración Propia

Tabla 2.12. Contrato: CU-04.1 generar_reporte

Nombre de la operación	generar_reporte
Referencia Cruzada	CU-04
Precondición	Debe existir al menos un evento finalizado con datos disponibles para el reporte.
Postcondición	Se generó un reporte con la información del evento y quedó disponible para su visualización o descarga.

Fuente: Elaboración propia

Tabla 2.13. Contrato: CU-04.2 desplegar_parametros

Nombre de la operación	desplegar_parametros
Referencia Cruzada	CU-04
Precondición	El formulario de reporte está visible.
Postcondición	El sistema recibió y registró los parámetros para generar el reporte.

Fuente: Elaboración propia

2.8.5. Caso de Uso: CU-05 Mantener encuesta

Tabla 2.14. Caso de uso narrativo: CU-05 Mantener encuesta

Caso de uso:	Crear Encuesta	
Código:	CU-05	
Resumen:	El actor diseña y crea una encuesta de satisfacción para que los asistentes puedan evaluarla el evento terminado.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador	
Precondiciones:	<ul style="list-style-type: none"> - El actor se encuentra autenticado. - El evento esta seleccionado. 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona la opción de crear encuesta.		
	2.- El sistema despliega el formulario para la configuración y preguntas.	
3.- El actor ingresa título, descripción y agrega preguntas.		
	4.- El sistema valida los datos ingresados.	
5.- Finaliza la encuesta.		
	6.- El sistema guarda la encuesta.	
	7.- Muestra un mensaje de éxito.	
Postcondición:	Se creó una encuesta exitosamente	
Flujos Alternativos: validación		
Actor	Sistema	
	4.a.1.- El sistema valida y muestra errores detectados, destacando los campos a corregir.	
4.a.2.- El actor corrige los errores		
	4.a.3.- el sistema revalúa y si están bien continúan con el flujo normal	

Fuente: Elaboración propia

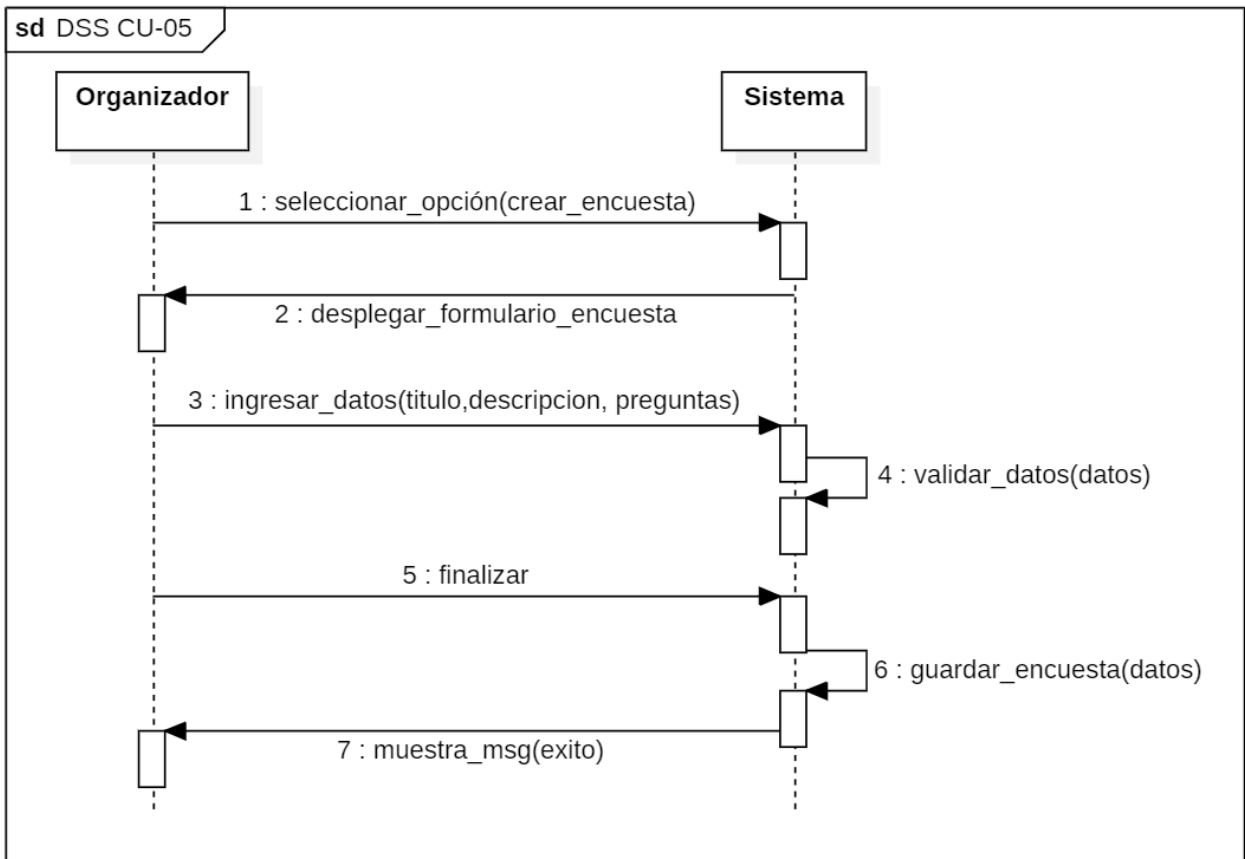


Figura 2-8. Diagrama Secuencia: CU-05 Mantener encuesta
Fuente: Elaboración Propia

Tabla 2.15. Contrato: CU-05.1 crear_encuesta

Nombre de la operación	crear_encuesta(titulo, descripción, preguntas)
Referencia Cruzada	CU-05
Precondición	Debe existir un evento activo al que asociar la encuesta.
Postcondición	Se creó una encuesta asociada al evento con las preguntas especificadas.

Fuente: Elaboración propia

Tabla 2.16. Contrato: CU-05.2 validar_datos

Nombre de la operación	validar_datos(datos)
Referencia Cruzada	CU-05
Precondición	El actor ingresó título, descripción y preguntas.
Postcondición	Se verificó que el título, descripción y preguntas fueran válidos. Se preparó la encuesta para ser guardada si no hubo errores

Fuente: Elaboración propia

2.8.6. Caso de Uso: CU-06 Administrar postulación de ponencias

Tabla 2.17. Caso de uso narrativo: Administrar postulación de ponencias

Caso de uso:	Administrar postulación de ponencias	
Código:	CU-06	
Resumen:	Se administran las postulaciones, aceptando y rechazando postulaciones.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador	
Precondiciones:	<ul style="list-style-type: none"> - Existen ya postulaciones de eventos. - El actor se encuentra autenticado. 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona administrar postulaciones.		
	2.- El sistema despliega interfaz crear postulaciones, listar postulaciones etc.	
3.- El actor selecciona listar postulaciones.		
	4.- El sistema despliega las ponencias postuladas.	
5.- El actor selecciona una postulación para revisar.		
	6.- El sistema despliega los detalles de la postulación, título, descripción etc. en conjunto con las opciones de aceptar y rechazar.	
7.-El actor seleccionar aceptar o rechazar.		
	8.- El sistema actualiza el estado de la postulación.	
	9.- El sistema formulario de confirmación.	
10.- El actor confirma la acción.		
	11.- El sistema guarda la información.	
	12.- el sistema muestra un mensaje de éxito.	
Postcondición:	La ponencia quedó como aceptada o rechazada.	

Fuente: Elaboración propia

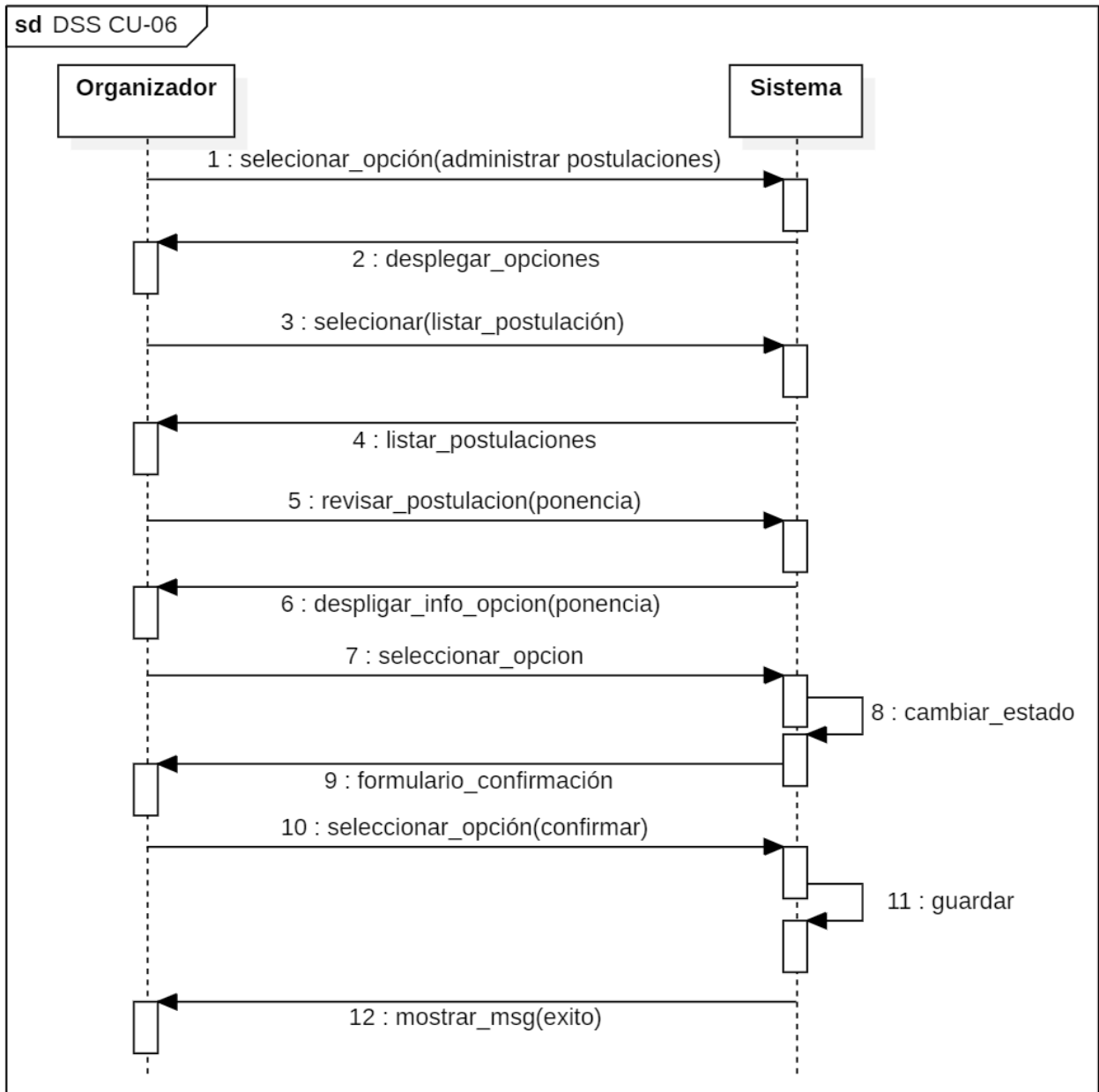


Figura 2-9. Diagrama Secuencia: CU-06 Administrar Postulación
Fuente: Elaboración Propia

Tabla 2.18. Contrato: CU-06 administrar_postulaciones

Nombre de la operación	administrar_postulaciones(postulacion_id, acción)
Referencia Cruzada	CU-06
Precondición	Debe existir una postulación pendiente asociada a un evento válido.
Postcondición	La postulación se aceptó o rechazó según la acción indicada, y su estado se actualizó en el sistema.

Fuente: Elaboración propia

2.8.7. Caso de Uso: CU-07 Asignar rol

Tabla 2.19. Caso de uso narrativo: CU-07 Asignar rol

Caso de uso:	Asignar rol
Código:	CU-07
Resumen:	El organizador asigna roles específicos a usuarios registrados para colaborar en la organización y operación del evento.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Organizador
Precondiciones:	- El actor debe de estar autenticado. - El evento esta seleccionado.
Flujo Normal	
Actor	Sistema
1.- El actor seleccionar asignar rol.	2.- El sistema despliega formulario de búsqueda.
3.- El actor ingresa datos del usuario.	4.- El sistema muestra una lista de los usuarios.
5.- El actor selecciona un usuario y le asigna un rol.	6.- El sistema guarda el nuevo estado.
	7.- El sistema muestra mensaje de éxito.
Postcondición:	Se asignó un rol a un usuario

Fuente: Elaboración propia

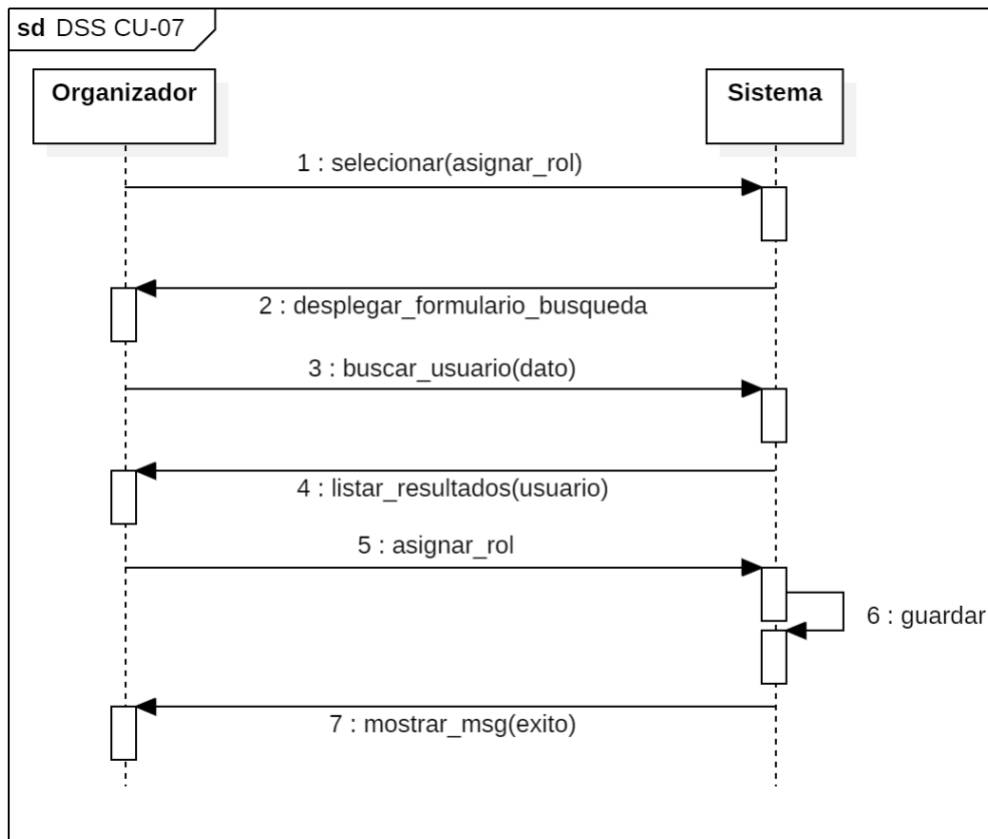


Figura 2-10. Diagrama Secuencia: CU-07 Asignar Rol

Fuente: Elaboración Propia

Tabla 2.20. Contrato: CU-07 asignar_rol

Nombre de la operación	asignar_rol(usuario_id, rol)
Referencia Cruzada	CU-07
Precondición	El usuario destino debe existir en el sistema.
Postcondición	Se asignó correctamente el rol al usuario especificado

Fuente: Elaboración propia

2.8.8. Caso de Uso: CU-08 Administrar agenda

Tabla 2.21. Caso de uso narrativo CU-08 Administrar agenda

Caso de uso:	Administrar agenda
Código:	CU-08
Resumen:	Se crea una agenda de presentaciones con los datos relevantes
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Organizador
Precondiciones:	- El Actor se encuentra autenticado. - Evento ya ha sido creado y seleccionado.
Flujo Normal	
Actor	Sistema
1.- El organizador selecciona crear agenda.	2.- El sistema despliega formulario que solicitara los siguientes datos, cantidad de secciones, franja horaria.
3.-El organizador ingresa los datos solicitados.	4.- El sistema guarda los datos.
	5.- Fin de caso de uso.
Postcondición:	Se creó la agenda del evento

Fuente de elaboración propia

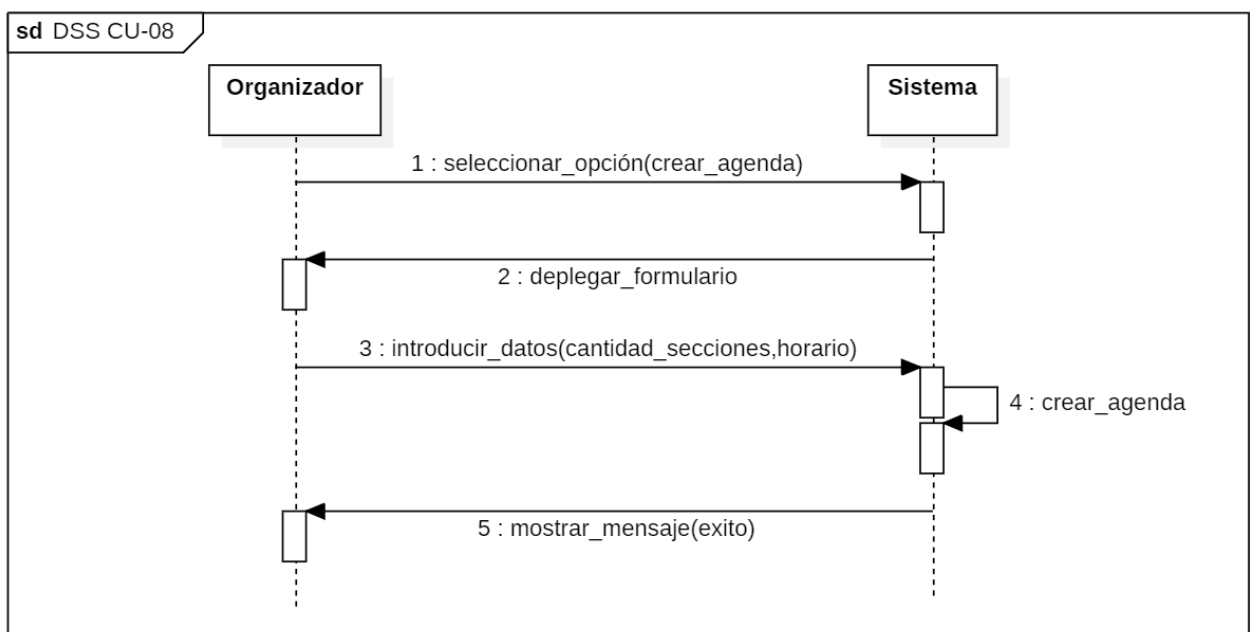


Figura 2-11. Diagrama Secuencia: CU-08 Administrar agenda

Fuente: Elaboración Propia

Tabla 2.22. Contratos: CO-08 crear_agenda

Nombre de la operación	crear_agenda(cantidad_seccion, horario)
Referencia Cruzada	CU-08
Precondición	Debe existir un evento activo sin una agenda previamente creada.
Postcondición	Se creó una agenda asociada al evento con los bloques horarios definidos.

Fuente: Elaboración Propia

2.8.9. Caso de Uso: CU-09 Mantener nivel de patrocinio

Tabla 2.23. Caso de uso narrativo: CU-09 Mantener nivel de patrocinio

Caso de uso:	Gestionar nivel de patrocinio
Código:	CU-03
Resumen:	Crear el nivel de patrocinio
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Organizador
Precondiciones:	- El Actor se encuentra autenticado. - El evento ya ha sido creado y seleccionado.
Actor	Sistema
1.- El organizador selecciona crear nivel patrocinio	2.- El sistema despliega formulario de creación de patrocinio con los siguientes datos: titulo, descripción, cantidad de mensajes que pueden enviar.
3.- El organizador ingresa los datos de título, descripción y la cantidad mensaje a enviar.	4.- El sistema guarda los datos.
	5.- El sistema muestra un mensaje de éxito.
Postcondición:	Se creó el nivel de patrocinio.

Fuente de elaboración propia

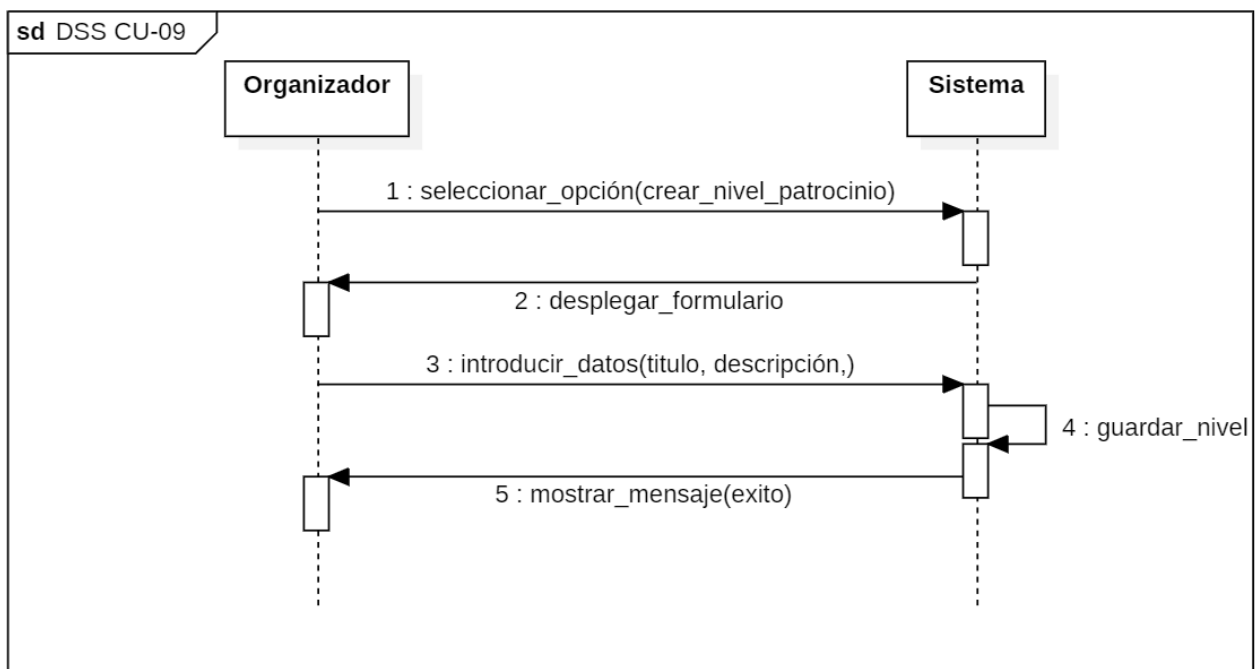


Figura 2-12. Diagrama Secuencia: CU-09 Gestionar categoría de patrocinio

Fuente: Elaboración Propia

Tabla 2.24. Contratos: CO-09 Gestionar categoría de patrocinio

Nombre de la operación	crear_nivel_patrocinio(titulo, descripción)
Referencia Cruzada	CU-09
Precondición	Debe existir un evento activo al que asociar el nivel de patrocinio.
Postcondición	Se creó un nuevo nivel de patrocinio con sus beneficios y monto, asociado al evento.

Fuente: Elaboración Propia

2.8.10. Caso de Uso: CU-10 Asignar cantidad de notificaciones

Tabla 2.25 Caso de uso narrativo CU-10 Asignar cantidad de notificaciones

Caso de uso:	Asignar cantidad de notificaciones	
Código:	CU-10	
Resumen:	El organizador asigna cantidad de notificaciones a otros usuarios para enviar información relevante.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador	
Precondiciones:	<ul style="list-style-type: none"> - EL actor se encuentra autenticado. - Los usuarios tienen un rol de patrocinador, coorganizador, patrocinador - El evento ya ha sido creado y seleccionado. 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona asignar notificaciones.		
	2. El sistema despliega una lista de los usuarios que tengan el rol de coorganizador y patrocinador.	
3.- El actor selecciona un usuario.		
	4.- El sistema muestra la opción de asignar notificaciones.	
5.- El actor asigna notificaciones.		
	6.- El sistema guarda el cambio.	
	7.- Muestra un mensaje de éxito.	
Postcondición:	Se asignó una cantidad de notificaciones para poder enviar.	
Flujo Alternativo: Sin usuarios elegibles		
	2.a El sistema no encontró usuarios con los roles de organizador, coorganizador o expositor.	
	3.a El sistema muestra mensaje indicando que no hay usuarios para seleccionar.	

Fuente: Elaboración propia

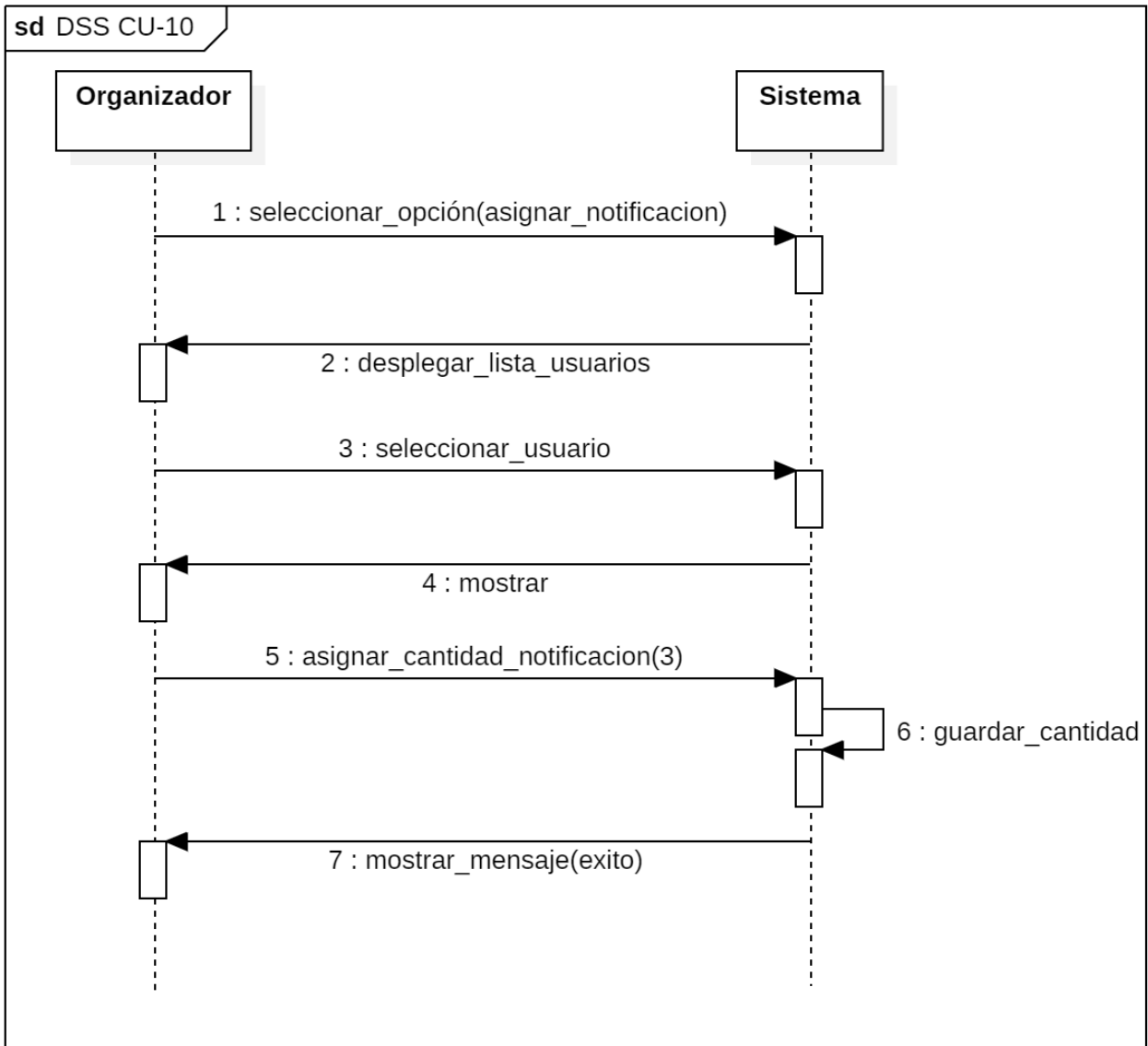


Figura 2-13. Diagrama Secuencia: CU-10 Asignar cantidad de notificación
Fuente: Elaboración Propia

Tabla 2.26. Contrato: CU-10 asignar_notificaciones

Nombre de la operación	asignar_notificaciones(usuario_id, cantidad)
Referencia Cruzada	CU-10
Precondición	Usuario destino debe existir en el sistema y debe de tener un rol de patrocinador o coorganizador
Postcondición	Se asignó correctamente la cantidad de notificaciones disponibles al usuario especificado.

Fuente: Elaboración propia

2.8.11. Caso de Uso: CU-11 Mantener Perfil

Tabla 2.27. Caso de uso narrativo: CU-11 Mantener perfil

Caso de uso:	Mantener perfil
Código:	CU-11
Resumen:	Se mantienen en perfiles de los usuarios.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Patrocinador, coorganizador, expositor
Precondiciones:	- Los actores se encuentra autenticado. - El evento ya ha sido creado y seleccionado.
Actor	Sistema
1.- El actor selecciona crear perfil.	2.- El sistema despliega formulario de creación del perfil con los siguientes títulos, datos, imagen descripción.
3.- El actor ingresa los datos de imagen logotipo y descripción.	4.- El sistema guarda los datos.
	5.-muestra mensaje de éxito.
Post Condición:	Se creó exitosamente el perfil del actor.

Fuente de elaboración propia

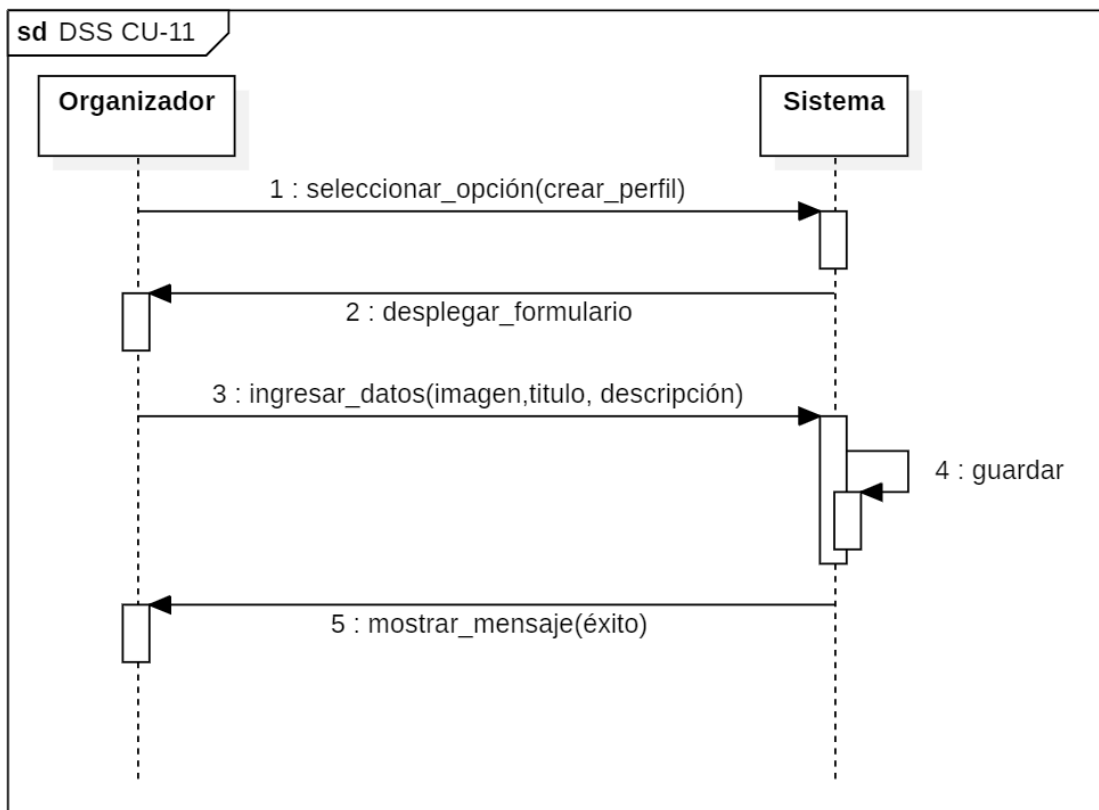


Figura 2-14. Diagrama Secuencia: CU-11 Mantener Perfil

Fuente: Elaboración Propia

Tabla 2.28. Contratos: CU-11 crear_perfil

Nombre de la operación	crear_perfil(titulos, datos_personales, imagen, descripción)
Referencia Cruzada	CU-11
Precondición	El usuario debe estar autenticado y no debe tener un perfil previamente creado en el sistema
Postcondición	Se creó un perfil

Fuente: Elaboración Propia

2.8.12. Caso de Uso: CU-12 Monitorear inscripciones al evento

Tabla 2.29. Caso de uso narrativo: CU-12 Monitorear inscripciones al evento

Caso de uso:	Monitorear inscripciones al evento	
Código:	CU-12	
Resumen:	El organizador supervisa en tiempo real o de forma periódica la cantidad y estado de las inscripciones al evento, permitiéndole tomar decisiones de comunicación, logística o capacidad.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Organizador, Coorganizador	
Precondiciones:	<ul style="list-style-type: none"> - El evento debe de estar seleccionado. - Las inscripciones al evento deben de estar activa. 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona la opción de monitorear inscripciones.		
	2.- El sistema despliega opciones para visualizar, inscripciones por día, total, disponibilidad etc.	
3.- El actor selecciona los ítems que le interesan.		
	4.- El sistema procesa la información.	
	5.- El sistema muestra en pantalla la información.	
	6.- El sistema da la opción de descargar la información en el formato a elegir.	
7.- El actor selecciona el formato.		
	8.- El sistema genera el informe.	
	9.- muestra el mensaje de éxito.	
Postcondición:	Se generó un reporte visual.	

Fuente: Elaboración propia

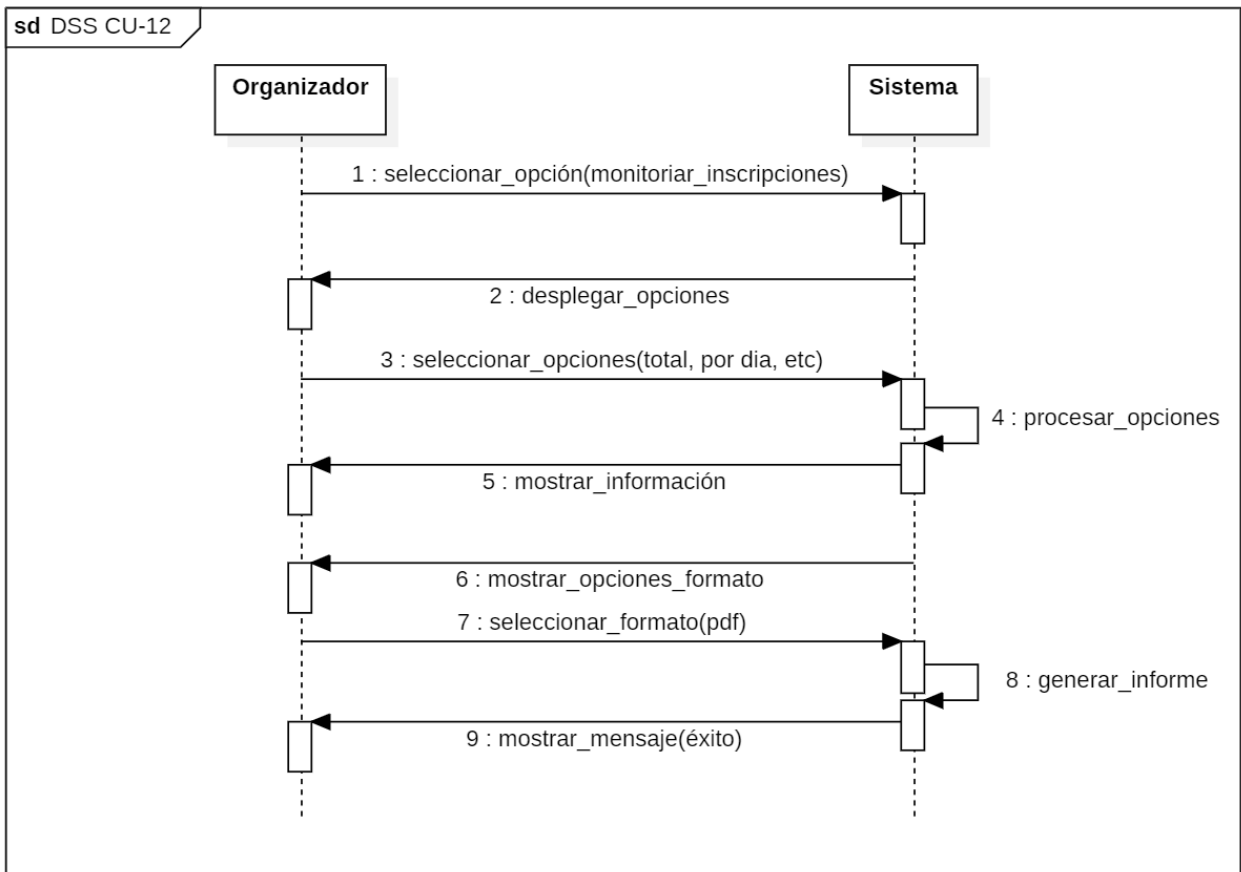


Figura 2-15. Diagrama Secuencia: CU-12 Monitorar inscripciones al evento
Fuente: Elaboración Propia

Tabla 2.30. Contrato: CU-12 monitorear_incripciones

Nombre de la operación	monitorear_incripciones(items)
Referencia Cruzada	CU-12
Precondición	Debe existir al menos un evento activo con inscripciones registradas.
Postcondición	Se obtuvo la información solicitada sobre las inscripciones del evento, disponible para visualización o análisis.

Fuente: Elaboración propia

Tabla 2.31. Contrato: CU-12.2 procesar_opciones

Nombre de la operación	procesar_opciones()
Referencia Cruzada	CU-12
Precondición	Las opciones están definidas y son válidas.
Postcondición	Se filtró y procesó la información correspondiente según los criterios.

Fuente: Elaboración propia

2.8.13. Caso de Uso: CU-13 Obtener Entrada

Tabla 2.32. Caso de uso narrativo: CU-13 Obtener Entrada

Caso de uso:	Obtener entrada	
Código:	CU-13	
Resumen:	EL participante obtiene una entrada para ingresar al evento.	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Usuario	
Precondiciones:	<ul style="list-style-type: none"> - El actor se encuentra autenticado. - Tiene seleccionado el evento 	
Flujo Normal		
Actor	Sistema	
1.- El actor selecciona obtener entrada.		
	2.- El sistema despliega formulario de tipo de entrada, presencial u online.	
3.- El actor selecciona la entrada presencial.		
	4.- El sistema revisa si quedan entradas disponibles.	
	5.- El sistema asigna una entrada con su respectivo número.	
	6.- Muestra un mensaje de éxito.	
Post Condición:	El usuario obtuvo su entrada para el evento	
Flujos Alternativos: Selecciona online		
Actor	Sistema	
3.a- El participante selecciona la entrada presencial.	4.a El sistema revisa si quedan entradas disponibles.	
	5.a el sistema asigna una entrada con su respectivo número.	
	6.a Muestra un mensaje de éxito.	

Fuente de elaboración propia

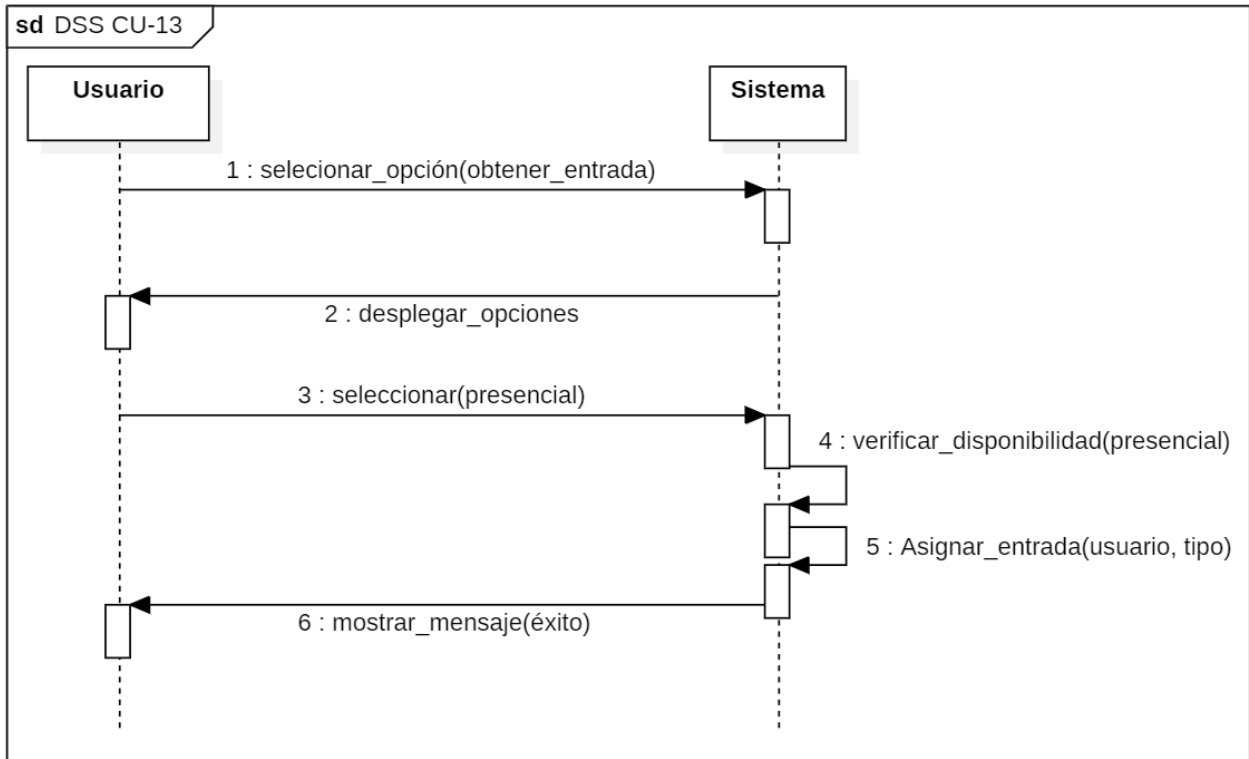


Figura 2-16. Diagrama Secuencia: CU-13 Obtener Entrada

Fuente: Elaboración Propia

Tabla 2.33. Contratos: CO-13 asignar_entrada

Nombre de la operación	asignar_entrada(evento_id)
Referencia Cruzada	CU-10
Precondición	El evento debe existir y tener entradas disponibles
Postcondición	Se asignó una entrada al usuario para el evento especificado y se actualizó el stock de entradas disponibles

Fuente: Elaboración Propia

2.8.14. Caso de Uso: CU-14 Cancelar entrada

Tabla 2.34. Caso de uso narrativo: CU-14 Cancelar entrada

Caso de uso:	Cancelar entrada
Código:	CU-11
Resumen:	El participante cancela la entrada al evento
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Participante
Precondiciones:	- El actor se encuentra autenticado. - Tiene seleccionado el evento
Actor	Sistema
1.- El actor selecciona la opción de cancelar entrada.	
	2.- El sistema confirma si tiene una entrada en el evento.
	3.- El sistema despliega confirmación de cancelar la entrada.
4.- El participante confirma la cancelación de entrada ingresando su contraseña.	
	5.- El sistema valida la contraseña.
	6.- El sistema cancela la entrada.
	7.- El sistema despliega confirmación de la cancelación.
Postcondición:	Se liberó la entrada al evento y el participante quedó sin una entrada asociada.

Fuente de elaboración propia

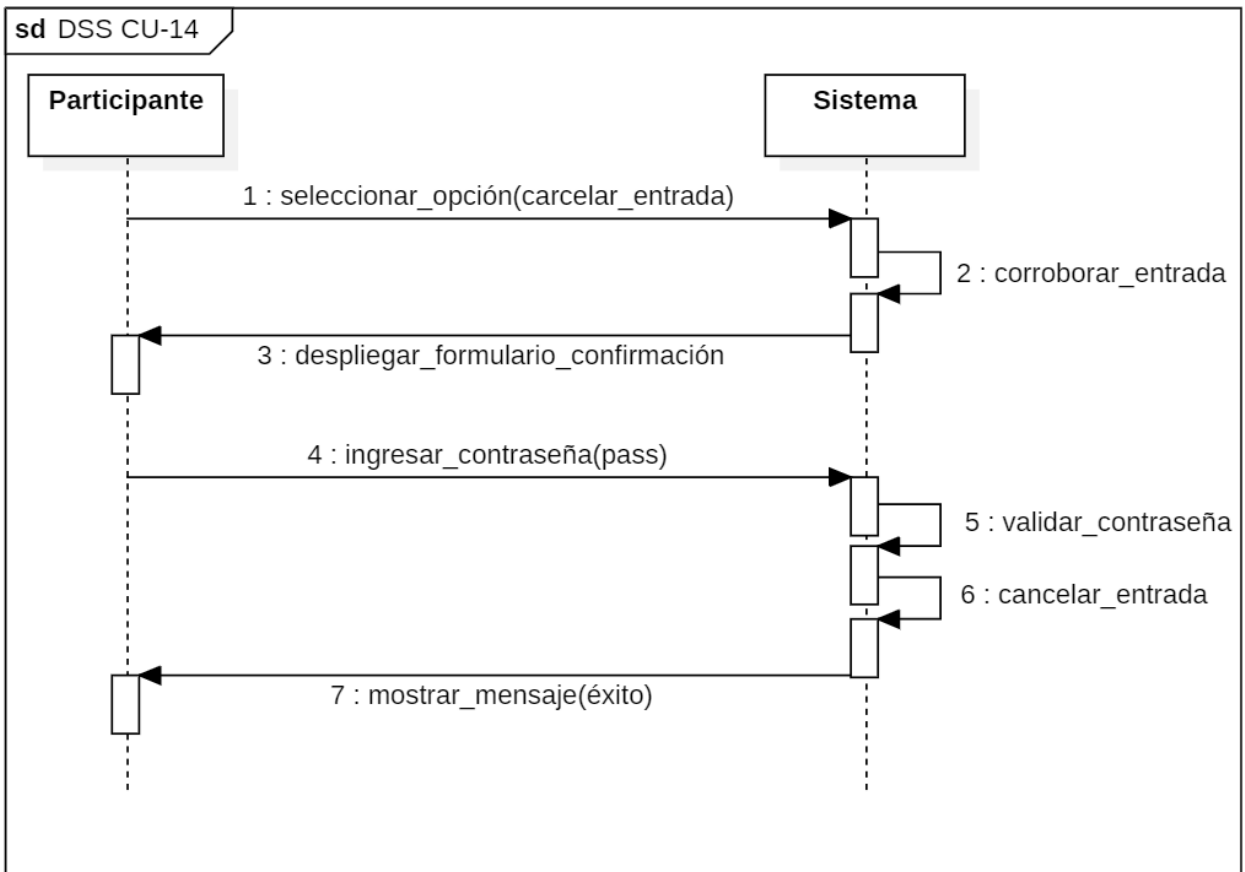


Figura 2-17. Diagrama Secuencia CU-14 Cancelar Entrada
Fuente: Elaboración Propia

Tabla 2.35. Contratos: CO-14 cancelar_entrada

Nombre de la operación	cancelar_entrada(entrada_id)
Referencia Cruzada	CU-14
Precondición	Tener una entrada a un evento
Postcondición	Se canceló la entrada al evento

Fuente: Elaboración Propia

2.8.15. Caso de Uso: CU-15 Mantener cuenta

Tabla 2.36 Caso de uso narrativo: CU-15 Mantener cuenta

Caso de uso:	Mantener cuenta	
Código:	CU-15	
Resumen:	Los usuarios administran su información personal	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Coorganizador, expositor, participante, usuario, patrocinador	
Precondiciones:	El usuario debe de estar autenticado	
Flujo Normal		
Actor	Sistema	
1.- El actor seleccionar "mi cuenta".		
	2.- El sistema muestra los datos actuales como, nombre, email etc.	
3.- El actor modifica los datos como nombre, email etc.		
	4.- El sistema valida los datos ingresados.	
	5.- El sistema despliega interfaz de confirmación.	
6.- El actor ingresa su contraseña como confirmación.		
	7.- El sistema valida la contraseña.	
	8.- El sistema guarda el cambio.	
	9.- El sistema muestra un mensaje de éxito.	
Postcondición:	Se actualizó los datos.	

Fuente: Elaboración propia

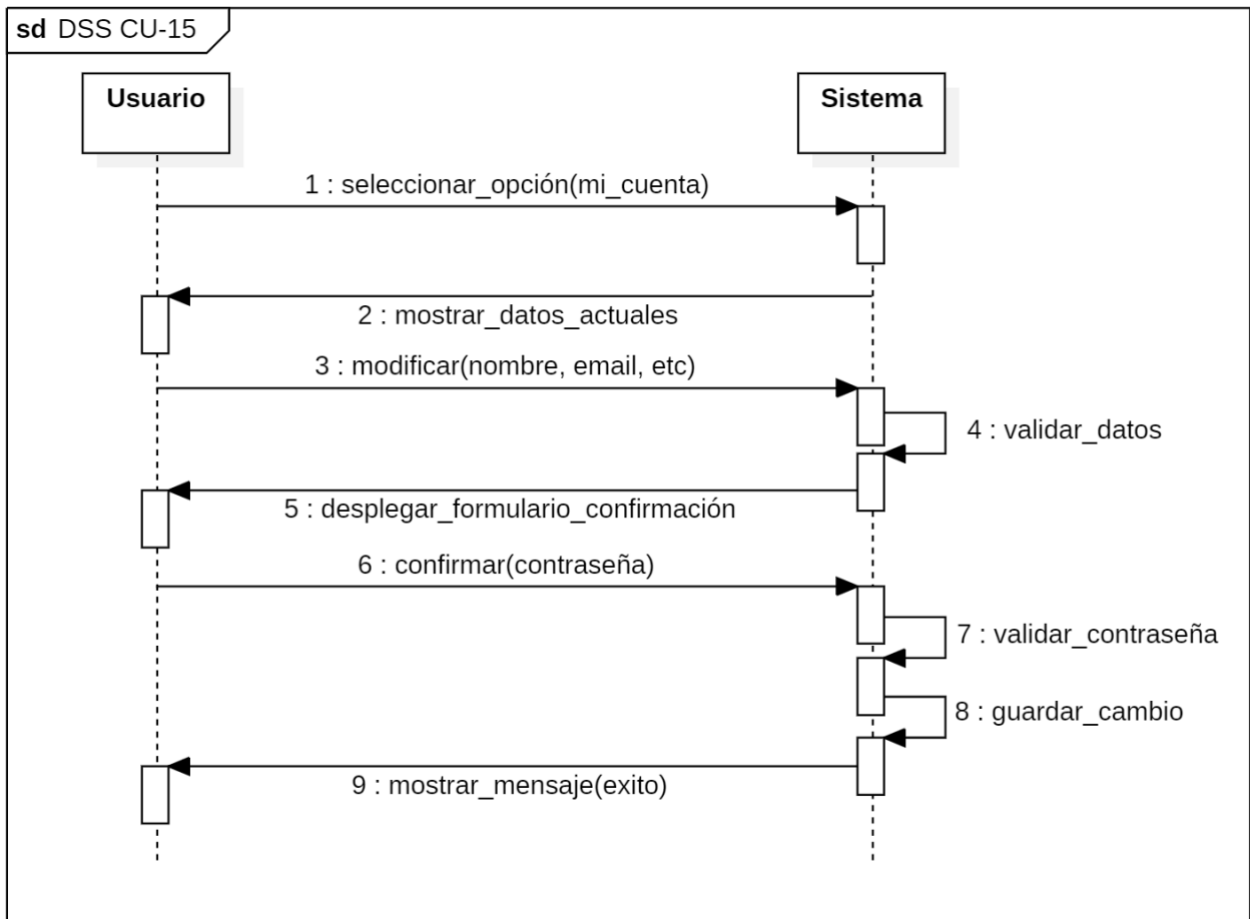


Figura 2-18. Diagrama Secuencia: CU-15 Mantener Cuenta
Fuente: Elaboración Propia

Tabla 2.37. Contrato: CU-15 modificar

Nombre de la operación	modificar(nombre, email)
Referencia Cruzada	CU-15
Precondición	El usuario debe estar autenticado
Postcondición	Se actualizó correctamente los datos de la cuenta en el sistema.

Fuente: Elaboración propia

2.8.16. Caso de Uso: CU-16 Registrar ingreso

Tabla 2.38. Caso de uso narrativo: CU-16 registrar ingreso

Caso de uso:	Comprobar entrada
Código:	CU-16
Resumen:	Se comprueba que la entrada del participante es válida y que corresponde a la persona que la presenta.
Tipo:	Evidente.
Referencias Cruzadas:	N/A
Actor:	Organizador y Coorganizador
Precondiciones:	- El actor se encuentra autenticado.
Actor	Sistema
1.- El coorganizador selecciona comprobar entrada.	
	2.- El sistema despliega formulario de solicitando datos como, numero de entrada, Rut, nombre y apellido.
3.- El organizador ingresa los datos del participante.	
	4.- El sistema comprueba la información.
	5.- El sistema guarda como presente.
	6.- Despliega información de éxito la entrada es válida.
Post Condición:	La entrada se validó y se permitió el ingreso.

Fuente: Elaboración propia

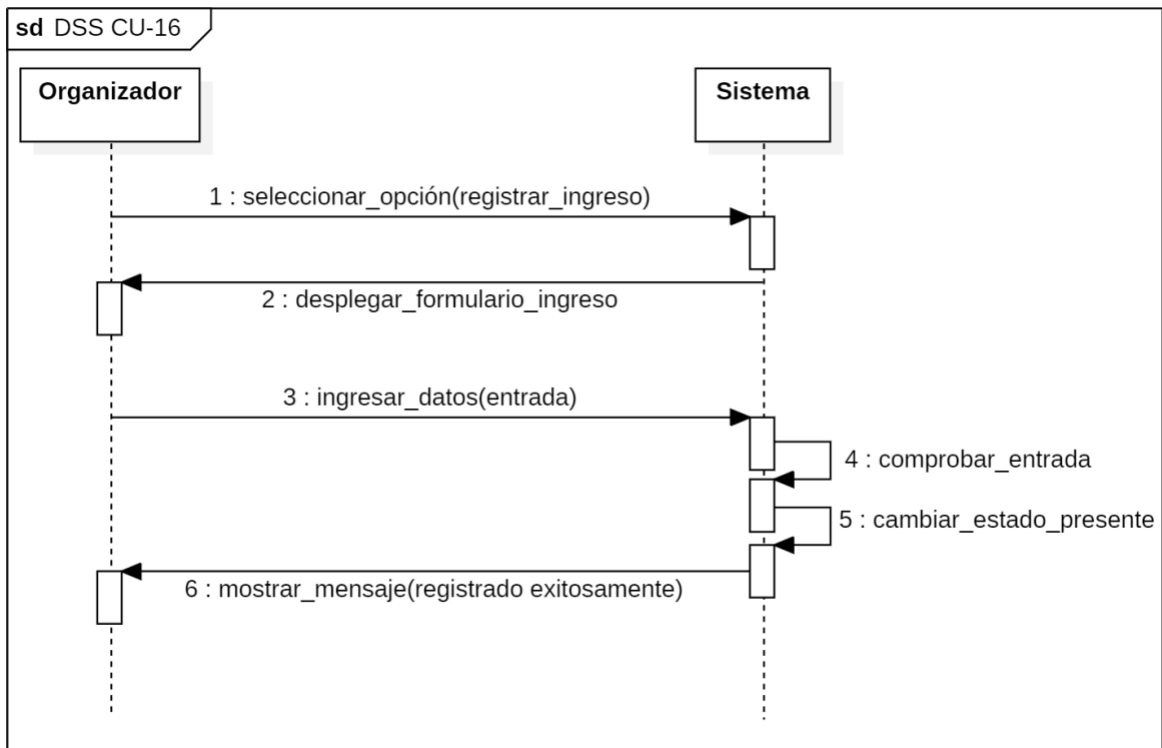


Figura 2-19. Diagrama Secuencia: CU-16 Registrar ingreso

Fuente: Elaboración Propia

Tabla 2.39. Contratos: CO-16 comprobar_entrada

Nombre de la operación	comprobar_entrada(entrada)
Referencia Cruzada	CU-16
Precondición	Debe existir un evento activo en proceso de validación. El Rut y la entrada deben corresponder a una entrada emitida.
Postcondición	Se verificó la validez de la entrada. Si es válida, se marca como utilizada, si no, se informa que es inválida.

Fuente: Elaboración Propia

2.8.17. Caso de Uso: CU-17 Responder encuesta

Tabla 2.40. Caso de uso narrativo: responder encuesta.

Caso de uso:	Responder encuesta	
Código:	CU-17	
Resumen:	Los participantes responden las preguntas de la encuesta	
Tipo:	Evidente	
Referencias Cruzadas:	N/A	
Actor:	Participante	
Precondiciones:	El actor debe de estar autenticado	
Flujo Normal		
Actor	Sistema	
1.- El actor seleccionar responder encuesta.		
	2.- El sistema verifica que el usuario haya asistido al evento.	
	3.- El sistema despliega la encuesta.	
4.- El actor responde las preguntas de la encuesta.		
	5.- El sistema valida que la preguntas fueron respondidas.	
	6.- Guardar respuestas.	
	7.- Mostrar mensaje de éxito.	
Postcondición:	Se guardó las respuestas y que el usuario ya ha respondido.	
Flujos Alternativos: El no asistió al evento		
Actor	Sistema	
	3.- El sistema despliega mensaje de no asistencia al evento	
	4.- El sistema despliega un formulario con la pregunta de porque no asistió al evento	
5.- El participante responde		
	6.- el sistema valida que ha respondido.	
	7.- el sistema guarda el mensaje.	
	8.- muestra un mensaje de éxito.	

Fuente: Elaboración propia

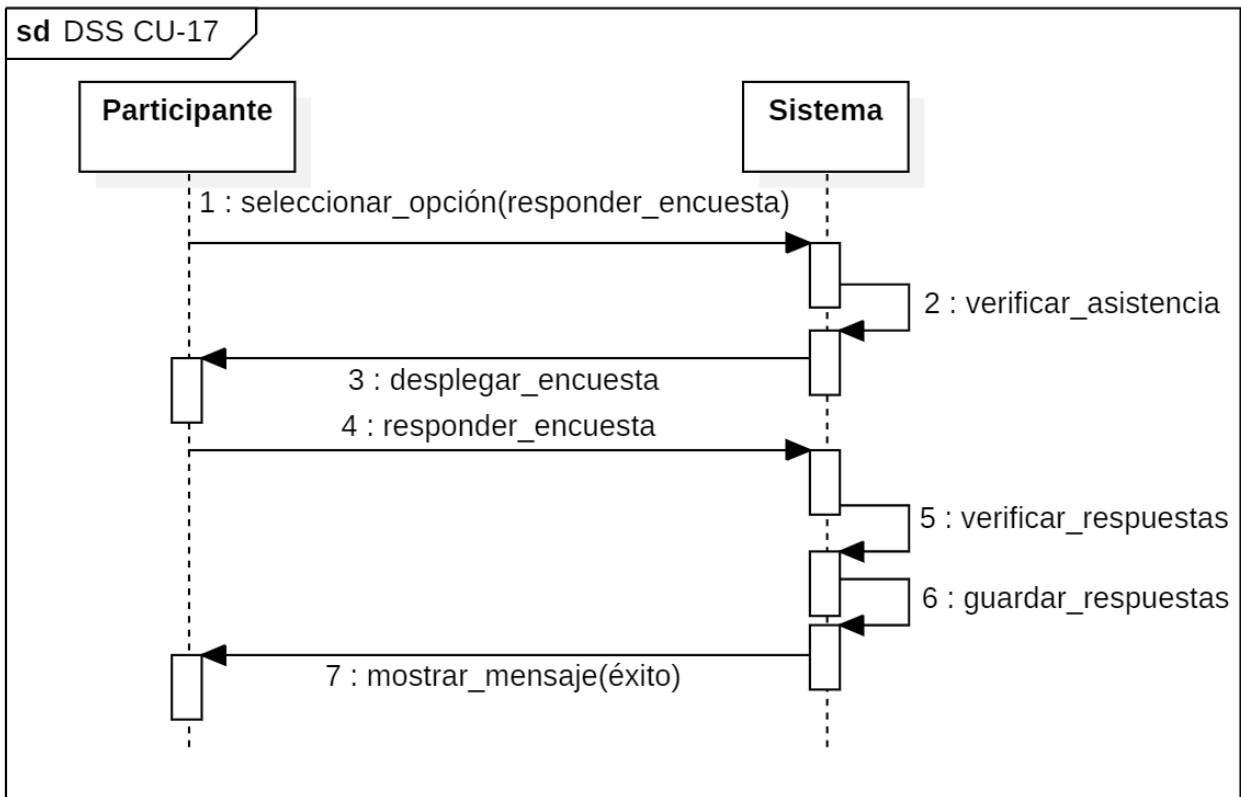


Figura 2-20. Diagrama Secuencia: CU-17 Responder encuesta
Fuente: Elaboración Propia

Tabla 2.41. Contrato: CU-17 validar_repuestas

Nombre de la operación	validar_repuestas (respuestas)
Referencia Cruzada	CU-17
Precondición	El actor completó el formulario de la encuesta.
Postcondición	Se verificó que todas las respuestas requeridas fueron completadas.

Fuente: Elaboración propia

Tabla 2.42. Contrato: CU-17 guardar_respuestas

Nombre de la operación	guardar_repuestas (encuesta_id,usuario,respuestas)
Referencia Cruzada	CU-17
Precondición	La encuesta debe estar activa y debe estar asociada a un evento al que el usuario haya asistido
Postcondición	Se registró las respuestas del usuario de la encuesta correspondiente.

Fuente: Elaboración propia

2.8.18. Caso de Uso: CU-18 Administrar ponencia

Tabla 2.43. Caso de uso narrativo: CU-18 Administrar postulación

Caso de uso:	Administrar ponencia
Código:	CU-18
Resumen:	Crear y enviar ponencia.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Usuario, participante, expositor
Precondiciones:	- El actor se encuentra autenticado. - El actor tiene seleccionado el evento
Actor	Sistema
1.- El actor selecciona crear ponencia.	
	2.- El sistema despliega formulario de creación de ponencia.
3.- El actor ingresa los datos de la ponencia, titulo, resumen, etc.	
	4.- El sistema verifica que los datos estén.
	5.- El sistema despliega opción de guarda y guardar y enviar.
6.-El actor selecciona guardar y enviar.	
	7.- El sistema guarda y envía postulación.
	8.- El sistema muestra un mensaje de éxito.
Post Condición:	Se envió la presentación para la evaluación.

Fuente de elaboración propia

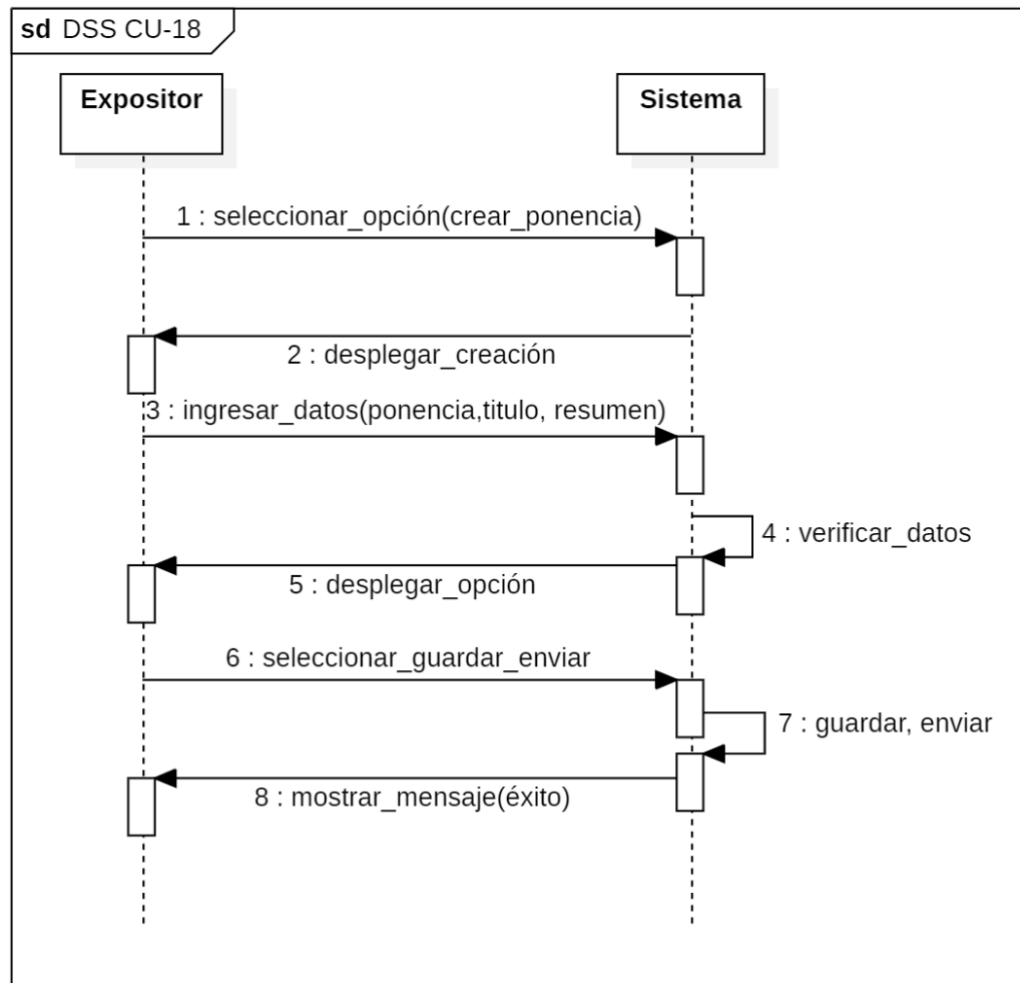


Figura 2-21. Diagrama Secuencia: CU-18 Administrar postulación

Fuente: Elaboración Propia

Tabla 2.44. Contrato: CU-18 Administrar Ponencia

Nombre de la operación	enviar_postulación(titulo, descripción, resumen)
Referencia Cruzada	CU-18
Precondición	El evento debe estar recibiendo postulaciones, y el usuario no debe haber enviado una postulación duplicada.
Postcondición	Se registró la postulación del usuario y quedó en estado "pendiente de revisión".

Fuente: Elaboración Propia

2.8.19. Caso de Uso: CU-19 Cancelar ponencia aceptada

Tabla 2.45. Caso de uso narrativo: CU-19 Cancelar ponencia aceptada

Caso de uso:	Cancelar ponencia aceptada
Código:	CU-19
Resumen:	El expositor por decisión personal decide cancelar una ponencia que ya había sido aceptada.
Tipo:	Evidente
Referencias Cruzadas:	N/A
Actor:	Expositor
Precondiciones:	<ul style="list-style-type: none"> - El actor debe de estar autenticado - Debe de tener una ponencia aceptada - El evento debe de estar seleccionado
Flujo Normal	
Actor	Sistema
1.- El actor selecciona la opción de cancelar ponencia.	
	2.- El sistema despliega el listado de ponencia aceptada y detalles de esta.
3.- El actor selecciona una ponencia a cancelar.	
	4.- El sistema solicita motivo de la cancelación.
5.- El actor da su motivo para cancelar.	
	6.- El sistema solicita contraseña para confirmación.
7.- El actor introduce la contraseña.	
	8.- El sistema valida la contraseña.
	9.- cancela la postulación.
	10.- muestra un mensaje de éxito.
Postcondición:	Se canceló la ponencia.

Fuente: Elaboración propia

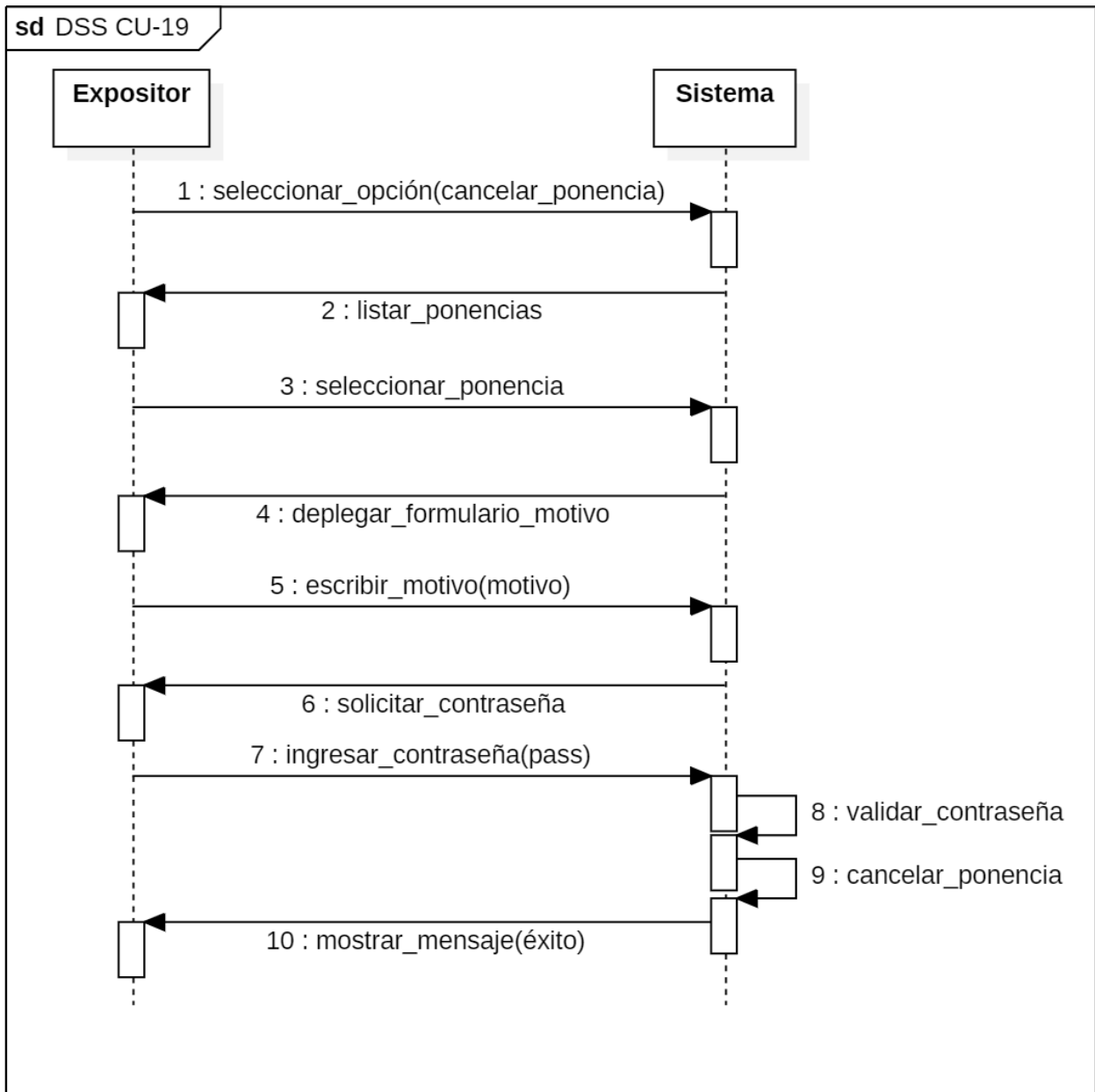


Figura 2-22. Diagrama Secuencia: CU-19 Cancelar Ponencia
Fuente: Elaboración Propia

Tabla 2.46. Contrato CU-19 Cancelar Ponencia

Nombre de la operación	cancelar_ponencia(ponencia_id)
Referencia Cruzada	CU-19
Precondición	La ponencia debe haber sido previamente aceptada y no debe haberse realizado aún.
Postcondición	Se canceló la ponencia y se actualizó su estado en el sistema.

Fuente: Elaboración propia

PITULO 3. ASPECTO RELEVANTES DEL DISEÑO PARA LA ALTERNATIVA SELECCIONADA

3. ASPECTO RELEVANTES DEL DISEÑO PARA LA ALTERNATIVA SELECCIONADA

En este capítulo se presenta aspectos relevantes del diseño para la alternativa seleccionada.

3.1. Arquitectura de software

En el diseño del sistema se ha decidido implementar una combinación de arquitectura por capas y arquitectura por niveles. Esta elección responde a la necesidad de contar con una estructura clara, modular y fácilmente escalable, acorde a los requerimientos funcionales y técnicos del sistema.

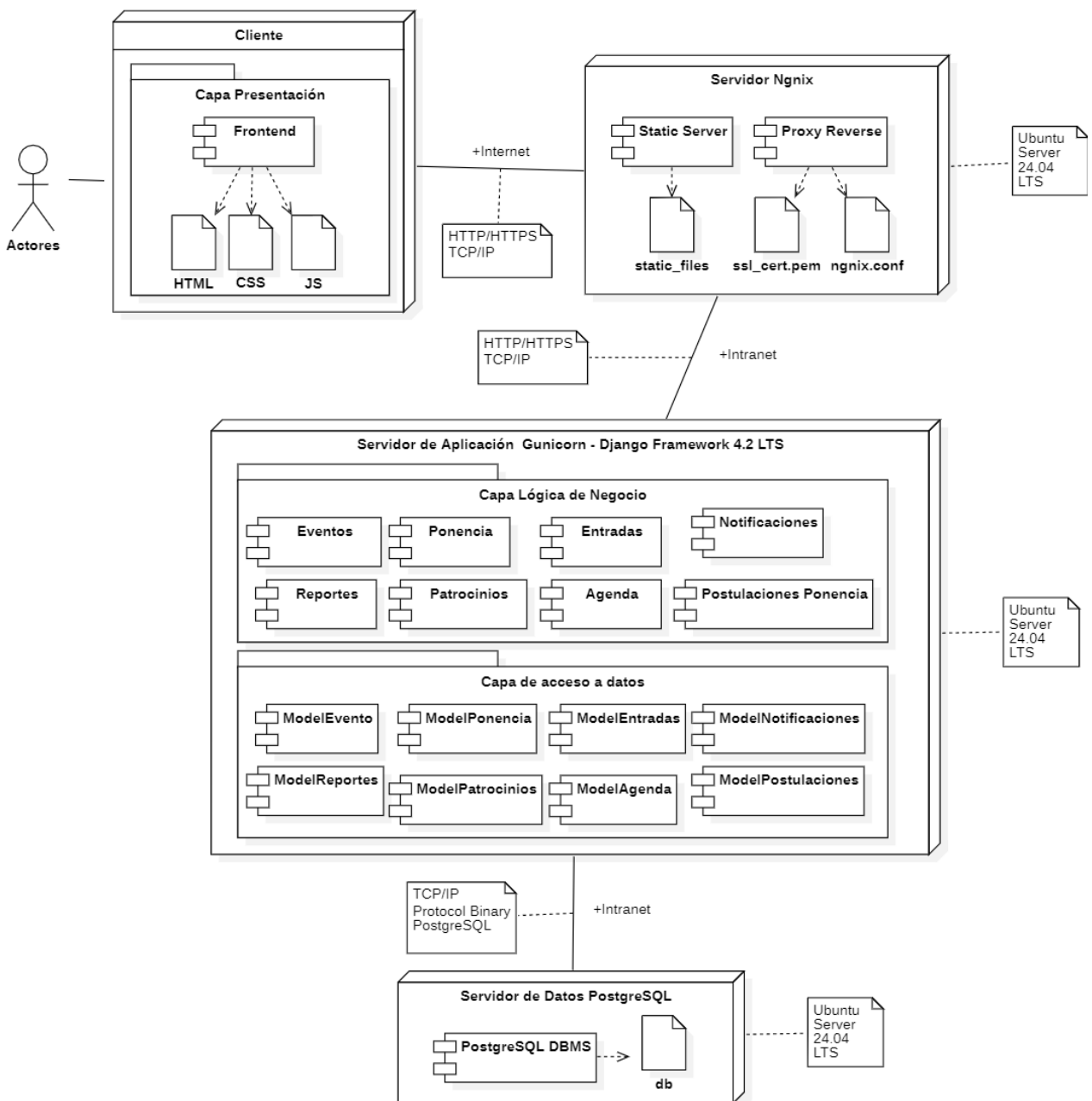


Figura 3-1. Diagrama de despliegue
Fuente: Elaboración Propia

- **Arquitectura de capas:** La arquitectura por capas permite organizar el software en secciones bien definidas, como la capa de presentación, la lógica de negocio y el acceso a datos. Cada capa se encarga de una responsabilidad específica, lo que facilita el mantenimiento, la reutilización del código y la separación de preocupaciones. Se dejaron los componentes de software según la capa.
- **Arquitectura por niveles:** Por otro lado, la arquitectura por niveles complementa este enfoque al distribuir físicamente las distintas capas en entornos separados, como el cliente, el servidor de aplicaciones y la base de datos. Esto mejora la seguridad, el rendimiento y permite una mayor flexibilidad en el despliegue del sistema. Se agregó una capa más.

La incorporación de un servidor de balanceo de carga entre el cliente y el servidor de aplicaciones permite distribuir eficientemente las solicitudes, mejorando el rendimiento y la disponibilidad del sistema. Además, aporta tolerancia a fallos y facilita la escalabilidad sin afectar la experiencia del usuario.

- **Sistema Operativo**

Para el sistema operativo de producción, se empleará Ubuntu Server 24.04 LTS con soporte extendido para actualizaciones de seguridad hasta el año 2029, Esta elección se basa en su gran comunidad y documentación de ayuda, además de contar con soporte de pago ofrecido por la empresa canonical y es de libre utilización.

- **Lenguajes de programación y tecnologías.**

El lenguaje de programación elegido es Python en su versión 3.11 que tiene soporte hasta el año 2027. Esta elección se debe a la Corporación PythonChile que difunde la utilización del lenguaje Python.

La elección de tecnología web es el uso del framework Django en su versión 4.2 LTS con soporte extendido hasta abril del año 2026, Django cuenta con una arquitectura robusta llamada MTV que es una leve variación de la arquitectura MVC. Además, cuenta con componentes y herramientas que facilitan el desarrollo, con un enfoque en seguridad y al ser de propósito general se encuentran en un repositorio general componente de terceros que se pueden utilizar, también cabe destacar la gran comunidad y popularidad de Django.

Se escogió como servidor web a nginx 1.24.0 o una versión superior compatible, la elección se debe a que incorpora la característica de cache de contenido estático, lo cual es ayuda a ahorrar recursos de procesamiento al evitar consultas y tiene un consumo de recursos más bajo en comparación con su competidor Apache Web Server En la elección de base de datos se optó por PostgreSQL 11, ya que aparte de ser una base de datos

relacional incorpora soporte para documentos tipos json, xml y key-value, características que pueden ser útil en el futuro.

- **Herramientas necesarias para el desarrollo del proyecto**

Las Herramientas de desarrollo incluyen:

- Git versión 2.40.0 o superior para el control de versiones del código.
- Los servicios de github.com para los repositorios del código.
- El editor de código Visual Stude Code 1.51 o superior, con complementos de detección de errores(linter) y formateador de código en Python y complementos de auto completado para Django.
- Como entorno virtual se utilizará la herramienta integrada en Python3 llamada venv.

3.2. Diseño de datos

A continuación, se presentan los modelos de datos que se utilizarán en el proyecto, acompañados por el diagrama de clases, el modelo relacional y el diccionario de datos. En este último, se especificarán los tipos de datos que se emplearán.

3.2.1. Modelo de Clases

El modelo de clases general ofrece una visión holística del sistema, presentando las relaciones entre clases y los elementos clave a implementar. Esto permite una comprensión global de la estructura y funcionamiento del sistema.

Como licencia práctica, la clase Usuario ha sido representada varias veces en el diagrama con el fin de evitar una congestión visual causada por múltiples líneas cruzadas. Esta duplicación visual mejora la legibilidad sin alterar el significado del modelo.

Además, el diagrama fue dividido en tres secciones principales, con el objetivo de facilitar su lectura. Esta segmentación permite una visualización más clara de las relaciones entre clases, reduce la complejidad visual y mejora la comprensión del diseño.

Por último, se ha aplicado la convención de nomenclatura *snake_case* en los atributos y métodos, en línea con las buenas prácticas recomendadas para el desarrollo en Python, favoreciendo la coherencia y mantenibilidad del código a largo plazo y se ha utilizado la notación "..." en algunas clases para indicar que existen atributos u operaciones adicionales que no se muestran en el diagrama.

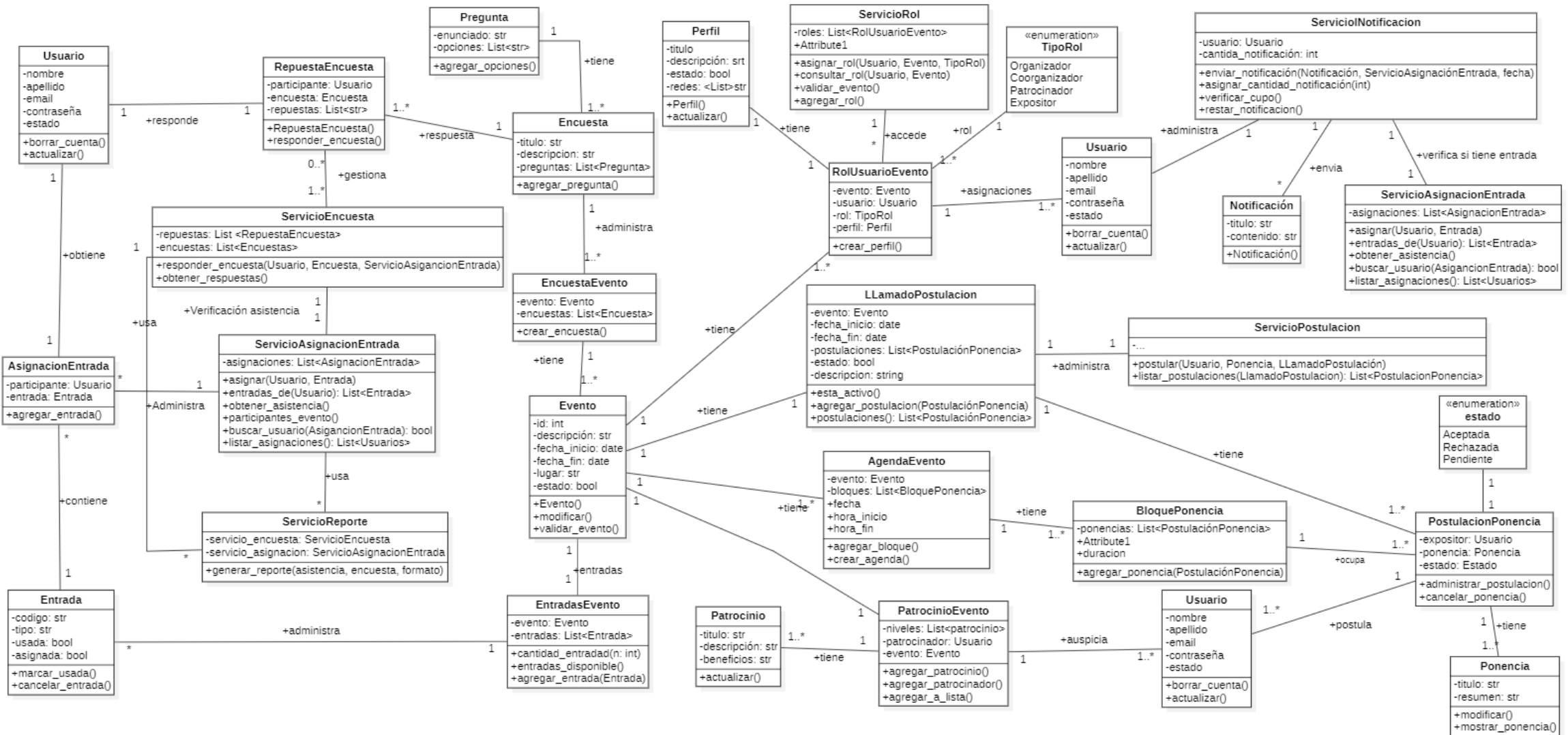


Figura 3-2 Modelo Clases General
Fuente: Elaboración Propia

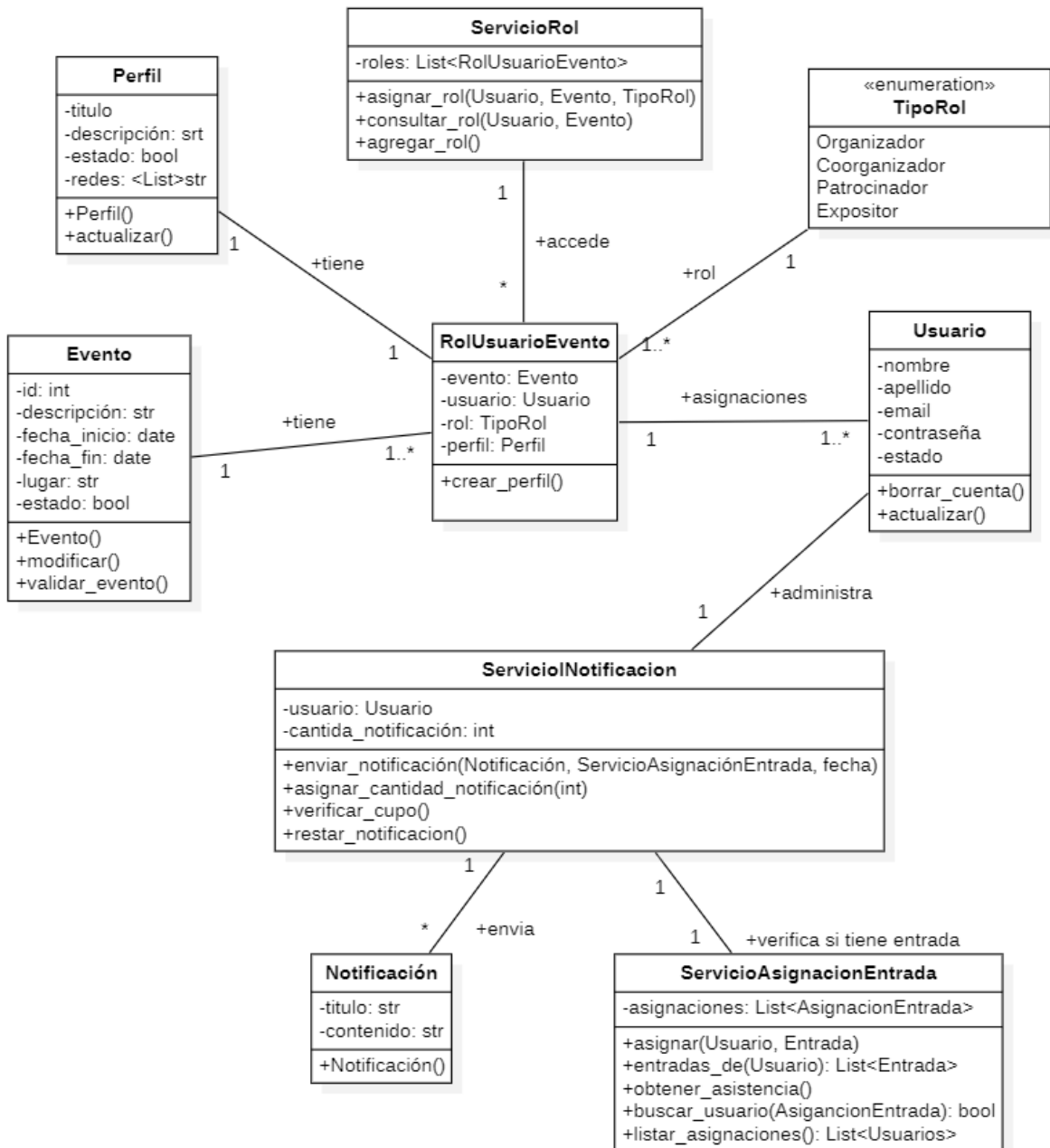


Figura 3-3. Modelo Clases de Roles y Notificación

Fuente: Elaboración Propia

El primer diagrama del modelo de clases agrupa los elementos fundamentales relacionados con la gestión de usuarios, roles y notificaciones dentro del sistema. Incluye clases como usuario, perfil, rol_usuario_evento y tipo_rol las cuales permiten representar de forma flexible los distintos roles que un usuario puede asumir en un evento, como expositor, organizador o patrocinador. También se incorpora la clase evento que actúa como eje central de las relaciones.

Por otro lado, se integran componentes relacionados con la lógica de notificaciones, como control_notificacion y notificacion, así como los servicios servicio_rol y servicio_asignacion_entrada que encapsulan comportamientos específicos. Esta

estructuración facilita una separación clara entre las responsabilidades de cada clase y promueve un diseño coherente con principios como el de responsabilidad única (SRP) y la inversión de dependencias (DIP) del enfoque SOLID

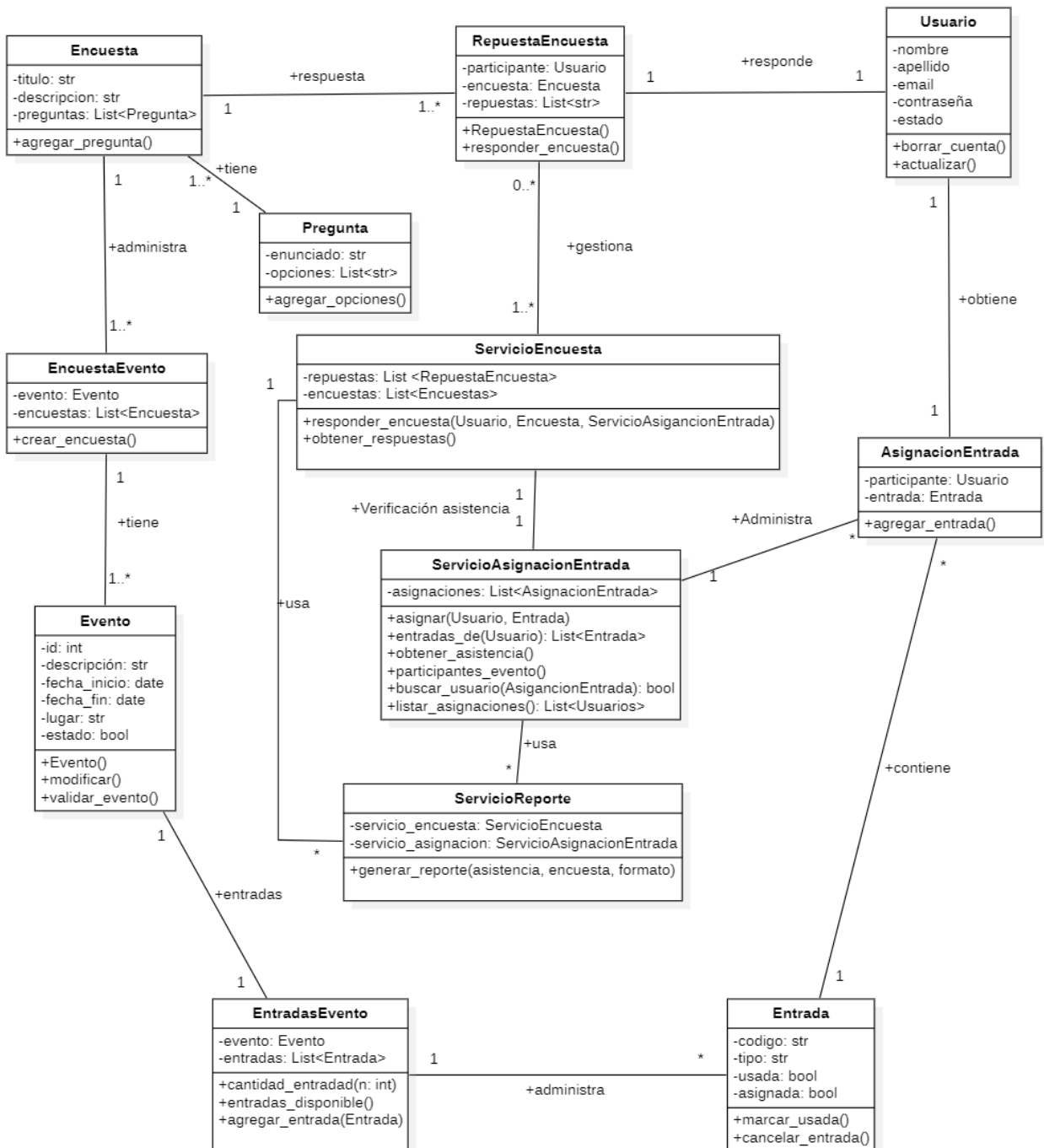


Figura 3-4. Modelo Clases de encuesta y entradas

Fuente: Elaboración propia

El segundo diagrama agrupa las clases relacionadas con la gestión de encuestas y asignación de entradas. Incluye encuesta, pregunta, respuesta_encuesta y encuesta_evento, que permiten vincular formularios a eventos específicos y recopilar respuestas de los usuarios.

También incorpora entrada y asignacion_entrada, para modelar la entrega de accesos a los eventos. Los servicios servicio_encuesta y servicio_asignacion_entrada encapsulan la lógica de negocio correspondiente.

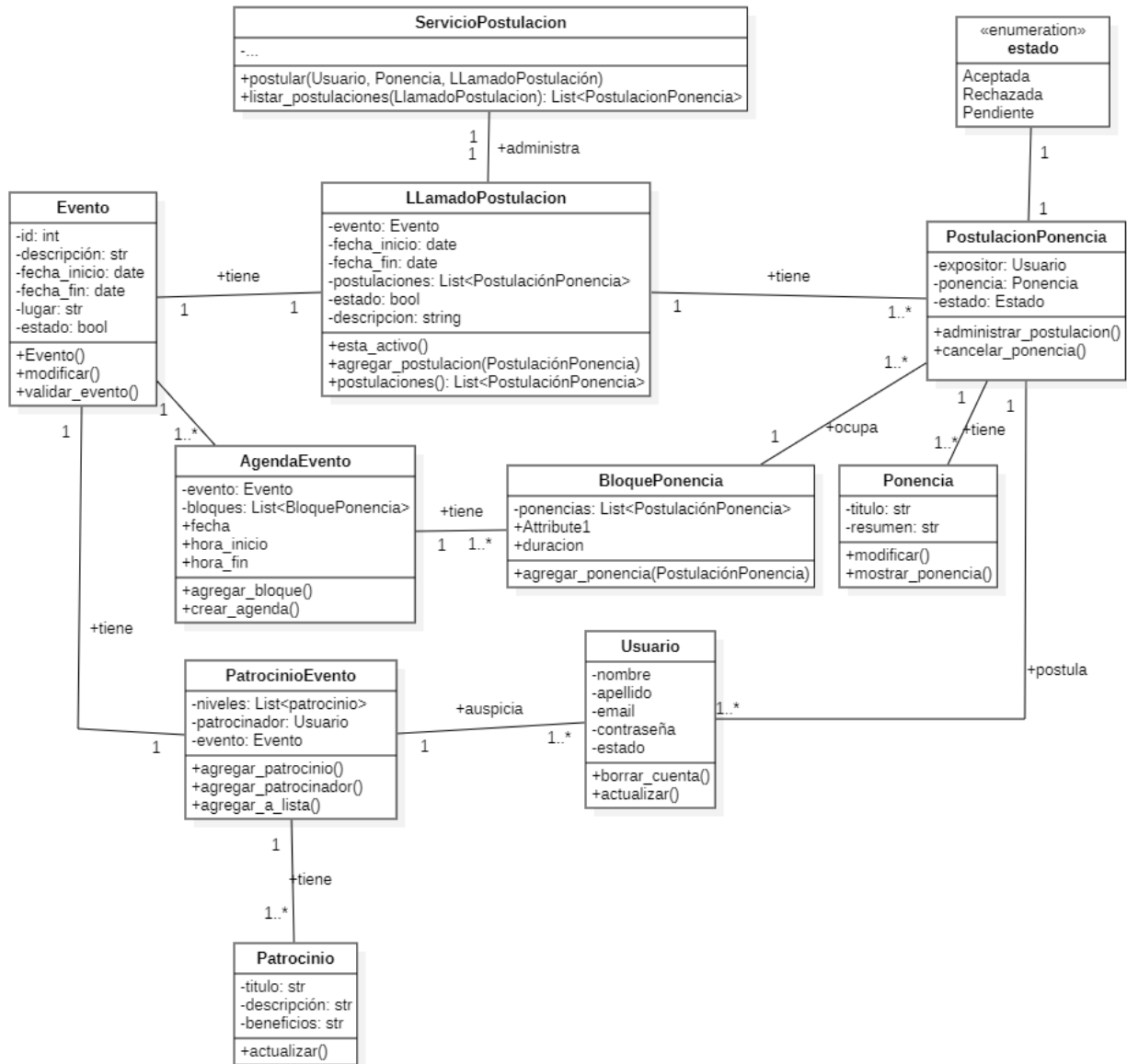


Figura 3-5. Modelo de clases postulación, ponencia y patrocinio
Fuente: Elaboración propia

El tercer diagrama se enfoca en la gestión de ponencias y patrocinios. Incluye clases como agenda_evento y bloque_horario para organizar la programación, así como llamado_postulacion, postulacion_ponencia y ponencia para gestionar el proceso de postulación de ponencia de los expositores. También se representa patrocinio y patrocinio_evento para vincular auspicios a eventos específicos.

El servicio servicio_postulacion centraliza la lógica relacionada con las postulaciones. Esta estructura facilita el control del flujo del evento y la interacción con los distintos actores involucrados.

3.2.2. Modelo Dato Relacional

El modelo relacional es una representación estructurada del sistema de gestión de eventos, donde la información se organiza en tablas relacionadas para administrar usuarios, roles, ponencias, entradas, encuestas y patrocinadores de forma coherente.

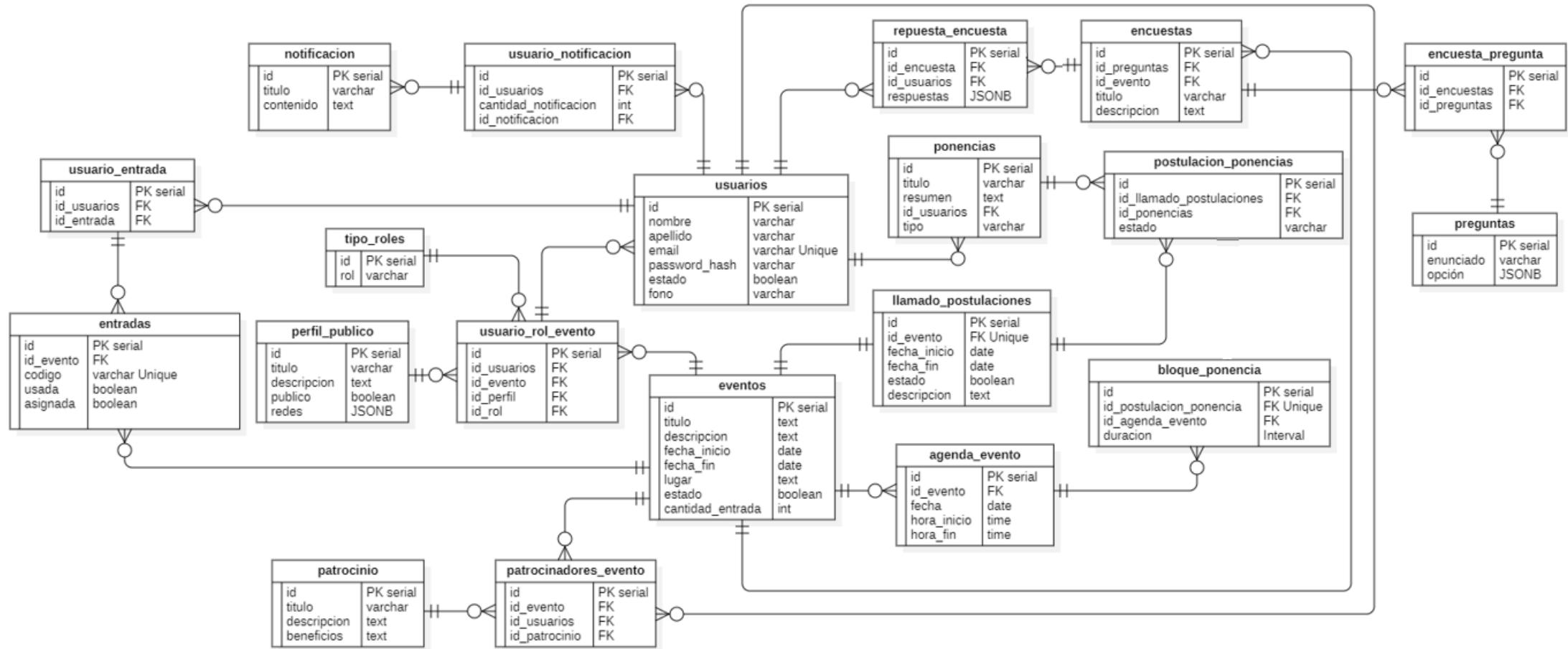


Figura 3-6. Modelo relacional

Fuente: Elaboración Propia

3.2.3. Diccionario de datos

El diccionario de datos sirve para documentar detalladamente los elementos del modelo relacional.

Tabla 3.1. Diccionario de Datos: usuarios

Nombre:	usuarios	
Descripción:	Almacena la información personal y de autenticación de cada usuario registrado en el sistema.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único
nombre	Varchar(50)	Nombre. Valor no nulo.
apellido	Varchar(50)	Nombre. Valor no nulo.
password_hash	Varchar(150)	Caracteres secretos y cifrados. Valor no nulo y único.
email	Varchar(254)	Correo electrónico del usuario. Valor no nulo.
fono	Varchar(15)	Número de teléfono. Valor no nulo.
estado	Boolean	Si está activo (True) o desactivado (False)

Fuente: Elaboración propia

Tabla 3.2. Diccionario de Datos: usuario_rol_evento

Nombre:	usuario_rol_evento	
Descripción:	Representa la asignación de un rol específico a un usuario dentro de un evento determinado.	
Clave Primaria:	id	
Claves Foráneas:	id_usuario → Usuarios id_evento → Eventos id_perfil → Perfil_publico id_rol → Tipo_rols	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_usuarios	Integer	Identificador del usuario al que se le asigna un rol en un evento. Valor no nulo.
id_evento	Integer	Identificador del evento en el que el usuario participa con un rol. Valor no nulo. Único
id_perfil	Integer	Identificador del perfil específico del usuario dentro del evento (si aplica).
id_rol	Integer	Identificador del rol que el usuario desempeña en el evento. Valor no nulo. Único.

Fuente: Elaboración propia

Tabla 3.3. Diccionario de Datos: tipo_rols

Nombre:	tipo_rols	
Descripción:	Define los diferentes tipos de roles que pueden ser asignados a los usuarios dentro del sistema, como organizador, expositor, patrocinador o coorganizador.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
rol	Varchar(40)	Indica el nombre o tipo de rol asignado. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.4. Diccionario de Datos: perfil_publico

Nombre:	perfil_publico	
Descripción:	Almacena información adicional y detallada asociada a un usuario, dependiendo del rol que desempeña dentro de un evento.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
titulo	Varchar(60)	Nombre y Apellido. Valor no nulo.
descripcion	Text (100)	Caracteres secretos y cifrados. Valor no nulo.
publico	Boolean	Correo electrónico del usuario. Valor no nulo.
redes	JSONB	Redes sociales.

Fuente: Elaboración propia

Tabla 3.5. Diccionario de Datos: eventos

Nombre:	eventos	
Descripción:	Almacena la información principal de los eventos gestionados en el sistema, incluyendo su nombre, descripción, fechas, ubicación y estado.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
titulo	Varchar(100)	Nombre o título del evento. Valor no nulo.
descripcion	Text	Breve resumen o detalle del contenido y propósito del evento. Valor no nulo.
fecha_inicio	Date	Fecha en que comienza el evento. Valor no nulo.
fecha_fin	Date	Fecha en que finaliza el evento. Valor no nulo.
lugar	Varchar(150)	Ubicación física o virtual donde se realiza el evento.
estado	Boolean	Si está activo (True) o desactivado (False).
cantidad_entrada	Integer	Cantidad de entrada que tiene el evento. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.6. Diccionario de Datos: usuario_notificacion

Nombre:	usuario_notificacion	
Descripción:	Cantidad de notificaciones que tiene un usuario para enviar.	
Clave Primaria:	id	
Claves Foráneas:	id_usuario → usuarios id_notificacion → notificacion	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único. Valor no nulo
id_usuario	Integer	Identificador de la notificación enviada al usuario. Valor no nulo.
cantidad_notificacion	Integer	Número de veces que se puede enviar una notificación vía mail. Valor no nulo.
id_notificacion	Integer	Correo electrónico del usuario. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.7. Diccionario de Datos: notificación

Nombre:	notificación	
Descripción:	Almacena los mensajes del sistema que pueden ser enviados o mostrados a los usuarios, como alertas, recordatorios o actualizaciones de eventos.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
titulo	Varchar(50)	Breve encabezado o asunto de la notificación. Valor no nulo.
contenido	Text	Texto descriptivo o cuerpo del mensaje que contiene la información completa de la notificación. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.8. Diccionario de Datos: entradas

Nombre:	entradas	
Descripción:	Representa una entrada o boleto para un evento, incluyendo su tipo, estado de uso y asignación a un usuario.	
Clave Primaria:	id	
Claves Foráneas:	Id_evento → eventos	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
codigo	Varchar(100)	Identificador único de la entrada (puede ser alfanumérico). Valor no nulo.
tipo	Varchar(100)	Categoría o clase de la entrada (por ejemplo: general, VIP, estudiante). Valor no nulo.
usada	Boolean	Indica si la entrada ya fue utilizada para acceder al evento (TRUE o FALSE). Valor no nulo.
asignada	Boolean	Indica si la entrada ha sido asignada a un usuario (TRUE o FALSE). Valor no nulo.
id_evento	Integer	Identificador que vincula esta entrada individual con el evento específicos definido. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.9. Diccionario de Datos: usuario_entrada

Nombre:	usuario_entrada	
Descripción:	Representa la relación entre un usuario y una entrada específica, indicando qué entrada fue asignada a qué usuario.	
Clave Primaria:	id	
Claves Foráneas:	id_usuario → usuarios id_entrada → entradas	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_usuario	Integer	Identificador del usuario al que se le ha asignado la entrada. Valor no nulo.
id_entrada	Integer	Identificador de la entrada asignada al usuario. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.10. Diccionario de Datos: patrocinio

Nombre:	patrocinio	
Descripción:	Almacena información sobre los distintos tipos o niveles de patrocinio disponibles para los eventos, incluyendo su nombre, descripción y los beneficios que ofrece a los patrocinadores.	
Clave Primaria:	id	
Claves Foráneas:	n/a	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
titulo	Varchar(100)	Nombre o título del plan de patrocinio (por ejemplo: "Oro", "Plata", "Patrocinador Principal").
descripción	Text	Detalle general del patrocinio y su propósito o alcance. Valor no nulo.
beneficios	Text	Lista o resumen de los beneficios que se ofrecen al patrocinador (como visibilidad en el evento, stand, menciones, etc.)Valor no nulo.

Fuente: Elaboración propia

Tabla 3.11. Diccionario de Datos: patrocinadores_evento

Nombre:	patrocinadores_evento	
Descripción:	Representa la relación entre usuarios que actúan como patrocinadores y los eventos que patrocinan, especificando además el tipo o nivel de patrocinio que ofrecen.	
Clave Primaria:	id	
Claves Foráneas:	id_evento → eventos id_usuario → usuarios id_patrocinio → patrocinio	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único .
id_evento	Integer	Identificador del evento que recibe el patrocinio. Valor no nulo.
id_usuario	Integer	Identificador del usuario que actúa como patrocinador. Valor no nulo.
id_patrocinio	Integer	Identificador del tipo o nivel de patrocinio otorgado por el usuario al evento). Valor no nulo.

Fuente: Elaboración propia

Tabla 3.12. Diccionario de Datos: agenda_evento

Nombre:	agenda_evento	
Descripción:	Define la planificación del evento, relacionando los bloques de horario con un evento específico para organizar su cronograma.	
Clave Primaria:	id	
Claves Foráneas:	id_evento → eventos id_bloque_horario → bloque_ponencia	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_evento	Integer	Identificador del evento al que pertenece esta agenda. Valor no nulo.
id_bloque_ponencia	Integer	Identificador del bloque ponencia programado dentro del evento. Valor no nulo.
fecha	Date	Identificador del día que se realizara las actividades. No nulo.
hora_inicio	Time	Hora de inicio de las actividades del día. No nulo.
hora_fin	Time	Hora fin de las actividades del día. No nulo.

Fuente: Elaboración propia

Tabla 3.13. Diccionario de Datos: bloque_horario

Nombre:	bloque_ponencia	
Descripción:	Representa una ponencia y su tiempo dentro de la agenda de un evento, en el cual puede asignarse una ponencia previamente postulada y aceptada.	
Clave Primaria:	Id	
Claves Foráneas:	id_agenda_evento → agenda_evento id_postulacion_ponencia → bloque_ponencia	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_agenda_evento	Integer	Identificador de la agenda a la que pertenece este bloque horario. Valor no nulo.
id_postulacion_ponencia	Integer	Identificador de la ponencia asignada a este bloque horario (si aplica).
duracion	Interval	La duración de la ponencia.

Fuente: Elaboración propia

Tabla 3.14. Diccionario de Datos: postulación_ponencias

Nombre:	postulacion_ponencias	
Descripción:	Representa un intervalo de tiempo dentro de la agenda de un evento, en el cual puede asignarse una ponencia previamente postulada y aceptada.	
Clave Primaria:	Id	
Claves Foráneas:	id_llamado_postulaciones → llamado_postulaciones id_ponencia → ponencias	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_llamado_postulaciones	Integer	Identificador del llamado a postulación al que responde la ponencia. Valor no nulo.
id_ponencia	Integer	Identificador de la ponencia propuesta por el expositor. Valor no nulo.
Estado	Varchar(10)	Estado actual de la postulación (por ejemplo: pendiente, aceptada, rechazada). Valor no nulo.

Fuente: Elaboración propia

Tabla 3.15. Diccionario de Datos: llamado_postulaciones

Nombre:	llamado_postulaciones	
Descripción:	Representa una convocatoria oficial dentro de un evento para que los expositores envíen propuestas de ponencias, con fechas límite y estado actual del proceso.	
Clave Primaria:	id	
Claves Foráneas:	id_evento → eventos	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_evento	Integer	Identificador del evento al que pertenece este llamado a postulación. Valor no nulo.
fecha_inicio	Date	Fecha en la que comienza el período de recepción de postulaciones. Valor no nulo.
fecha_fin	Date	Fecha en la que finaliza el período de recepción de postulaciones. Valor no nulo.
estado	Boolean	Estado actual del llamado (por ejemplo: abierto, cerrado, evaluando). Valor no nulo.
descripcion	Text	Información adicional o instrucciones sobre el llamado a postulación. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.16. Diccionario de Datos: ponencias

Nombre:	ponencias	
Descripción:	Almacena la información básica de una ponencia propuesta por un usuario, incluyendo su título, resumen y el usuario que la propone.	
Clave Primaria:	id	
Claves Foráneas:	id_usuario → usuarios	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
Título	Varchar(100)	Título de la ponencia propuesta. Valor no nulo.
resumen	Varchar(255)	Breve descripción o sinopsis del contenido de la ponencia. Valor no nulo.
id_usuarios	Integer	Identificador del usuario que presenta la ponencia (expositor). Valor no nulo.
tipo	Varchar(30)	Tipo de la ponencia, charla, keynote, taller.

Fuente: Elaboración propia

Tabla 3.17. Diccionario de Datos: encuestas

Nombre:	encuestas	
Descripción:	Representa una encuesta asociada a un evento, compuesta por una o más preguntas, y diseñada para recopilar opiniones, evaluaciones o información de los participantes.	
Clave Primaria:	id	
Claves Foráneas:	id_preguntas → encuesta_pregunta id_evento → evento	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_preguntas	Integer	Identificador o referencia a las preguntas asociadas a la encuesta. Valor no nulo.
id_evento	Integer	Identificador del evento al que pertenece la encuesta. Valor no nulo.
titulo	Varchar(20)	Título o nombre de la encuesta. Valor no nulo.
descripcion	Varchar(250)	Texto explicativo sobre el propósito o contenido de la encuesta. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.18. Diccionario de Datos: pregunta

Nombre:	preguntas	
Descripción:	Almacena las preguntas que forman parte de encuestas, incluyendo su texto principal (enunciado) y, si corresponde, las opciones posibles de respuesta.	
Clave Primaria:	id	
Claves Foráneas:	No hay	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
enunciado	Varchar(120)	Texto que contiene la formulación o contenido principal de la pregunta. Valor no nulo.
opcion	JSONB	Conjunto de posibles respuestas, almacenadas como texto separado por comas o en formato estructurado (ej. JSON), en caso de preguntas de opción múltiple o selección. Valor no nulo.

Fuente: Elaboración propia

Tabla 3.19. Diccionario de Datos: encuesta_pregunta

Nombre:	encuesta_pregunta	
Descripción:	Relaciona las preguntas con una encuesta específica, permitiendo estructurar el contenido de la encuesta y definir el orden de aparición de cada pregunta	
Clave Primaria:	id	
Claves Foráneas:	id_encuestas → encuestas id_preguntas → preguntas	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_encuestas	Integer	Identificador de la pregunta que se incluye en la encuesta.
id_preguntas	Integer	Número que indica la posición o secuencia de la pregunta dentro de la encuesta.

Fuente: Elaboración propia

Tabla 3.20. Diccionario de Datos: repuesta_encuesta

Nombre:	repuesta_encuesta	
Descripción:	Almacena las respuestas dadas por un usuario a una encuesta específica, incluyendo el identificador del usuario, la encuesta y el conjunto de respuestas proporcionadas.	
Clave Primaria:	id	
Claves Foráneas:	id_encuestas → encuestas id_usuarios → usuarios	
Estructura		
Nombre	Tipo	Descripción
id	Serial	Identificador único.
id_encuestas	Integer	Identificador de la encuesta a la que pertenece la pregunta.
id_usuarios	Integer	Número que indica la posición o secuencia de la pregunta dentro de la encuesta.
respuestas	JSONB	Conjunto de respuestas del usuario a las preguntas de la encuesta. Puede almacenarse en formato estructurado (por ejemplo, JSON).

Fuente: Elaboración propia

3.3. Diagrama de secuencia extendido

Los diagramas de secuencia extendido son fundamentales en el análisis y diseño de sistemas. Facilitan la visualización y comprensión detallada de las interacciones entre actores y los diversos objetos que participan en un proceso específico.

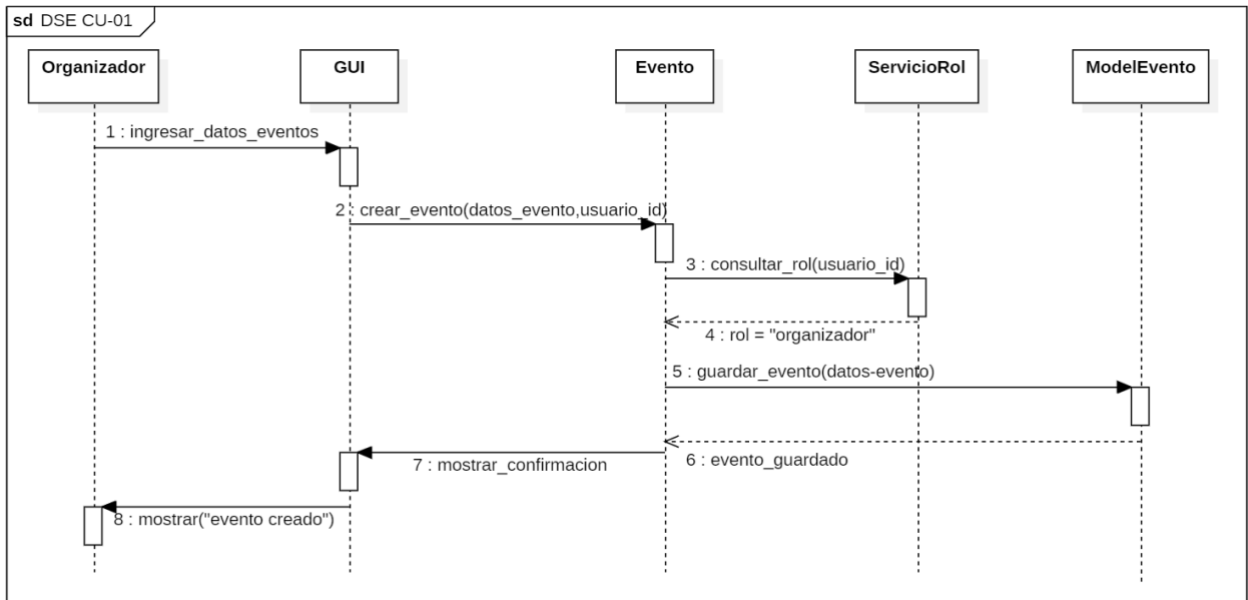


Figura 3-7. DSE: CU-01 Mantener evento
Fuente: Elaboración propia

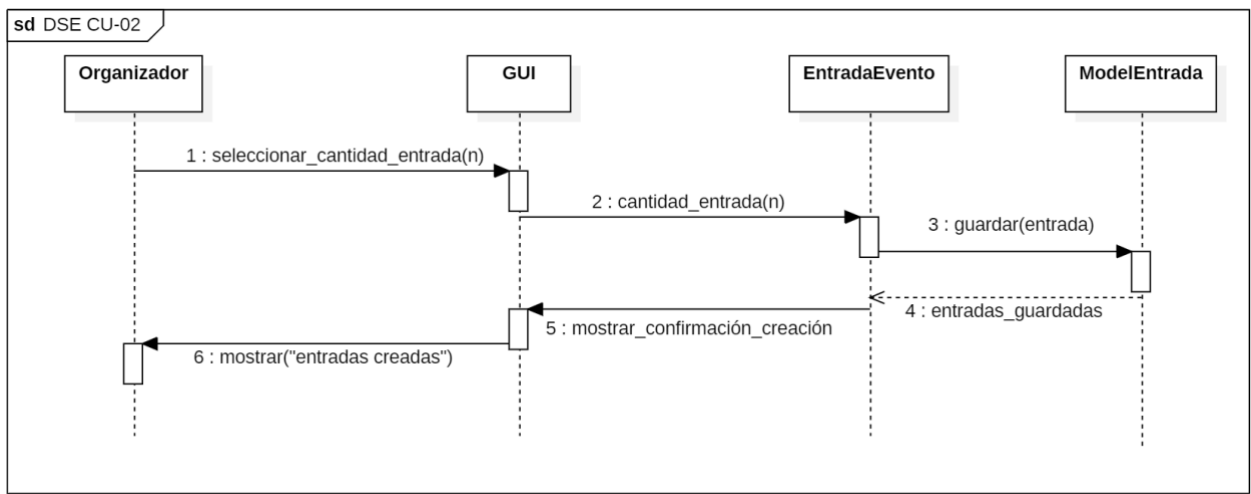


Figura 3-8 DSE: CU-02 Administrar entradas
Fuente: Elaboración propia

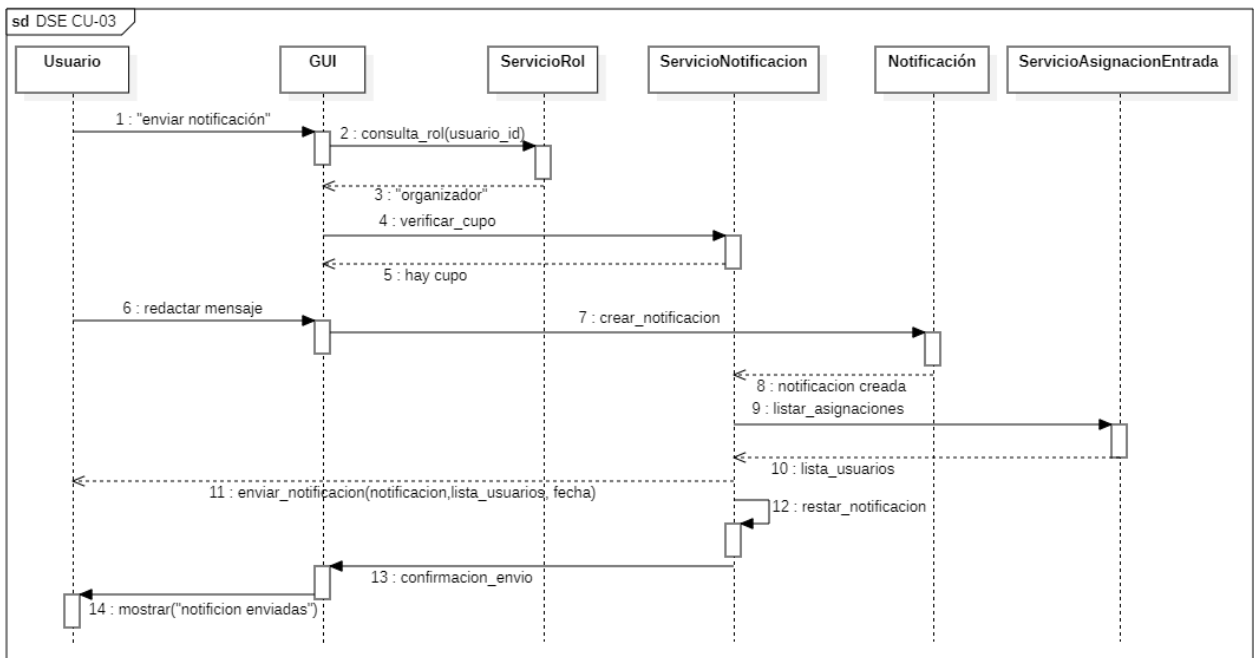


Figura 3-9 DSE: CU-03 Enviar notificación
Fuente: Elaboración Propia.

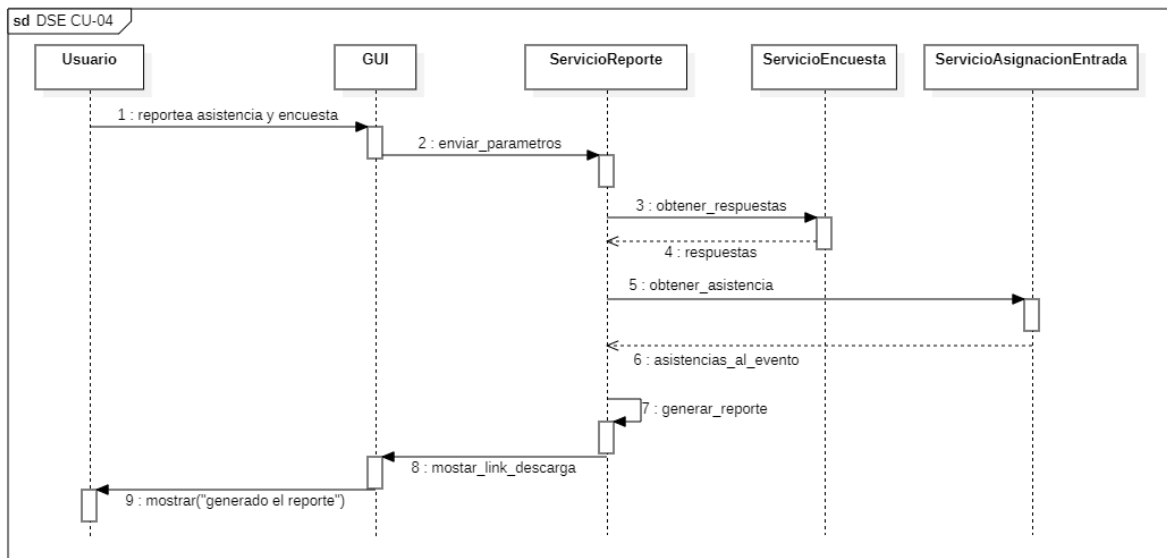


Figura 3-10 DSE: CU-04 Obtener reporte asistencia y encuesta
Fuente: Elaboración Propia

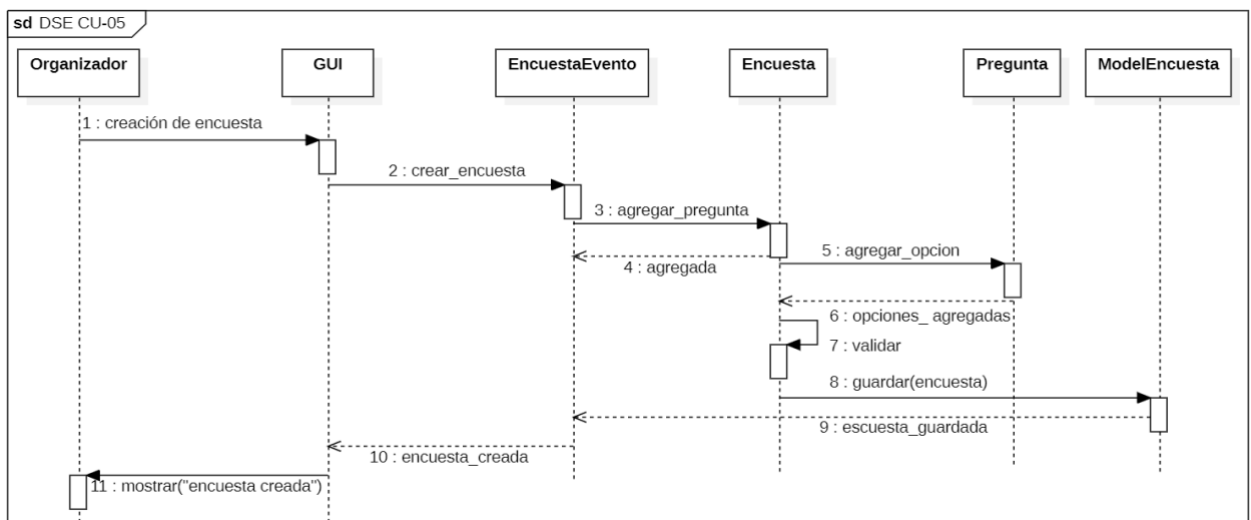


Figura 3-11 DSE: CU-05 Mantener Encuesta
Fuente: Elaboración Propia

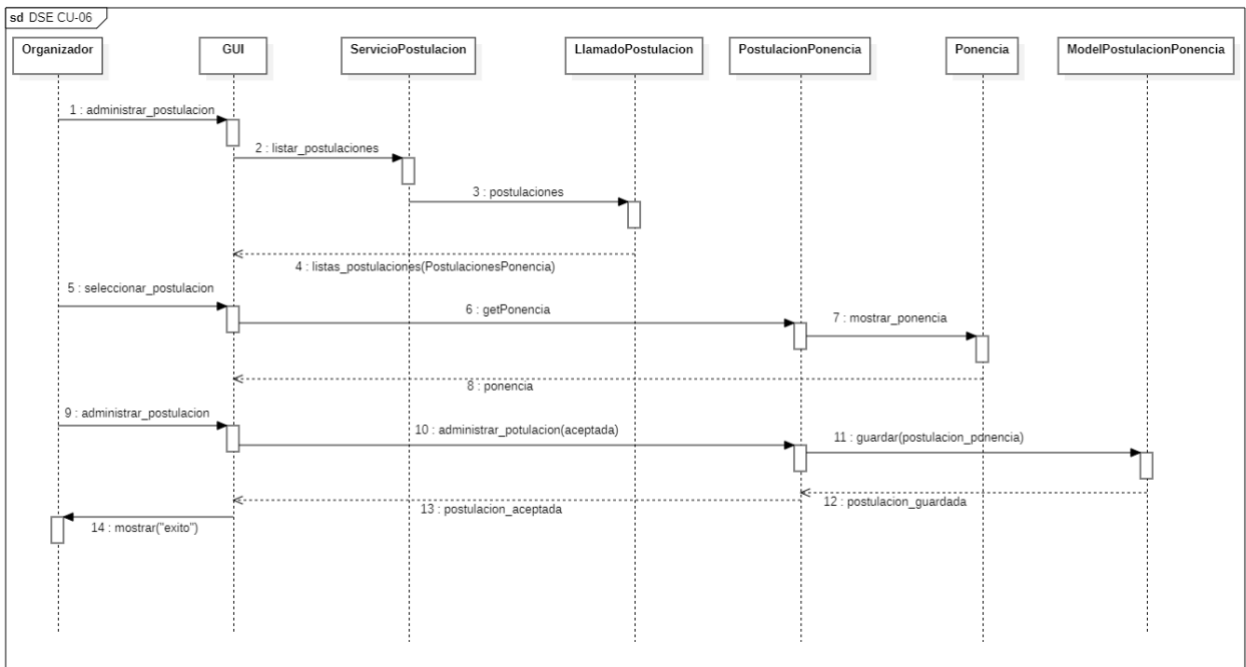


Figura 3-12 DSE: CU-06 Administrar postulación de ponencias
Fuente: Elaboración Propia

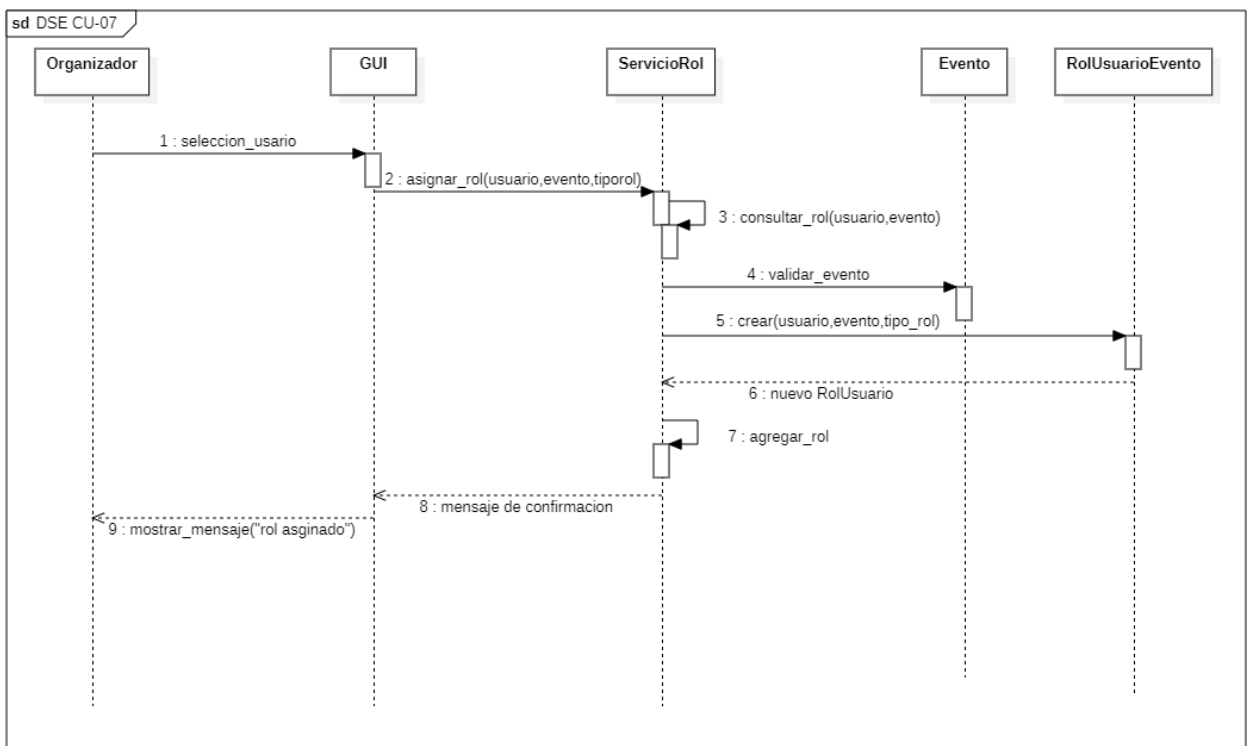


Figura 3-13 DSE: CU-07 Asignar rol
Fuente: Elaboración Propia

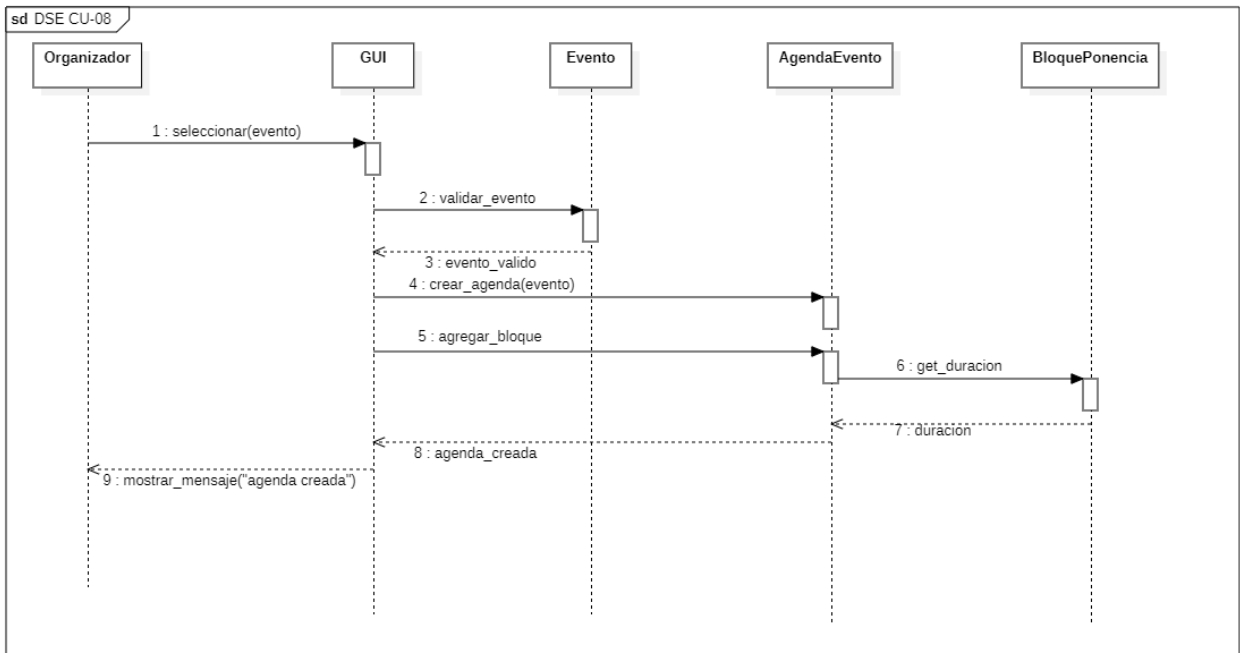


Figura 3-14 DSE: CU-08 Administrar agenda
Fuente: Elaboración Propia

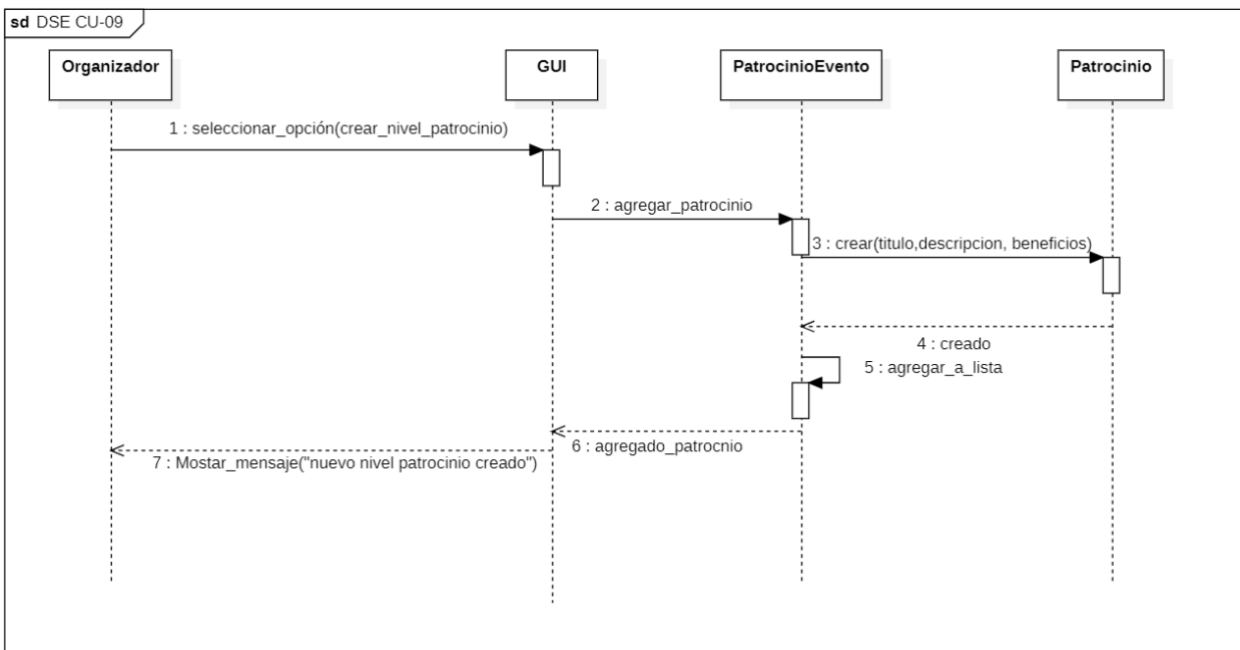


Figura 3-15 DSE: CU-09 Mantener nivel de patrocinio
Fuente: Elaboración Propia

3.4. Diseño de interfaz

Para el diseño de la interfaz se utilizarán los principios de usabilidad web de Jacob Nielsen.



Figura 3-16. 10 leyes de usabilidad de Jacob Nielsen

Fuente: <https://www.gluo.mx/blog/las-10-heuristicas-de-usabilidad-en-diseno-de-interfaz>

1. **Visibilidad del estado del sistema:** La web de gestión de eventos debe proporcionar retroalimentación visual clara para indicar el estado actual del sistema. Por ejemplo, mostrar mensajes de confirmación o error al realizar acciones como registrar un participante o agregar una ponencia.
2. **Coincidencia entre el sistema y el mundo real:** La terminología, las opciones y las instrucciones utilizadas en la web deben ser claras y coherentes con el contexto de los eventos y la gestión. Esto facilitará la comprensión y la interacción de los usuarios.
3. **Control y libertad del usuario:** La web debe permitir a los usuarios deshacer acciones no deseadas o salir de procesos sin problemas. Por ejemplo, incluir botones de cancelar o retroceder para que los usuarios puedan revertir acciones o salir de formularios sin completarlos.
4. **Consistencia y estándares:** La web debe seguir convenciones y estándares web establecidos para que los usuarios puedan comprender fácilmente cómo interactuar con ella. Mantener la consistencia en el diseño, la navegación y la estructura de la información es fundamental.
5. **Prevención de errores:** La web debe ayudar a prevenir errores proporcionando validaciones claras en los formularios y orientando a los usuarios sobre los datos que

se esperan. Además, se deben proporcionar mensajes de error descriptivos y sugerencias para solucionar los problemas.

6. **Reconocimiento en lugar de recuerdo:** Los usuarios deben poder reconocer fácilmente las opciones y acciones en lugar de tener que recordar información o instrucciones. Proporcionar menús claros, botones descriptivos y elementos visuales intuitivos facilitará la navegación y la interacción.
7. **Flexibilidad y eficiencia de uso:** La web debe permitir a los usuarios realizar tareas de manera eficiente y ofrecer opciones para adaptarse a diferentes niveles de experiencia. Por ejemplo, proporcionar atajos de teclado, funciones de búsqueda y personalización de la interfaz.
8. **Estética y diseño minimalista:** La web de gestión de eventos debe tener un diseño limpio y atractivo que no sobrecargue visualmente a los usuarios. El uso de espacios en blanco, la organización clara de la información y la priorización de los elementos más relevantes ayudarán a que los usuarios se centren en la tarea actual.
9. **Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Cuando ocurra un error, la web debe proporcionar mensajes de error claros y orientación para ayudar a los usuarios a comprender qué salió mal y cómo solucionarlo. También se pueden proporcionar enlaces o recursos de ayuda para la recuperación.
10. **Ayuda y documentación:** La web debe proporcionar una ayuda contextual y documentación relevante para que los usuarios puedan encontrar respuestas a sus preguntas o acceder a instrucciones detalladas cuando lo necesiten. Esto puede incluir preguntas frecuentes, tutoriales o manuales de usuario.

En la página principal del sistema de gestión de eventos, se observa una adecuada aplicación de la visibilidad del estado del sistema, ya que el menú superior y los elementos como el calendario y el mapa mantienen al usuario informado sobre la ubicación, fecha y opciones disponibles en todo momento.

La consistencia se refleja en el uso de un diseño familiar y estándar, con un menú en la parte superior, botones claramente identificados y una disposición lógica de los contenidos que facilita la navegación.

Además, el lenguaje claro y en español empleado en los textos y en los botones permite que los usuarios comprendan fácilmente las acciones que pueden realizar, como obtener una entrada o postular una charla. Finalmente, el minimalismo se manifiesta en una interfaz limpia y enfocada, donde solo se presentan los elementos esenciales, evitando la sobrecarga de información y favoreciendo una experiencia de usuario sencilla y directa.

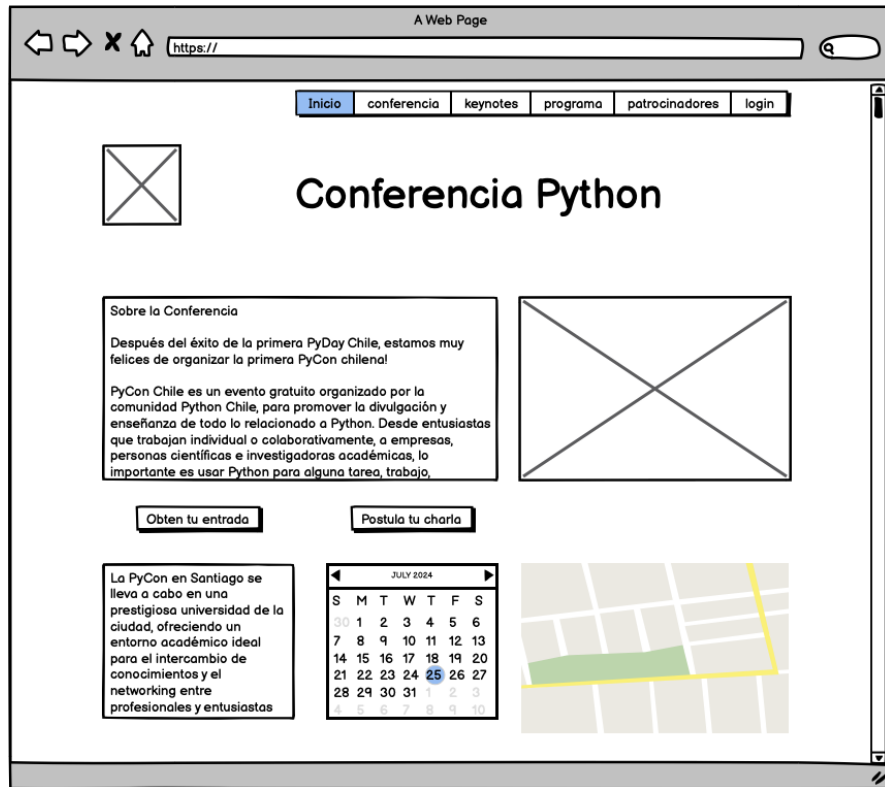


Figura 3-17. Pantalla principal
Fuente: Elaboración Propia

En la página de "Keynote", la visibilidad del sistema se asegura mediante una presentación clara que combina la imagen del expositor con un texto explicativo junto a ella, facilitando la comprensión inmediata del contenido. La consistencia se mantiene con un diseño ordenado y familiar, que ubica la información de forma lógica y fácil de seguir. El principio de reconocimiento mejor que recuerdo se aplica al mostrar iconos visibles de las redes sociales en una fila horizontal, permitiendo al usuario identificar rápidamente las plataformas disponibles. Además, el diseño minimalista evita distracciones, ofreciendo una experiencia visual limpia y centrada en la información esencial.

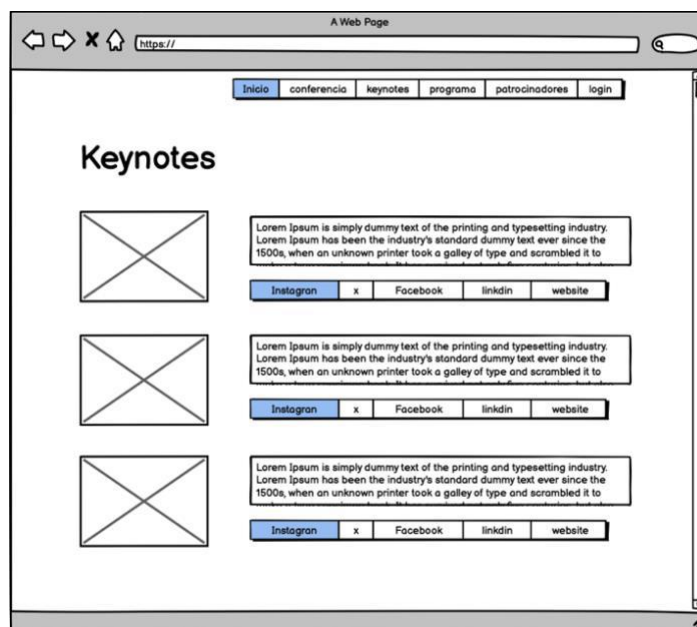


Figura 3-18. Pantalla Keynotes
Fuente: Elaboración Propia

En la página "Cronograma" del sistema de gestión de eventos, la visibilidad del estado se logra gracias a una presentación clara y estructurada de la agenda, facilitando la comprensión de las sesiones.

La consistencia se mantiene con un formato uniforme y encabezados claros, mientras que el principio de reconocimiento sobre recuerdo ayuda a los usuarios a identificar fácilmente la información sin memorizarla. La organización eficiente permite una navegación rápida tanto para usuarios nuevos como expertos, y el diseño minimalista evita la sobrecarga visual, favoreciendo una experiencia ágil y clara.

Python Programming Event Sche	Name	Description	Linked
Hora			
09:00 - 10:00	Giacomo Guilizzoni Founder & CEO	Keynote: The Future of Python	Linked
10:00 - 11:00	Marco Botton Tuttofare	Workshop: Building Efficient Python Applic	Linked
11:00 - 12:00	Mariah Maclochlan Better Half	Talk: Python in Data Science	Linked
12:00 - 13:00	Valerie Liberty Head Chef	Session: Python for Beginners	Linked
13:00 - 14:00	Lunch Break		
14:00 - 15:00	John Doe Senior Developer	Talk: Advanced Python Techniques	Linked
15:00 - 16:00	Jane Smith Data Scientist	Workshop: Machine Learning with Python	Linked
16:00 - 17:00	Emily Johnson AI Specialist	Session: AI Development with Python	Linked
17:00 - 18:00	Closing Remarks and Networ		

Figura 3-19. Pantalla Cronograma
Fuente: Elaboración Propia

En la página de "Auspiciadores" del sistema de gestión de eventos, se mantiene la visibilidad del estado gracias al menú superior constante que facilita la navegación. La consistencia se refleja en el uso del mismo diseño estructural que las páginas anteriores, con un título claro y destacado.

Cada auspiciante se presenta en un recuadro uniforme que incluye su logo, descripción y un botón para abrir su sitio web, lo que aplica el principio de reconocimiento mejor que recuerdo al mostrar elementos visuales claros. Además, el diseño minimalista y ordenado, con los recuadros alineados uno al lado del otro, evita la sobrecarga y facilita una visualización limpia y accesible para el usuario.

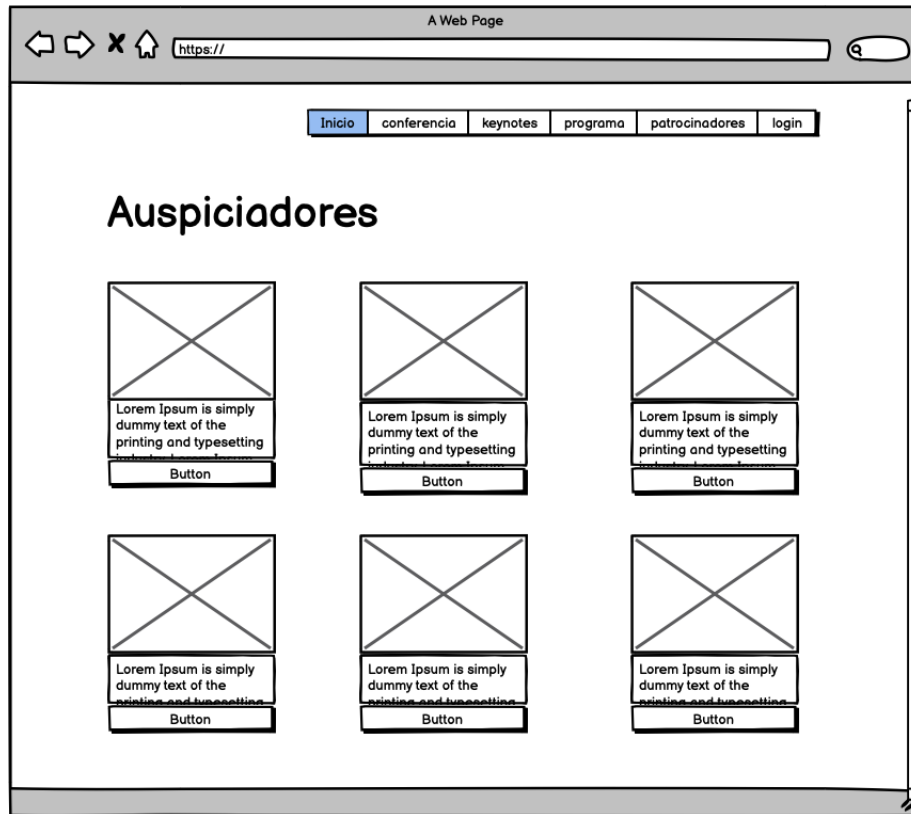


Figura 3-20. Pantalla Auspiciadores
Fuente: Elaboración propia

Considerando los 10 principios de usabilidad de Jakob Nielsen y analizando su aplicación en las distintas páginas del sistema de gestión de eventos, se puede concluir que el diseño general cumple de manera efectiva con principios clave como la visibilidad del estado, la consistencia, el reconocimiento mejor que recuerdo y el minimalismo, lo que facilita la navegación, comprensión y eficiencia para diferentes tipos de usuarios.

Sin embargo, aunque el diseño es claro y ordenado, podría beneficiarse de un mayor enfoque en la prevención y manejo de errores, así como en ofrecer ayuda o documentación accesible para usuarios que requieran soporte adicional. En conjunto, la interfaz proporciona una experiencia intuitiva y limpia, pero mejorar la retroalimentación y el soporte fortalecería aún más la usabilidad.

CONCLUSIONES Y RECOMENDACIONES

El desarrollo del sistema para la **Corporación Python Chile** permitió cumplir satisfactoriamente con el objetivo general del proyecto, que consistía en mejorar la gestión de eventos mediante una solución tecnológica integral. A lo largo del trabajo, se abordaron diversos desafíos propios del proceso organizativo, logrando avances significativos tanto a nivel funcional como en términos de aprendizaje técnico y práctico.

Entre los principales logros, se destacan los siguientes aspectos:

- Consolidación de información: Se logró centralizar todos los datos relevantes del evento en una sola plataforma, accesible para los distintos perfiles de usuario, como organizadores, expositores, patrocinadores y participantes.
- Automatización de tareas clave: Funciones como la creación de eventos, la gestión de entradas, la administración de encuestas y el envío de notificaciones fueron diseñadas para simplificar el trabajo del equipo organizador, reduciendo la carga operativa.
- Mejora de la experiencia del usuario: El proceso de inscripción fue optimizado para ofrecer una experiencia fluida, clara y segura, beneficiando especialmente a los participantes.
- Interacción efectiva con expositores: Se implementó un módulo de postulación de ponencias que permite a los expositores gestionar sus propuestas y realizar un seguimiento del proceso.
- Fortalecimiento del vínculo con patrocinadores y coorganizadores: Se diseñaron funcionalidades específicas para visibilizar su participación, como el envío de mensajes personalizados y niveles de patrocinio diferenciados.

Además, se contó con el valioso apoyo de miembros de la Corporación Python Chile, quienes brindaron retroalimentación práctica basada en su experiencia, lo cual fue clave para orientar el desarrollo del sistema hacia necesidades reales.

Desde el punto de vista del aprendizaje, el proyecto permitió profundizar en áreas como:

- Modelado de bases de datos con enfoque en usuarios y roles.
- Aplicación de principios de diseño orientado a objetos (especialmente SOLID).

A partir del desarrollo realizado y del análisis de su implementación, se proponen las siguientes recomendaciones para continuar mejorando la solución:

- Integración con redes sociales: Facilitar la difusión de actividades a través de plataformas como LinkedIn, X (antes Twitter) o Instagram puede ampliar el alcance y la participación del público.
- Agregar soporte multilingüe: Esto permitiría internacionalizar los eventos organizados y fomentar la inclusión de una audiencia más diversa.
- Mejoras en seguridad y auditoría: Se recomienda fortalecer los mecanismos de control sobre acciones sensibles como la edición o eliminación de eventos, mediante bitácoras y sistemas de doble verificación.

BIBLIOGRAFÍA

- [1] I. Jacobson, G. Booch, and J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. Madrid, España: Addison-Wesley, 2000.
- [2] R. S. Pressman and B. R. Maxim, *Ingeniería de software: Un enfoque práctico*, 9.^a ed. Ciudad de México, México: McGraw-Hill, 2021.
- [3] E. J. Braude, *Ingeniería de software: Una perspectiva orientada a objetos*. Ciudad de México, México: Alfaomega, 2003.
- [4] I. Sommerville, *Ingeniería de software*, 9.^a ed. Ciudad de México, México: Pearson, 2011.
- [5] C. Larman, *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al Proceso Unificado*, 2.^a ed. Madrid, España: Pearson, 2003.
- [6] D. Thomas and A. Hunt, *El programador pragmático: Viaje a la maestría*. 2.^a ed. Madrid, España: Anaya Multimedia, 2022.
- [7] A. Cockburn, *Writing Effective Use Cases*. Traducción y resumen por G. Maggi y M. Palmitesta, UTN – Facultad Regional Rosario, Argentina, 2002.
- [8] S. Adolph and P. Bramble, *Patterns for Effective Use Cases*. Traducción y resumen versión 1.01 por G. Maggi y M. Palmitesta, UTN – Facultad Regional Rosario, Argentina, 2003.
- [9] R. S. Pressman and B. R. Maxim, *Ingeniería de software: Un enfoque práctico*, 7.^a ed. Ciudad de México, México: McGraw-Hill, 2010.
- [10] J. Schmuller, *Aprendiendo UML en 24 horas*. Madrid, España: Prentice Hall, 2002.