

APLICACIÓN WEB PARA LA AUTOMATIZACIÓN DEL PROCESO DE EVALUACIÓN EN EVENTOS DE BAILE

Autor: Daniela Escobar Sánchez

Trabajo de grado para optar al título de Ingeniera en Desarrollo de Software

Asesores:

Cristina López Ramírez

Juan Alejandro López Carmona

Institución Universitaria Pascual Bravo

Facultad de Ingeniería, Ingeniería en Desarrollo de Software

Medellín

2025

Agradecimientos

En primer lugar, agradezco profundamente a mi familia por su apoyo incondicional durante este proceso académico, en el que cada palabra de aliento y compañía fueron fundamentales para superar los retos que implicó la culminación de mi formación profesional.

De manera especial, deseo expresar mi gratitud a mi hija, quien, a pesar de su corta edad, me ha acompañado con amor y paciencia en este camino. A sus once años ha sido mi mayor motivación, compartiendo conmigo los logros obtenidos desde la tecnología hasta la ingeniería, así como los sacrificios y esfuerzos que este proceso demandó. Este triunfo no es únicamente mío, sino también suyo, pues su presencia y comprensión me recordaron en cada momento la importancia de alcanzar esta meta.

Resumen

El presente trabajo tuvo como propósito diseñar y desarrollar un aplicativo web para la gestión y calificación de concursos de baile, con el fin de mejorar la transparencia, la trazabilidad y la eficiencia en los procesos de evaluación utilizados en este tipo de eventos. La investigación partió del análisis del contexto cultural de Medellín y de la identificación de problemas frecuentes en los certámenes, tales como la ausencia de herramientas especializadas, la consolidación manual de puntajes, los errores derivados del uso de planillas y la influencia de sesgos durante el juzgamiento. A partir de este diagnóstico, se definieron los requisitos funcionales del sistema y se construyó un prototipo basado en el patrón Modelo–Vista–Controlador, implementado con PHP, MySQL, HTML5, CSS3, JavaScript y Bootstrap.

El desarrollo incluyó la estructuración del modelo entidad–relación, la creación de módulos para la gestión de eventos, participantes, jueces, academias y criterios de evaluación, y la implementación de mecanismos de seguridad orientados al control de accesos y trazabilidad. Posteriormente, se realizaron pruebas funcionales, de integración y de usabilidad con usuarios de diferentes perfiles, quienes evaluaron el aplicativo desde la perspectiva de los roles asignados. Los resultados evidenciaron que la plataforma ofreció tiempos de respuesta adecuados, navegación intuitiva y una clara identificación de acciones, además de permitir la ejecución correcta de procesos como inscripción, calificación, consulta de historial y generación de reportes.

Como resultado articulado del proceso, la evidencia recopilada permitió concluir que el prototipo cumplió con los criterios de funcionalidad, claridad y simplicidad esperados en un Producto Mínimo Viable, constituyéndose en una herramienta pertinente para apoyar la gestión de concursos de baile y fortalecer la transparencia del proceso evaluativo en estos escenarios culturales.

Abstract

The purpose of this project was to design and develop a web application for the management and scoring of dance competitions, aiming to improve transparency, traceability, and efficiency in evaluation processes commonly used in these events. The study began with an analysis of the cultural context of Medellín and the identification of recurring issues in dance contests, such as the lack of specialized tools, manual score consolidation, errors associated with spreadsheets, and the influence of biases during judging. Based on this diagnosis, functional requirements were defined and a prototype was built under the Model–View–Controller architecture, implemented with PHP, MySQL, HTML5, CSS3, JavaScript, and Bootstrap.

The development process included the construction of the entity–relationship model, the creation of modules for managing events, participants, judges, academies, and evaluation criteria, as well as the incorporation of security mechanisms to ensure controlled access and data traceability. Functional, integration, and usability tests were conducted with users from different profiles, who interacted with the system according to their assigned roles. The results showed adequate response times, intuitive navigation, clear identification of actions, and correct execution of essential processes such as event registration, scoring, consultation of historical records, and report generation.

As a comprehensive outcome of this process, the evidence collected demonstrated that the prototype met the criteria of functionality, clarity, and simplicity expected from a Minimum Viable Product. Consequently, the system proved to be a suitable tool to support the management of dance competitions and to strengthen transparency within evaluation procedures in these cultural contexts.

Contenido

Agradecimientos	2
Resumen	3
Abstract	4
Tabla de figuras	7
Glosario	9
Introducción	12
Planteamiento del Problema	13
1.1 Descripción	13
1.2 formulación	16
2. Justificación	16
3. Objetivos	18
3.1 Objetivo general	18
3.2 Objetivos específicos	18
4. Marco Teórico	19
4.1 Antecedentes	21
5. Metodología	32
5.1 Tipo de proyecto	32
5.2 Método	32
5.3 Fase 1: Planeación	42
5.3.1 Estudio del contexto.	42
5.3.2 Alcance del proyecto	43
5.3.3 Viabilidad del proyecto	43
5.3.4 Limitación del aplicativo web	45
5.3.5 Enfoque metodológico	45
5.3.6 Herramientas de desarrollo	46
5.4 Fase 2: Análisis	47
5.5 Fase 3: Diseño	48

5.5.1 Requerimientos técnicos y funcionales.	48
5.5.2 Diagramas UML	48
5.5.3 Diseño del diagrama y modelo entidad–relación	53
5.5.4 Diagramas de flujo	65
5.5.5 Arquitectura MVC	74
5.5.6 Población y muestra	78
5.5.7 Mockups y Wireframes.	80
5.6.1 Base de datos y configuración en phpMyAdmin	96
5.7 Fase 5: Desarrollo	97
5.8 Fase 6: Implementación	98
5.9 Fase 7: Pruebas	105
5.10 Fase 8: Entrega y operación	107
6. Resultados	108
7. Conclusiones	110
8. Referencia Bibliográficas	112
9. Anexos	114

Tabla de figuras

Figura 1. Diagrama de causa-efecto sobre problemas en la evaluación de concursos de baile	15
Figura 2. Interfaz vista opciones Evento – Organizador	22
Figura 3. Interfaz vista crear eventos.	22
Figura 4. Interfaz vista listado registro de eventos.	23
Figura 5. Interfaz vista perfil de usuario.	24
Figura 6. Interfaz vista Home page lista y calendario eventos.	25
Figura 7. Interfaz vista opciones Fecha, tipo Evento – Organizador.	26
Figura 8. Diagrama de flujo sistema información de eventos culturales	28
Figura 9. Interfaz dashboard administrador registro de eventos y semilleros.	30
Figura 10. Interfaz dashboard usuario gestión de eventos y semilleros.	30
Figura 11. Interfaz dashboard administrador CRUD y listado de eventos y semillero	31
Figura 12. Caso de uso rol super administrador.	49
Figura 13. Caso de uso rol administrador academia.	50
Figura 14. Caso de uso rol juez.	51
Figura 15. Caso de uso rol participante/juez participante.	52
Figura 16. Diseño inicial diagrama entidad–relación con estructura limitada a una sola academia.	55
Figura 17. Segunda versión del modelo entidad relación con estructura multi academia y normalización del proceso de evaluación.	58
Figura 18. Tercera versión del modelo entidad–relación, con integración de participantes independientes, resolución de roles por evento y estructura final para evaluación y determinación de ganadores.	61
Figura 19. Cuarta versión del modelo entidad–relación, con Incorporación de las tablas de referencia para departamentos, ciudades y especialidades.	64
Figura 20. Diagrama de flujo CRUD jueces.	66
Figura 21. Diagrama de flujo CRUD jueces.	69
Figura 22. Diagrama de flujo CRUD eventos	72
Figura 23. Prototipo modelo MVC (modelo- vista- controlador)	75
Figura 24. Prototipo modelo MVC (modelo- vista- controlador)	76
Figura 25 Diagrama de paquetes y dependencias del aplicativo web	77
Figura 26. Interfaz de inicio de sesión.	81
Figura 27. Formulario de registro de participante desde la interfaz de administrador de academia	81
Figura 28. Formulario de registro de participantes individuales.	83
Figura 29. Interfaz de gestión y listado de participantes.	83
Figura 30. Interfaz de registro de academias	84
Figura 31. Interfaz de gestión y listado de academias.	85
Figura 32. Formulario de registro de eventos.	86
Figura 33. Formulario de registro de criterios de evaluación.	87
Figura 34. Formulario de registro de divisiones para participantes en eventos.	88
Figura 35. Interfaz de gestión, CRUD y listado de eventos.	89
Figura 36. Formulario de registro de jueces.	90
Figura 37. Interfaz de gestión de jueces y calificación de participantes.	91
Figura 38. Interfaz de gestión y listado de resultados.	92

Figura 39. Interfaz de gestión y listado de ganadores por división en eventos.	93
Figura 40. Interfaz de gestión y listado de calificaciones.	94
Figura 41. Mapa de archivos de la aplicación y su separación por capas.	95
Figura 42. Estructura de carpetas del proyecto en arquitectura (MVC).	96
Figura 43. Arquitectura de base de datos en localhost phpMyAdmin	97
Figura 44. Entorno local en servidor Apache con índice de vistas del sistema.	98
Figura 45. Pantalla de inicio del aplicativo web desplegado en el entorno real.	100
Figura 46. Dashboard superadmin desplegado en el entorno real.	100
Figura 47. Formulario CRUD academia desplegado en el entorno real.	101
Figura 48. Pasarela de pagos del aplicativo web desplegado en el entorno real.	103
Figura 49. Ticket de pago generado de un evento inscrito por un participante desplegado en el entorno real.	104
Figura 50. Resultados de pruebas unitarias por rol en el Test Explorer de Visual Studio	106

Glosario

Aplicación Web: Es un software accesible desde un navegador que permite a los usuarios interactuar con funcionalidades específicas desde cualquier dispositivo conectado a internet. En el contexto de este proyecto, la aplicación web permitirá gestionar administradores de academias, registros de participantes y jueces, así como el proceso de calificación y consolidación de resultados en concursos de baile (Pressman, 2010).

Base de datos: Sistema que permite almacenar, organizar y recuperar información de manera eficiente. Para este proyecto se utilizará MySQL, garantizando consultas rápidas y seguras sobre los datos de los concursos (Elmasri & Navathe, 2015).

Bootstrap: Framework CSS que facilita la creación de interfaces responsive y adaptables a distintos tamaños de pantalla. Se emplea para asegurar que jueces, administradores y participantes puedan acceder a la plataforma desde computadoras, tabletas y dispositivos móviles sin perder usabilidad (Marcotte, 2011).

Gestor de base de datos: Herramienta que permite administrar y mantener la información, gestionando tablas, consultas y seguridad de manera más intuitiva. En este proyecto se usará phpMyAdmin, integrado en el entorno XAMPP (Coronel & Morris, 2019).

Modelo-Vista-Controlador (MVC): Patrón de diseño que divide la aplicación en tres capas: Modelo (gestión de datos), Vista (interfaz de usuario) y Controlador (lógica de negocio), lo que facilita la organización del código y mejora la mantenibilidad del sistema. Este patrón guiará la estructura del desarrollo en PHP (Gamma, Helm, Johnson, Vlissides, 1995).

MySQL: Sistema de gestión de bases de datos relacional que permite consultas rápidas y seguras sobre datos estructurados. Será la base de datos principal del proyecto (Elmasri & Navathe, 2015).

PHP: Lenguaje de programación orientado al desarrollo web, utilizado para implementar la lógica de negocio del sistema y gestionar la interacción entre la base de datos y la interfaz de usuario (Welling & Thomson, 2017).

Programación Orientada a Objetos (POO): Modelo de programación que permite representar entidades del mundo real mediante clases y objetos, favoreciendo la reutilización y escalabilidad del código. En el proyecto, entidades como jueces, participantes, eventos y calificaciones se modelarán como objetos (Booch, 2007).

Responsive Design: Estrategia de diseño que asegura que la interfaz se adapte automáticamente a distintos tamaños de pantalla, optimizando la experiencia del usuario. Permitirá que la plataforma sea accesible desde cualquier dispositivo (Marcotte, 2011).

Servicios en la nube: Infraestructura tecnológica que ofrece recursos bajo demanda, escalabilidad y alta disponibilidad. Aunque inicialmente el sistema se desplegará en un hosting compartido, se contempla la migración a la nube si se requiere atender un mayor número de usuarios (Mell & Grance, 2011).

Visual Studio Code: Entorno de desarrollo integrado (IDE) que permite programar, depurar y gestionar proyectos en PHP, HTML, CSS y JavaScript. Será la herramienta principal para escribir y mantener el código del sistema (Microsoft, 2020).

XAMPP: Entorno de desarrollo que integra Apache, MySQL, PHP y otros componentes, utilizado para ejecutar la aplicación localmente, facilitando pruebas y desarrollo antes del despliegue en el servidor final (Apache Friends, 2020).

Usuarios: Personas que interactúan con el sistema, incluyendo administradores, participantes y jueces. Cada tipo de usuario tendrá roles y permisos específicos para garantizar la seguridad y trazabilidad de las acciones dentro de la plataforma (Pressman, 2010).

Roles de usuario: Definición de permisos específicos para cada tipo de usuario, asegurando que solo puedan realizar acciones autorizadas y consultar únicamente la información correspondiente a su perfil (Ferraiolo & Kuhn, 1992).

Criterios de calificación: Conjunto de indicadores establecidos para evaluar el desempeño de los participantes en los concursos de baile, permitiendo objetivar la puntuación y mejorar la consistencia de los resultados (Andrade, 2005; Brookhart, 2013).

Auditoría: Registro completo de las acciones realizadas en la plataforma, incluyendo información sobre el usuario, fecha, hora y tipo de actividad. Garantiza trazabilidad, transparencia y confiabilidad en la gestión de los concursos (International Organization for Standardization, 2013).

Introducción

La cultura y el arte representan pilares fundamentales en la construcción de identidad y cohesión social, y dentro de ellos la danza ocupa un lugar privilegiado como manifestación que integra tradición, innovación y participación ciudadana. En Colombia, este ámbito ha adquirido una relevancia creciente gracias a concursos, certámenes y festivales que no solo fomentan la preservación cultural, sino que también impulsan la economía creativa y generan espacios de encuentro comunitario. Ante este panorama, las tecnologías de la información se presentan como herramientas indispensables para modernizar los procesos de gestión y evaluación, garantizando mayor transparencia, confiabilidad y eficiencia en el desarrollo de los eventos.

En este contexto, el presente proyecto plantea el diseño de una aplicación web orientada a la evaluación en concursos de baile, concebida para centralizar la inscripción de participantes, la asignación de jurados, la administración de cronogramas y la calificación estandarizada de los concursantes. Además de atender una necesidad del sector cultural, la propuesta se constituye en un aporte académico y tecnológico que fortalece la gestión cultural y puede ser replicable en otros contextos artísticos. El documento se estructura en torno al planteamiento del problema, la justificación, los antecedentes investigativos, la metodología adoptada y los resultados esperados, ofreciendo un recorrido integral por el proceso de investigación y desarrollo del sistema.

Planteamiento del Problema

1.1 Descripción

En Colombia, y particularmente en Medellín, la danza es una práctica cultural de alto impacto social y artístico. A lo largo del año se realizan concursos, certámenes y festivales de salsa, tango, ballet, danza urbana y folclor que preservan tradiciones, dinamizan la economía cultural y fomentan la participación ciudadana. El Ministerio de Cultura reportó que en 2021 se registraron 751 eventos de danza, equivalentes al 8 % de los espectáculos de artes escénicas del país, aunque la cifra real probablemente sea mayor porque muchos eventos comunitarios no quedan registrados de manera oficial (Ministerio de Cultura, 2021). En Medellín, además de hitos como el Festival Internacional de Tango 2024, que congregó a miles de artistas y espectadores, se destaca la realización de Medellín Sí Danza 2025 con la participación de más de 800 artistas distribuidos en alrededor de 90 agrupaciones y un total de 35 actividades entre presentaciones, talleres y muestras escénicas, lo que confirma el alcance social y cultural de la danza y evidencia la necesidad de contar con mecanismos tecnológicos especializados para gestionar la evaluación en este tipo de certámenes (El Tiempo, 2024; Medellín.travel, 2025).

A pesar de este crecimiento, persiste un problema en la manera cómo se evalúa a los participantes, pues muchos concursos todavía emplean hojas de cálculo y formularios genéricos para registrar y consolidar los puntajes. La literatura ha mostrado que los errores en planillas son frecuentes y tienden a incrementarse conforme los archivos aumentan en tamaño y complejidad; en escenarios con varios criterios, múltiples rondas y distintos jueces, esta fragilidad técnica se convierte en un obstáculo que afecta la estandarización de los procesos, limita la trazabilidad de la información y disminuye la confianza en los resultados (Panko, 2000; Panko, 2016). La mezcla de entradas manuales, consolidaciones tardías y controles ex post deja vacíos de verificación que el público y los participantes no pueden auditar con facilidad.

De forma complementaria, la imparcialidad del juzgamiento también se ve comprometida. En paneles de jurados, la calificación puede verse afectada por sesgos de comparación cuando los jueces tienen acceso a los puntajes de otros o por presión de grupo cuando anticipan la tendencia general antes de emitir su valoración, lo que reduce la independencia de sus decisiones

(Furnham y Boo, 2011; Boen, Van Hoye y Van Auweele, 2008). Asimismo, distintos trabajos han observado sesgos de afinidad y sesgos nacionalistas, en los que los jueces favorecen a competidores con quienes comparten vínculos institucionales o de pertenencia, como nación, ciudad, club o academia. Zitzewitz (2014) documentó estos patrones en deportes de invierno y mostró que elevan de manera sistemática las notas del propio grupo y distorsionan posiciones finales; su lógica es aplicable a disciplinas artísticas juzgadas, como la danza, donde coexisten escuelas rivales, reputaciones previas y presión de pares. Cuando el sistema de evaluación muestra promedios en tiempo real, identifica públicamente la academia del participante o no controla el orden de presentación, se intensifica la influencia del orden en la percepción del evaluador, lo que agrava la falta de equidad.

En conjunto, la revisión del contexto y la evidencia empírica permite reconocer que la ausencia de herramientas especializadas para la evaluación en concursos de baile genera fallas en la confiabilidad, la imparcialidad y la trazabilidad de los resultados. Como se puede apreciar en la Figura 1, estas dificultades surgen de la interacción entre factores técnicos, operativos y humanos: la falta de estandarización en los procesos, la consolidación dispersa de datos y los sesgos de los jueces comprometen la transparencia y la credibilidad de los certámenes culturales. Este entramado de causas evidencia la necesidad de diseñar una solución tecnológica que promueva la independencia en la evaluación, implemente protocolos de calibración con criterios verificables, aplique métodos de normalización y agregación por juez, controle el orden de presentación y asegure un registro auditable.

Por tanto, la problemática identificada constituye la base del problema de investigación y da origen a la pregunta orientadora del proyecto.

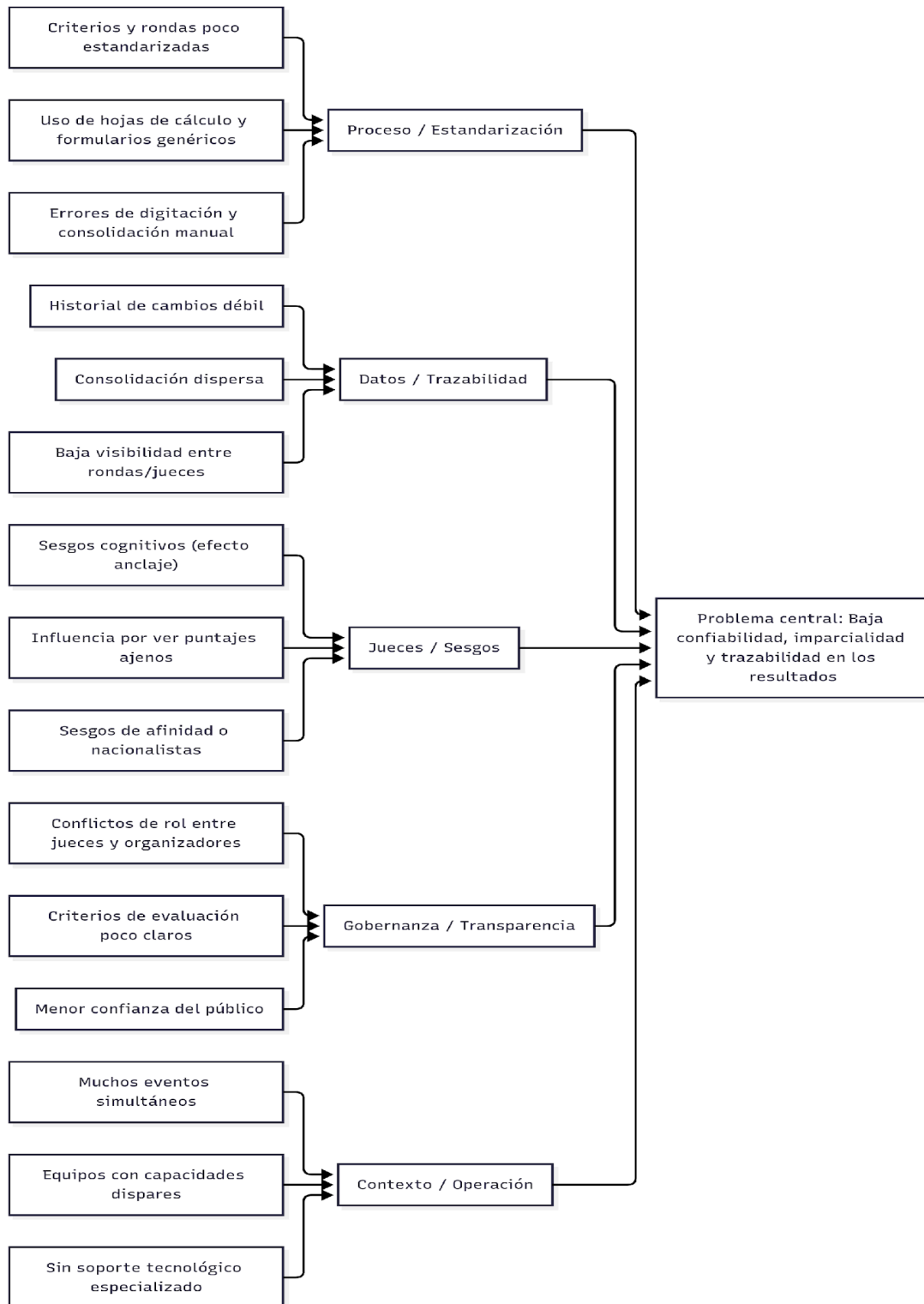


Figura 1. Diagrama de causa-efecto sobre problemas en la evaluación de concursos de baile
 Fuente: Elaboración propia.

1.2 formulación

A partir de este panorama surge la pregunta de investigación: ¿Cómo diseñar e implementar una aplicación web que automatice el proceso de evaluación en concursos de baile, garantizando la imparcialidad de los jueces, la trazabilidad de los resultados y la transparencia en la gestión del evento?

2. Justificación

El diseño de una aplicación web para la evaluación en concursos de baile encuentra su justificación en distintas dimensiones que abarcan lo cultural, lo social, lo tecnológico y lo académico. En primer lugar, la danza constituye una manifestación artística con alto impacto en la identidad cultural de Medellín y de Colombia en general, pues los concursos, festivales y certámenes de este tipo no solo conservan tradiciones, sino que también estimulan la participación ciudadana y fortalecen la economía creativa (Ministerio de Cultura, 2021; Medellín.travel, 2025). De esta manera, disponer de herramientas tecnológicas que garanticen transparencia y eficiencia en los procesos de evaluación aporta al fortalecimiento de la gestión cultural y responde a la necesidad de profesionalizar prácticas que históricamente han carecido de estandarización.

En segundo lugar, la dimensión social del proyecto radica en la importancia de asegurar la imparcialidad en la calificación de los participantes. Estudios previos han mostrado que los sesgos cognitivos y sociales afectan la objetividad de las evaluaciones, lo que puede influir en los resultados y generar desconfianza en los procesos (Furnham & Boo, 2011; Boen, Van Hoye & Van Auweele, 2008; Zitzewitz, 2014). La implementación de un sistema digital que limite la visibilidad de calificaciones ajenas y reserve al administrador la consolidación de resultados

permite reducir estos riesgos, garantizando que las competencias se desarrollen bajo criterios de equidad y credibilidad, lo que fortalece la confianza de artistas, jurados y público en general.

Desde una perspectiva tecnológica, la pertinencia del proyecto se respalda en la evidencia de que los sistemas de información permiten optimizar la gestión de datos, reducir errores humanos y garantizar la trazabilidad de procesos complejos. Según Turban, Pollard y Wood (2018), el uso de herramientas digitales bien estructuradas incrementa la eficiencia administrativa y proporciona información confiable para la toma de decisiones. En este sentido, una aplicación especializada para concursos de baile, que automatice la validación de datos, conserve trazas de auditoría y genere reportes estandarizados, ofrece un soporte técnico acorde con la magnitud de los eventos culturales contemporáneos.

Finalmente, en el plano académico, este proyecto representa un aporte significativo, dado que integra los avances de la ingeniería de software con las necesidades reales de la gestión cultural. La construcción de una plataforma que centralice la inscripción de participantes, la asignación de jurados, la administración de cronogramas y la calificación estandarizada de concursantes responde a una brecha identificada en investigaciones previas, que han priorizado aspectos administrativos, pero no la evaluación directa de los participantes (García, Carvajal & Sanabria, 2025; Universidad EAN, 2022). Así, el sistema no solo contribuye a la gestión eficiente de eventos, sino que también ofrece un modelo replicable en otros contextos culturales, fortaleciendo el ecosistema digital para la cultura en Colombia.

Este conjunto de problemáticas justifica la necesidad de diseñar una solución específica para los concursos de baile. La propuesta consiste en una aplicación web que permita configurar criterios y ponderaciones con flexibilidad, garantice que cada juez registre y consulte únicamente sus propias calificaciones y reserve para el administrador la consolidación y el ordenamiento final de los resultados. Con el fin de responder a la complejidad operativa del sector, la plataforma integrará funciones clave de gestión, como la inscripción de participantes, la asignación de jurados por categoría, la administración de cronogramas, el registro estandarizado de evidencias, la validación automática de datos y la generación de reportes y trazas de auditoría. Asimismo, el sistema incorporará la posibilidad de definir roles y estados diferenciados, de modo que los jueces puedan participar activamente en los certámenes y los concursantes, ya sean independientes o pertenecientes a una o varias academias, cuenten con un estatus claro de

participación en los eventos. De este modo, se protegerá la independencia de criterio al reducir señales que induzcan comparaciones entre jueces, se mejorará la trazabilidad mediante bitácoras completas de cada acción y se incrementará la confiabilidad al disminuir los errores de digitación propios de herramientas genéricas. En conjunto, estas capacidades se alinean con los esfuerzos por fortalecer las prácticas culturales con rigor, transparencia y equidad, ofreciendo un soporte operativo acorde con el volumen y la diversidad de eventos que se registran en el país (Ministerio de Cultura, 2021; DANE, 2020).

3. Objetivos

3.1 Objetivo general

Diseñar e implementar una aplicación web para la automatización del proceso de evaluación en concursos de baile en Medellín durante el año 2025, que permita registrar participantes y jurados, configurar criterios de calificación y consolidar resultados en tiempo real, con el fin de garantizar la transparencia de los puntajes, la trazabilidad de los datos y la eficiencia en la gestión de los eventos.

3.2 Objetivos específicos

Establecer criterios de diseño orientados a garantizar la transparencia, trazabilidad e independencia del juicio, con base en el levantamiento de requisitos funcionales y no funcionales de la aplicación web, a través de la revisión documental y el análisis de casos de uso.

Diseñar la arquitectura de la aplicación bajo el modelo multicapa, definiendo estructura de datos, roles de usuario y módulos de interacción para gestionar eventos, participantes, jurados y criterios de evaluación.

Implementar la aplicación web utilizando tecnologías adecuadas para garantizar integridad de datos, seguridad de la información y mantenibilidad del sistema.

Realizar pruebas funcionales y de usabilidad con escenarios controlados de usuarios, evaluando cumplimiento de requisitos, eficiencia en la captura y consolidación de resultados, y experiencia de uso de la plataforma.

En consecuencia, estos objetivos específicos orientan el desarrollo del aplicativo web hacia una solución integral, confiable y metodológicamente sólida para la gestión de concursos de baile. Su articulación permite que el sistema cuente con una arquitectura clara, una evaluación transparente por parte de los jueces, un manejo estructurado de la información y una experiencia de uso coherente con los requerimientos levantados. Además, garantizan que cada componente, desde la definición de criterios y la organización de datos hasta la implementación técnica y las pruebas funcionales para que contribuya a un producto final capaz de soportar procesos de calificación con independencia, precisión y trazabilidad.

4. Marco Teórico

A lo largo del tiempo, la evaluación en competencias artísticas ha pasado de apoyarse casi exclusivamente en la tradición y en el prestigio de los jurados a estructurarse en torno a reglas y criterios más definidos. Según Huber (2014), este cambio surgió de la necesidad de disminuir la subjetividad en los procesos de calificación y de otorgar mayor legitimidad a los resultados, especialmente en contextos donde las valoraciones dependen de la reputación personal de quienes califican. De manera complementaria, Macnamara (2005) señala que la introducción de escalas y rúbricas detalladas contribuyó a que tanto las competencias artísticas como las deportivas adoptaran mecanismos de evaluación más comprensibles y transparentes. Gracias a estas herramientas, los participantes y el público no solo pudieron comprender mejor los resultados, sino que también se redujo la posibilidad de sesgos derivados de apreciaciones individuales.

En los concursos de baile social del siglo XX se empezaron a establecer procedimientos para unificar las opiniones de varios jueces en un solo resultado. Al mismo tiempo, en disciplinas cercanas se hicieron ajustes importantes. Por ejemplo, el patinaje artístico reemplazó la escala de seis puntos por un código de puntuación con valores base, componentes y deducciones, lo que permitió explicar de manera más precisa cómo se sumaban los puntos y por qué un participante obtenía una posición determinada (International Skating Union, 2004). De manera paralela, en el baile deportivo los manuales de la federación internacional promovieron escalas con estándares técnicos y artísticos, además de reglas de consolidación conocidas por todos, de modo que la conversación se centrara en los criterios y no solo en percepciones individuales (World DanceSport Federation, 2013). Así, la tendencia general ha sido pasar de juicios implícitos a calificaciones con criterios explícitos y trazables, lo cual se conecta con el propósito de este trabajo.

Ahora bien, para que esa evaluación sea consistente entre jueces y rondas, no basta con asignar una calificación. Por un lado, las rúbricas sirven para definir de manera más clara qué atributos se observan en el desempeño y cuál es el esquema de valoración, incluyendo dimensiones, descriptores y niveles de logro que reducen ambigüedades y hacen más confiables las calificaciones (Andrade, 2005; Brookhart, 2013). Por otro lado, cuando la captura de datos y la consolidación dependen de hojas de cálculo genéricas, suelen aparecer errores de fórmula o de transcripción, sobre todo cuando el evento crece en categorías y criterios. Powell, Baker y Lawson (2009) estudiaron este problema y recomendaron controles y registros formales para mantener la integridad de la información. Además, los marcos de calidad de datos destacan la importancia de atributos como exactitud, completitud y trazabilidad durante todo el ciclo de vida de la información, lo cual resulta clave si en algún momento es necesario sustentar decisiones (International Organization for Standardization, 2015).

También influyen factores humanos. La psicología del juicio muestra que las personas, incluso las expertas, pueden verse afectadas por señales del contexto como el anclaje o la conformidad. Si un juez observa puntajes parciales o tendencias de otros jurados, es más probable que las calificaciones se parezcan y se pierda independencia de criterio. Limitar la visibilidad de calificaciones ajenas y postergar la socialización de resultados favorece decisiones más autónomas y consistentes (Mussweiler y Strack, 1999; Asch, 1955; Plessner y Haar, 2006).

En la ingeniería de software esto se traduce en el uso de controles de acceso por roles, que permiten definir permisos específicos para cada perfil, y en los registros de auditoría, que documentan cada acción realizada con información del usuario, la fecha y el entorno, lo que asegura trazabilidad (Ferraiolo y Kuhn, 1992; International Organization for Standardization, 2013). De esta forma, una plataforma web que restringe la visibilidad de los jueces, consolide resultados a cargo del administrador y guarde bitácoras completas se ajusta a las buenas prácticas de evaluación y permite medir aspectos como independencia del juicio, integridad y consistencia de los datos, tiempos de cierre y aceptación de uso (Nielsen, 1993; Davis, 1989).

En conjunto, estos antecedentes muestran que la evaluación en concursos artísticos necesita criterios claros y mecanismos que reduzcan la subjetividad, aunque en la danza no se puede dejar de lado el valor estético y cultural que caracteriza a la práctica. Degli Esposti (2020) señala que las herramientas digitales deben buscar eficiencia sin olvidar la dimensión humana de las actividades que median. En este mismo sentido, el concepto de Sociedad 5.0, propuesto por el Gobierno de Japón, plantea la integración de tecnologías digitales con capacidades sociales y humanas para lograr un uso más crítico y equitativo de los sistemas (Cabinet Office, Government of Japan, 2019). Bajo este enfoque, un sistema de evaluación para concursos de baile no solo debe asegurar transparencia técnica y confiabilidad de los resultados, sino también garantizar que la valoración de lo estético sea justa y equilibrada en relación con el papel de los jurados y organizadores.

4.1 Antecedentes

En Colombia, el desarrollo de sistemas digitales para la gestión de eventos culturales y académicos ha mostrado avances significativos, aunque los proyectos que se enfocan específicamente en la calificación automatizada de participantes son limitados. Un caso relevante es la tesis *Desarrollo de una Aplicación Web para la Organización y Registro de Eventos del Politécnico Gran Colombiano* (2021), cuyo objetivo principal fue crear una aplicación web capaz de centralizar los procesos de organización y registro de eventos institucionales, optimizando la gestión de información y la generación de reportes. Este trabajo permitió consolidar los procesos administrativos de eventos, mejorando la eficiencia institucional frente a la alta frecuencia de

actividades y proporcionando una perspectiva clara sobre la importancia de las herramientas digitales para mejorar la gestión de eventos.

La aplicación *PoliEventos* implementó una interfaz sencilla que permitió crear, visualizar y modificar eventos de forma centralizada. Como se muestra en la Figura 2, el sistema cuenta con un menú principal desde el cual los usuarios pueden acceder a los módulos de creación, consulta y registro de invitados y ubicaciones. Por su parte, la Figura 3 presenta el formulario de creación de eventos, que integra campos para el registro de fechas, responsables y enlaces a encuestas institucionales. Finalmente, la Figura 4 exhibe la lista consolidada de eventos, organizada por identificadores y fechas, lo que facilitó el control interno y la trazabilidad administrativa.

Aunque la aplicación no contempló la evaluación de participantes ni mecanismos de ponderación por criterios, su arquitectura demostró el potencial de las plataformas web para sistematizar la gestión cultural y académica. Este antecedente aporta una referencia útil para el desarrollo del presente proyecto, al evidenciar cómo los procesos de registro, seguimiento y consolidación pueden integrarse en un entorno digital adaptable a la evaluación de concursos de baile, incorporando además aspectos de imparcialidad, trazabilidad y transparencia.



Figura 2. Interfaz vista opciones Evento – Organizador
Fuente: Politécnico Grancolombiano (2021).

PoliEventos

Nombre
Nombre del Evento

Fecha de Inicio
Fecha de Inicio

Fecha de Fin
Fecha de Fin

Fue Realizado el Evento?
Fue Realizado el Evento?(true/false)

Motivo de No Realizacion
Por qué no se realizó? (dejar vacío en caso contrario)

Encuesta del Evento
URL a la Encuesta en Microsoft Forms

Resultados de la Encuesta
Resultados de la Encuesta en Microsoft Forms

ID Organo Institucional
ID Organo Institucional

ID Escuela
ID Escuela (dejar vacío si no hay)

ID Programa
ID Programa (dejar vacío si no ha)

ID Organizador
ID Organizador

Crear Evento Volver



Figura 3. Interfaz vista crear eventos.
Fuente: Politécnico Gracolombiano (2021).

PoliEventos

Lista de Eventos

ID	Nombre	Fecha de Inicio	Fecha de Fin	idPrograma	idEscuela	idOrganoInstitucional	idOrganizadorEvento
15	Maratones de Programación ACM 2021-1	2021-08-30T08:00:00.000Z	2021-08-31T16:00:00.000Z	5	5	5	5
25	Feria de Bienestar Universitario 2021-2	2021-08-09T08:00:00.000Z	2021-08-13T16:00:00.000Z			15	15
35	POLI SPEAKERS Alexander López Noviembre 2021	2021-10-01T10:00:00.000Z	2021-10-01T12:00:00.000Z	55	25	5	25

Volver



Figura 4. Interfaz vista listado registro de eventos.
Fuente: Politécnico Gracolombiano (2021). Repositorio Politécnico Gracolombiano-<https://repositorio.policia.edu.co/handle/10901/25235>

De manera complementaria, la tesis *Desarrollo de una base de datos para la gestión de eventos de la Universidad de Cundinamarca, Facultad de Ingeniería* (García, Carvajal y Sanabria, 2025) presenta el desarrollo del sistema web EventPlace, diseñado para optimizar la gestión administrativa de eventos culturales y académicos en la Universidad de Cundinamarca, extensión Soacha. Este sistema reemplaza el uso de formularios de Google y matrices de Excel, los cuales,

aunque funcionales, resultan poco eficientes y desorganizados. EventPlace centraliza la información, permite acceder a datos de manera estructurada y mejora la medición y control de eventos, asegurando que la información esté disponible de manera confiable para la planificación y evaluación de las actividades.

Como se observa en la Figura 5, el sistema integra un perfil de usuario donde los participantes pueden consultar suscripciones, eventos próximos o pasados e intereses asociados, lo que favorece la personalización de la experiencia y el seguimiento individual. La Figura 6 presenta el calendario general de eventos, que organiza las actividades por meses y facilita la navegación cronológica, permitiendo identificar de manera visual los eventos activos. Finalmente, la Figura 7 muestra el módulo de eventos disponibles, en el cual los usuarios pueden explorar actividades académicas, culturales, deportivas, artísticas y tecnológicas, suscribirse o calificarlas una vez finalizadas.

Aunque el desarrollo no se enfoca en la calificación directa de los participantes, su estructura modular y su orientación hacia la trazabilidad y la gestión de datos constituyen un referente sólido para la presente propuesta. En conjunto, este antecedente reafirma la importancia de los sistemas web como soporte de la gestión cultural universitaria y demuestra la viabilidad de adaptar sus principios para construir soluciones orientadas a la evaluación estructurada de concursos de baile, con mayor imparcialidad, trazabilidad y control de información.



Figura 5. Interfaz vista perfil de usuario.

Fuente: García, Carvajal & Sanabria (2025), Universidad de Cundinamarca.

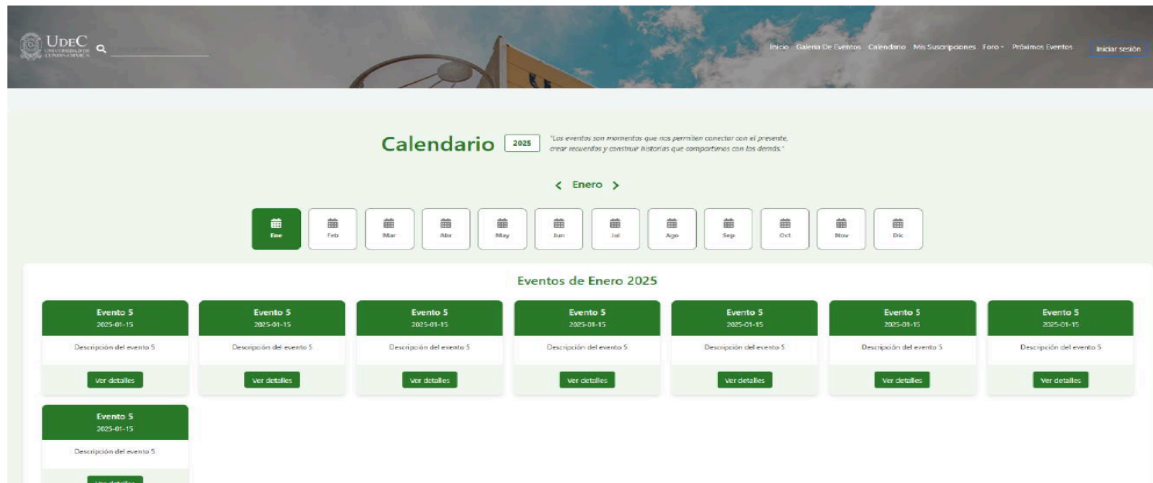


Figura 6. Interfaz vista Home page lista y calendario eventos.

Fuente: García, Carvajal & Sanabria (2025), Universidad de Cundinamarca.

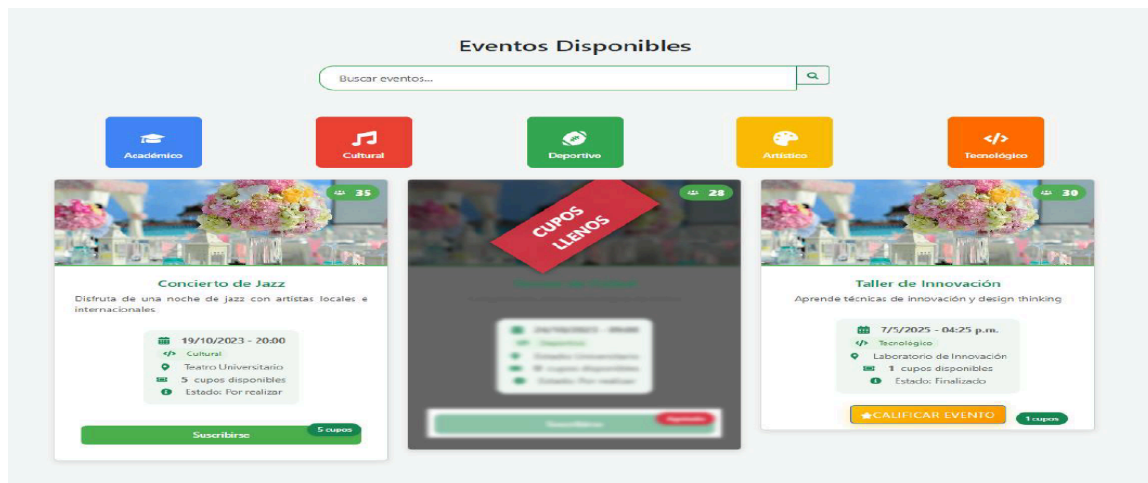


Figura 7. Interfaz vista opciones Fecha, tipo Evento – Organizador.

Fuente: García, Carvajal & Sanabria (2025), Universidad de Cundinamarca. Repositorio Universidad de Cundinamarca: <https://repositorio.ucundinamarca.edu.co/handle/20.500.12558/49>

Adicionalmente, la tesis *Propuesta de un Sistema de Información Unificado para Eventos Culturales en Medellín, Antioquia* de la Universidad EAN (s.f.) se centra en la creación de un sistema que integra los requerimientos funcionales de proveedores, organizadores y participantes de actividades culturales, bajo un modelo multi actor orientado a la planificación y difusión de

eventos. Este proyecto busca consolidar la información de los diferentes actores culturales en una sola plataforma, permitiendo la consulta, registro y organización de eventos de manera unificada y colaborativa.

Como se muestra en la Figura 8, la propuesta presenta un diagrama de flujo que detalla los procesos principales del sistema y las interacciones entre los módulos de ciudadanos, eventos y archivos de registro. Dicho esquema refleja un diseño orientado a la trazabilidad de datos y al filtrado de información según intereses y afinidades, lo que favorece la personalización de la oferta cultural y la optimización de los recursos institucionales. Aunque el trabajo no desarrolló una interfaz funcional, su representación de procesos evidencia una planificación integral del sistema, con potencial de implementación para la gestión cultural local.

Este antecedente constituye un aporte valioso para el presente proyecto, ya que plantea la necesidad de contar con plataformas que integren no solo la organización y registro de actividades culturales, sino también la posibilidad de evaluar y retroalimentar las participaciones de manera estructurada, objetiva y trazable, fortaleciendo así la transparencia y la confiabilidad de los procesos culturales y artísticos.

Propuesta de un Sistema de Información Unificado Para Eventos Culturales en Medellín, Antioquia

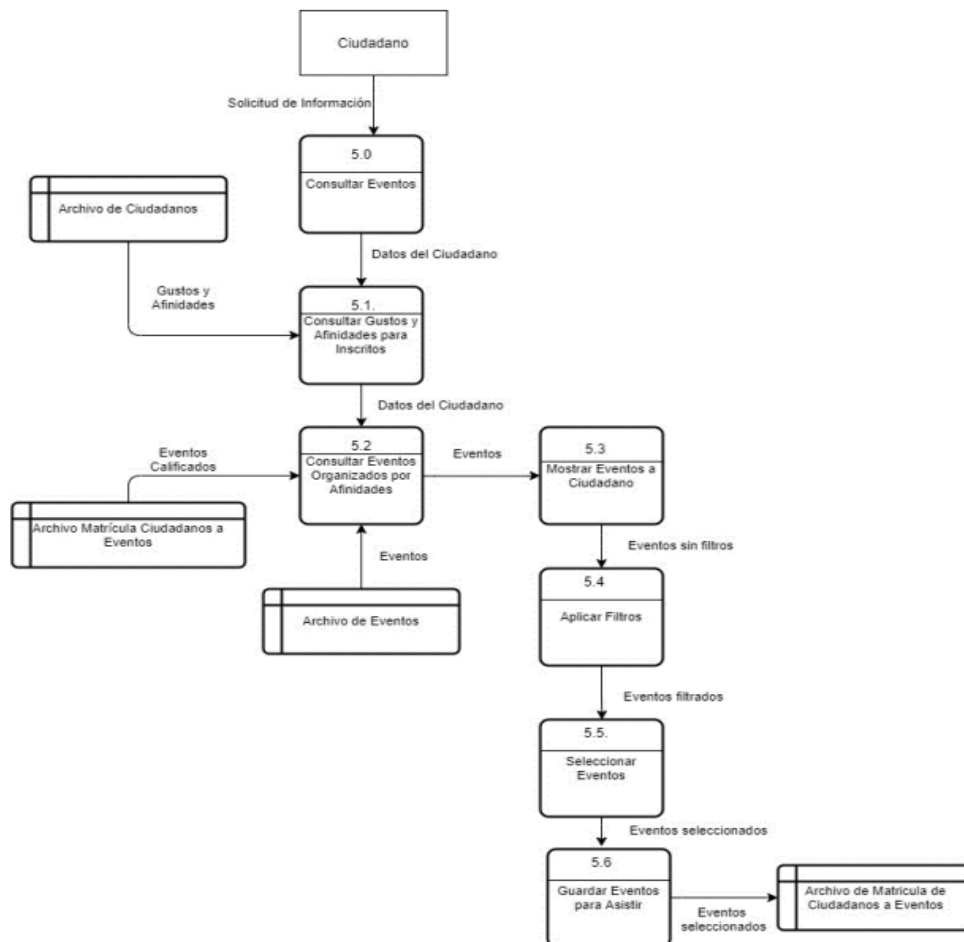


Figura 8. Diagrama de flujo sistema información de eventos culturales

Fuente: Universidad EAN (2022). Repositorio Universidad EAN:

<https://repository.universidadean.edu.co/server/api/core/bitstreams/c8ed63a1-6c59-4da8-84b1-1032b74777be/content?>

Por otro lado, la tesis *Diseño de un prototipo de aplicativo web para la gestión de semilleros de investigación de la Universidad Nacional Abierta y a Distancia* (Carvajal, 2019) desarrolló un sistema denominado **SIGESI**, orientado a facilitar la gestión de semilleros de investigación mediante la digitalización de sus procesos administrativos y académicos. Este prototipo incorporó módulos para el registro de integrantes, la administración de proyectos y la gestión de

eventos, lo que permitió a los líderes de semilleros realizar un seguimiento más detallado de las actividades y de los avances de cada grupo.

Como se muestra en la Figura 9, el sistema presenta un panel de información general en el que se pueden visualizar datos clave del semillero, como su estado, el número de integrantes, los proyectos activos y las solicitudes en curso. De esta manera, la interfaz funciona como un tablero de control que ofrece una visión rápida del estado actual de cada grupo y de sus actividades. Además, la Figura 10 muestra el panel principal del sistema, donde se incluyen estadísticas sobre la cantidad de integrantes y proyectos activos, junto con gráficos que permiten analizar la evolución del semillero a lo largo del tiempo. Gracias a este tipo de visualización, los líderes pueden tomar decisiones con mayor fundamento y planificar de manera más organizada las futuras actividades. Finalmente, la Figura 11 presenta la lista de usuarios registrados en el sistema, con sus nombres, roles y estados, lo que permite mantener un control actualizado sobre los miembros activos y diferenciar entre líderes, administradores e investigadores.

En conclusión, aunque el propósito del SIGESI no fue la calificación de participantes, el diseño de su módulo de eventos y su sistema de control de estado, que clasifica las actividades como activas o inactivas, aportan elementos valiosos para el desarrollo del presente proyecto. Además, demuestra cómo una plataforma web puede mejorar la organización, la trazabilidad y el seguimiento de actividades académicas y culturales, constituyéndose en una referencia importante para la creación de un sistema que también integre procesos de evaluación estructurada, transparente y confiable en el contexto de los concursos de baile.



Figura 9. Interfaz dashboard administrador registro de eventos y semilleros.
Fuente: Carvajal (2019), Universidad Nacional Abierta y a Distancia.



Figura 10. Interfaz dashboard usuario gestión de eventos y semilleros.
Fuente: Carvajal (2019), Universidad Nacional Abierta y a Distancia.

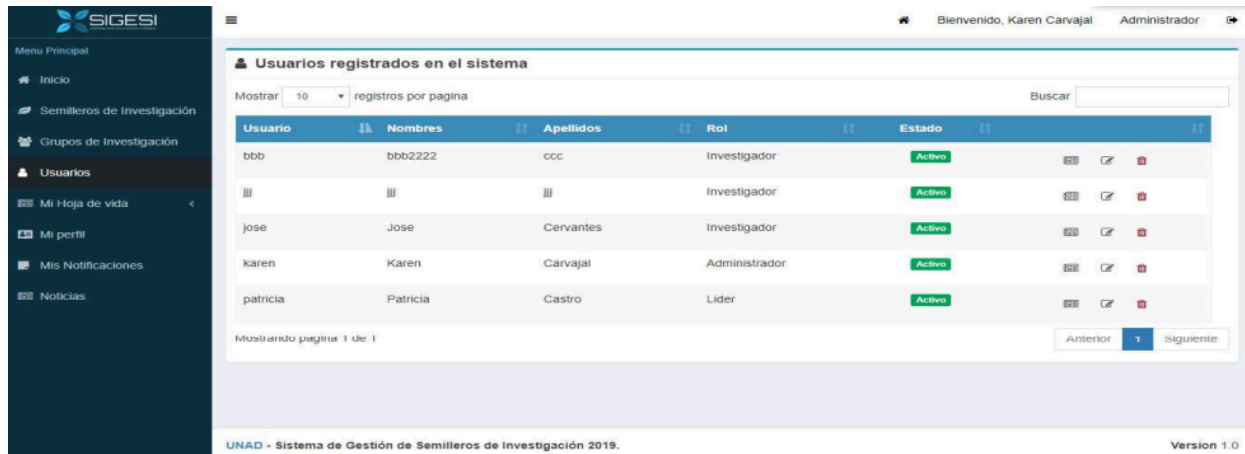


Figura 11. Interfaz dashboard administrador CRUD y listado de eventos y semillero
Fuente: Carvajal (2019), Universidad Nacional Abierta y a Distancia. Repositorio UNAD:
<https://repository.unad.edu.co/bitstream/handle/10596/27486/%20kpcarvajalb.pdf?sequence=1>

A partir del análisis comparativo de estos sistemas, se identifican avances significativos en la digitalización de procesos culturales y en la gestión de eventos artísticos; sin embargo, muestran limitaciones que generan oportunidades de mejora. Entre ellas se destacan la priorización de aspectos administrativos sobre la evaluación de los participantes, la falta de mecanismos claros para garantizar la imparcialidad y reducir los sesgos de los jurados, la limitada trazabilidad de los resultados y, en algunos casos, la ausencia de cumplimiento integral del marco legal vigente en cuanto a protección de datos y propiedad intelectual. Estas falencias evidencian la necesidad de una plataforma que centralice la evaluación como elemento principal, automatice la calificación de participantes, incorpore criterios adaptables de evaluación, ofrezca visibilidad controlada de puntajes y reporte transparente de resultados, y cumpla con las normativas legales colombianas vigentes. Además, el sistema debe permitir que cada usuario tenga un estado definido, de modo que los jueces puedan participar en los eventos asignados y los participantes puedan inscribirse en uno o varios eventos, o no pertenecer a ninguna academia, según corresponda.

Por otro lado, la administración de los roles y el seguimiento del estado de cada usuario y de cada evento a través de tablas estructuradas se convierte en un elemento fundamental para garantizar la trazabilidad, el control y la organización eficiente de los certámenes culturales. En consecuencia, la propuesta de este trabajo de grado se enfoca en diseñar y desarrollar un sistema web que no solo optimice la organización y gestión de eventos culturales y académicos, sino que también permita la administración clara de roles, la participación controlada de usuarios en los

distintos eventos y la retroalimentación efectiva de los resultados, asegurando equidad, trazabilidad y confiabilidad en todo el proceso.

5. Metodología

5.1 Tipo de proyecto

Este proyecto corresponde a una investigación aplicada y exploratoria. La investigación aplicada se orienta a resolver problemas concretos mediante la utilización de conocimientos teóricos y prácticos, con el fin de generar una solución útil en contextos reales (Salinas, 2012). En este caso, se busca desarrollar una aplicación web que automatice el proceso de evaluación en concursos de baile, atendiendo una necesidad específica de este tipo de eventos y validando su efectividad mediante pruebas funcionales y de usabilidad.

Por otra parte, la investigación exploratoria permite indagar sobre prácticas actuales y soluciones tecnológicas existentes en el ámbito de la evaluación de competencias artísticas. Esta combinación metodológica no solo posibilita la implementación de un sistema, sino también la comprensión del contexto del problema y de los actores involucrados, incluyendo jueces, participantes y organizadores, asegurando que el software propuesto responda de manera pertinente a la realidad del sector.

5.2 Método

El desarrollo del proyecto se basa en un enfoque incremental y funcional, centrado en el Producto Mínimo Viable (PMV). Este enfoque permite construir el sistema de manera progresiva, implementando primero las funcionalidades esenciales y agregando mejoras o módulos adicionales conforme se avanza en el desarrollo y se obtienen retroalimentaciones.

El PMV definido para esta investigación incluye:

- Registro a eventos y academias.

- Calificación de participantes según categoría y división.
- Validación de formularios de registro para garantizar integridad de los datos.
- Gestión de roles de usuario: superadministrador, administrador de academia, jueces y participantes.
- Implementación de pasarela de pagos para inscripción a academias y eventos.

El análisis de la información combinó técnicas cuantitativas, para evaluar la frecuencia y tendencias en las respuestas, y técnicas cualitativas, especialmente en la interpretación de entrevistas, con el objetivo de identificar patrones, problemas recurrentes y oportunidades de mejora que orienten el desarrollo del sistema.

El desarrollo se diseñó siguiendo etapas secuenciales del PMV, que incluyen planeación, análisis de requerimientos, diseño de la arquitectura, implementación de funcionalidades esenciales, pruebas de validación y ajustes finales. Este enfoque simple permite gestionar de manera efectiva el proyecto de manera individual, priorizando la entrega de un producto funcional y operativo, garantizando que cada módulo se construya con calidad y pueda ser evaluado de forma independiente antes de su integración final.

Semana 1 (12/08 – 16/08)

Objetivo: Definir PMV y estructura inicial de base de datos.

Tareas:

- Reunión inicial con asesores para establecer el PMV.
- Analizar requerimientos funcionales y no funcionales mínimos.
- Crear modelo entidad-relación de la base de datos.

- Definir tablas y relaciones principales (usuarios, eventos, jueces, academias, divisiones, criterios, pagos) sin normalización completa

Entregable: Documento con PMV y modelo ER inicial.

Semana 2 (12/08 – 17/09)

Objetivo: Construcción de vista login y creación de la base de datos

Tareas:

- Crear las tablas, atributos y relaciones de la base de datos en lenguaje sql
- Generar backup del script
- Conectar el controlador y el modal del formulario de login con la tabla users para el almacenamiento de los datos y validación.

Entregable:

- Primer entregable del archivo bd.sql para almacenar la información
- Primera vista de login con aceptación de credenciales y redireccionamiento al boceto inicial de una vista dashboard en crudo visible en el localhost.

Semana 3 (18/08 – 23/09)

Objetivo: Construcción de vistas iniciales y dashboard módulo super administrador.

Tareas:

- Crear vista de registro de academias
- Crear vista de registro de participantes academia y participantes independientes.
- Crear vista de registro de jueces.
- Crear vistas de registro de eventos.
- Crear vistas de registro de divisiones y criterios

- Creación secciones del sidebar
- Creación botones de acción y logout de la sesión.

Entregable: Diseño front-end de la vista dashboard del módulo superadministrador terminado

Semana 3 (25/08 –30 /08)

Objetivo: Continuación implementación de vistas modulo super administrador y construcción vistas iniciales administrador academia

Tareas:

- Crear vista de registro de academias
- Crear vista de registro de participantes academia y participantes independientes.
- Crear vista de registro de jueces.
- Crear vistas de registro de eventos.
- Crear vistas de registro de divisiones y criterios
- Creación secciones del sidebar
- Crear botones para acciones y logout de la sesión.

Entregable:

Diseño front-end de la vista dashboard del módulo administrador academia terminada.

Semana 4 (01/09 – 05/09)

Objetivo: Construcción de vistas iniciales y dashboard módulo jueces

Tareas:

- Crear vista de grid listado de eventos asignados
- Crear vista de registro de calificaciones
- Crear vista de historial de calificaciones y reportes
- Crear vistas de filtro eventos y calificaciones.
- Creación secciones del sidebar
- Creación botones de acción y logout de la sesión.

Entregable:

Diseño front-end de la vista dashboard del módulo juez terminado

Semana 5 (08/09 – 13/09)

Objetivo: Construcción de vistas iniciales y dashboard módulo participante academia e independiente

Tareas:

- Crear vista de grid listado de eventos disponibles para inscripción
- Crear vista de registro de calificaciones
- Crear vista de historial de calificaciones y eventos celebrados
- Crear vistas listado eventos pendientes
- Creación secciones del sidebar
- Creación botones de acción y logout de la sesión.

Entregable:

Diseño front-end de la vista dashboard del módulo participante academia y participante independiente terminado

Semana 6 (15/09 – 20/09)

Objetivo: Implementación de CRUD para roles y sus módulos

Tareas:

- Establecer reglas de negocio y triggers en modales, controladores, vistas y en el archivo bd.sql donde valide las acciones de los campos de los formularios y la acción de los botones.
- Implementar lógica de asociación y parámetros entre tablas con otras
- Implementar validación y restricción de campos de texto, checkbox, combobox, grids, buttons que hay en los formularios.
- Implementación de estandarización tipográfica (UTF-8) en los formularios, modales de detalles y grids.

Entregable:

- primer prototipo funcional de formularios de registro y edición con validaciones básicas visible en el localhost.
- modelo ER actualizado con las tablas correspondientes y sus relaciones entre ellas.

Semana 7 (22/09 – 27/09)

Objetivo: Diseñar y aplicar arquitectura MVC para los módulos.

Tareas:

- Implementar controladores y modelos PHP para las vistas de los módulos en sus roles (super administrador, administrador academia, jueces, participantes) y sus secciones (divisiones, criterios, eventos)
- Integrar validaciones en backend y frontend.

Entregable: Módulos funcionales en backend y frontend con arquitectura MVC aplicada.

Semana 8 (28/09 – 16/10)

Objetivo: Maquetación en Bootstrap con personalización de estilos (CSS) predefinidos y cambio de lógica vista de secciones en el módulo super admin

Tareas:

- Implementación de plantillas en todas las vistas, modales, formularios, ventanas emergentes y botones.
- Construir una vista interna en cada vista de cada sección
- Corregir y crear vista aparte de registro y modificación de participantes por academia
- Corregir y crear vista aparte de registro y modificación de participantes independientes por academia
- Crear vista aparte de registro y modificación de jueces por academia
- Corregir y crear vistas aparte de registro y modificación de eventos por academia
- Corregir y crear vistas aparte de registro y modificación de divisiones y criterios por academia
- Creación de vista reportes estadísticos globales por academia

Entregable: Módulo super administrador completo.

Semana 9 (22/09 – 27/09)

Objetivo: Normalización de la base de datos

Tareas:

- Asociar las tablas aplicando normalización con tablas que permiten la relación sin redundancia y duplicidad de datos.
- Creación tabla juez-participante para la participación de eventos
- Creación tabla registro de pago para el registro del pago que hace el participante con evento

- Validar trazabilidad de los registros en la base de datos en ejecución en phpmyadmin.

Semana 6 (17/10 – 21/10)

Objetivo: Integración de pasarela de pagos y pruebas iniciales.

Tareas:

- Implementar pago de inscripción a academias y eventos.
- Probar funcionalidades principales (registro, calificación, asignación de roles).
- Ajustar errores de flujo y validaciones detectadas.

Entregable: Plataforma con pagos funcionales y pruebas iniciales completas.

Semana 7 (22/10 – 25/10)

Objetivo: Optimización y preparación para pruebas finales.

Tareas:

- Integrar módulos de frontend y backend.
- Revisar consistencia de datos y relaciones de base de datos.
- Ajustar interfaz y accesibilidad según retroalimentación.

- Migrar información almacenada dentro de controladores dentro del código al archivo sql en phpmysql por medio de consultas.
- Realizar ajuste de campos de texto, combobox y checklist de formularios debido a la modificación y migración de la información.
- Validar y verificar las restricciones de campos de datos de formularios de todos los módulos

Entregable: Plataforma integrada lista para pruebas funcionales.

Semana 8 (26/10 – 14/11)

Objetivo: Pruebas funcionales y corrección de errores.

Tareas:

- Realizar pruebas unitarias y de integración de módulos.
- Corregir errores encontrados y documentar cambios.
- Validar el funcionamiento adecuado de todo el software por medio de las pruebas de usuario.
- Revisar trazabilidad de datos y seguridad básica.

Entregable: Informe de pruebas y plataforma corregida.

Semana 9 (15/11 – 20/11)

Objetivo: Preparación para entrega final y documentación.

Tareas:

- Redacción de manual de usuario y documentación técnica.

- Validación final de formularios, calificaciones y reportes.
- Ajustes finales en diseño de la plantilla y experiencia de usuario.

Entregable: Documentación lista y plataforma estable.

Semana 10 (21/11 –24 /11)

Objetivo: Implementación final y entrega de PMV completo.

Tareas:

- Configuración e instalación del entorno de desarrollo local en otro equipo, instalando XAMPP para ejecutar el servidor Apache y el motor de base de datos MySQL.
- Implementación e importación del script de la base de datos para el despliegue del aplicativo web en servidor APACHE localhost, garantizando su correcto funcionamiento en el servidor local.
- Se verificaron las funcionalidades principales junto con los asesores, comprobando el rendimiento y la estabilidad de los módulos implementados.

Entregable: PMV entregado, funcional y documentado

Semana 11 (07/11 – 13/11)

Objetivo: Pruebas de usuario y retroalimentación final.

Tareas:

- Realizar pruebas con roles creados en el aplicativo web (participantes, jueces y academias).
- Recolectar retroalimentación sobre usabilidad y experiencia.

- Ajustar errores menores y optimizar tiempos de respuesta.

Entregable: Informe de pruebas de usuario y plataforma ajustada.

Semana 12 (14/11 – 17/11)

Objetivo: Cierre del proyecto y entrega definitiva.

Tareas:

- Preparar presentación final del PMV para asesores y evaluadores.
- Revisar documentación técnica y manual de usuario completos.

Entregable: PMV final entregado y respaldado, con documentación completa.

5.3 Fase 1: Planeación

5.3.1 Estudio del contexto

Se realizó una investigación preliminar sobre el proceso actual de evaluación en concursos de baile desarrollados en academias y espacios culturales de Medellín, con el propósito de identificar limitaciones y necesidades del sector. A partir de un rastreo documental de reglamentos y experiencias de concursos locales, se evidenció que en la mayoría de las academias la calificación de los participantes aún se gestiona de manera manual, ya sea en planillas físicas o mediante hojas de cálculo básicas (Alcaldía de Medellín, 2023). Este procedimiento, aunque funcional en términos operativos, presenta desventajas significativas, entre ellas la subjetividad en la aplicación de criterios, la lentitud en la tabulación de puntajes y la propensión a errores humanos en el registro de resultados. Investigaciones en el ámbito académico y cultural han señalado que la evaluación de competencias artísticas suele estar marcada por sesgos relacionados con la percepción individual de los jueces y la falta de estandarización en los criterios (Camacho, 2021; Buckroyd, 2000). Asimismo, el análisis documental permitió identificar que, en diversos eventos, los puntajes presentaban inconsistencias debido a fallas en la consolidación manual de la información, lo que generaba

inconformidad entre organizadores y participantes. Estas problemáticas justifican la necesidad de diseñar un aplicativo que permita automatizar la gestión de calificaciones, de manera que se garantice mayor transparencia, agilidad y confiabilidad en los procesos de evaluación.

5.3.2 Alcance del proyecto

El alcance del proyecto se centra en el desarrollo de un aplicativo web para la gestión de concursos de baile, el cual estará estructurado bajo la arquitectura Modelo-Vista-Controlador (MVC) utilizando PHP, con el fin de asegurar una separación clara entre datos, lógica de negocio y presentación. En este marco, los modelos gestionan la información, los controladores administrarán las reglas de negocio y las vistas permiten la interacción con los usuarios, garantizando así un sistema organizado y eficiente. La plataforma incluirá las funcionalidades definidas en el Producto Mínimo Viable (PMV), tales como el registro de academias, participantes y eventos, la validación de formularios, la asignación de roles diferenciados para super administrador, administradores de academia, jueces y participantes, además de un módulo de calificación por categorías y divisiones con generación automática de reportes consolidados en tiempo real. Asimismo, se implementará una pasarela de pagos destinada a la inscripción en academias y eventos, de manera que se optimice la gestión integral de los concursos. Por lo tanto, no se consideran en esta fase funcionalidades adicionales como análisis estadísticos avanzados o integraciones externas, ya que estas exceden el alcance inicialmente planteado y podrán contemplarse en futuras versiones del sistema.

5.3.3 Viabilidad del proyecto

La factibilidad del proyecto fue analizada desde tres dimensiones fundamentales: técnica, económica y operativa, con el propósito de determinar si el desarrollo del aplicativo web podía ejecutarse de manera realista dentro de las condiciones académicas, temporales y tecnológicas disponibles. En la dimensión técnica, se consideró viable su implementación debido a que el sistema se desarrolló con herramientas de software libre como XAMPP, Visual Studio Code,

PHP, MySQL, HTML, CSS y JavaScript, todas accesibles sin necesidad de licencias comerciales. Esto permitió que el aplicativo pudiera configurarse y evaluarse desde cualquier equipo de cómputo estándar, sin depender de infraestructura especializada ni de capacidades avanzadas de servidor.

Desde la perspectiva económica, la propuesta resultó particularmente favorable, ya que el proceso de validación no requirió la compra de hosting ni dominio. En su lugar, el aplicativo fue evaluado de manera presencial por los usuarios y por los docentes asesores, quienes accedieron al sistema directamente desde el entorno local preparado para tal fin. Esta decisión evitó costos adicionales y se ajustó a las limitaciones propias de un proyecto académico elaborado por una sola estudiante. De igual manera, las herramientas utilizadas no generaron gastos de adquisición, lo que reforzó la sostenibilidad económica del desarrollo.

En términos operativos, la viabilidad se fortaleció gracias a que las pruebas del sistema fueron realizadas de forma presencial con un grupo etario previamente seleccionado, lo que permitió asegurar que la muestra fuera idónea y suficientemente diversa para obtener una retroalimentación amplia y detallada. La interacción directa con los participantes facilitó observar el comportamiento real del usuario sin depender de entornos remotos y permitió verificar aspectos como la navegación, la intuición de las acciones, la correcta comprensión de los iconos y la claridad del flujo de trabajo. Así mismo, el docente asesor técnico y el asesor metodológico evaluaron el aplicativo en persona, lo que permitió validar su funcionamiento integral y certificar que cumplía con las condiciones necesarias para ser presentado como Producto Mínimo Viable.

Finalmente, desde la perspectiva del tiempo, el proyecto también resultó realizable porque antes del inicio del semestre ya existía un boceto preliminar del aplicativo y una hoja de ruta conceptual que había surgido de la investigación previa sobre la necesidad de una herramienta especializada en la evaluación de concursos de baile. Este insumo inicial permitió avanzar de manera ágil en las fases de análisis, diseño y construcción, optimizando los esfuerzos y garantizando que el desarrollo pudiera completarse dentro de los plazos establecidos por el proyecto de grado.

5.3.4 Limitación del aplicativo web

En términos de usabilidad y desempeño, el aplicativo web presenta limitaciones vinculadas principalmente a las condiciones bajo las cuales fue desarrollado. El tiempo disponible, el alcance del proyecto y los recursos tecnológicos acotados llevaron a priorizar funcionalidades esenciales, concentrándose en lograr una interfaz comprensible, tiempos de respuesta aceptables y compatibilidad con los navegadores y dispositivos más comunes. De igual forma, el sistema se ejecuta en entornos locales sin infraestructura de servidor dedicada, lo cual restringe pruebas de alta concurrencia o simulaciones de carga real propias de escenarios productivos. Pese a ello, el aplicativo cumple adecuadamente con los objetivos establecidos y el proceso de desarrollo se orientó a asegurar que el esfuerzo se concentra en entregar un producto estable y funcional, lo cual está plenamente alineado con la definición del PMV.

Por otro lado, el funcionamiento interno del software incorpora limitaciones propias del modelo de roles, permisos y validaciones definidos en su estructura. Cada rol (Superadministrador, Administrador, Juez y Participante) posee un conjunto específico de acciones permitidas, lo que asegura control, orden y seguridad en los procesos; sin embargo, esto también implica restricciones intencionales sobre qué operaciones puede realizar cada actor dentro del sistema. La visibilidad de datos, el acceso a formularios, la gestión de eventos, la inscripción y la calificación dependen estrictamente del rol activo, y cualquier interacción fuera de dicho alcance es bloqueada por las reglas de control implementadas. Asimismo, las validaciones para inscripción, asignación de jueces, estados de usuario y comprobación de disponibilidad funcionan bajo condiciones predefinidas que buscan evitar inconsistencias, pero que, al mismo tiempo, limitan la flexibilidad del sistema frente a escenarios atípicos. Estas restricciones forman parte de la lógica de negocio diseñada para garantizar integridad, coherencia y seguridad en el proceso de calificación, aun cuando implique limitar ciertas acciones que no están contempladas dentro del flujo estándar de la aplicación.

5.3.5 Enfoque metodológico

El desarrollo no se realizó siguiendo una metodología ágil tradicional, sino que se estructuró en fases o etapas definidas a partir del enfoque del Producto Mínimo Viable (PMV). Cada fase incluye un conjunto de actividades y entregables que permiten avanzar de manera progresiva hacia un prototipo funcional. Esta estrategia se considera adecuada dado que el proyecto se ejecuta de manera individual y requiere un esquema claro de prioridades. Además, la estructuración por fases garantiza una entrega ordenada de resultados parciales, lo que facilita validar avances en momentos específicos y realizar los ajustes pertinentes sin comprometer los tiempos establecidos.

5.3.6 Herramientas de desarrollo

Para la construcción del sistema se seleccionaron herramientas que permiten equilibrar eficiencia, accesibilidad y facilidad de implementación.

- **PHP** se emplea como lenguaje base de programación debido a su integración nativa con bases de datos MySQL y a que permite estructurar la aplicación en el patrón MVC. Con ello se gestionará la lógica de negocio, la manipulación de datos y la conexión con la base de datos.
- **SQL**, a través de un documento con extensión .sql, se utilizará para definir las tablas, relaciones y consultas necesarias. En este sentido, se emplearán sentencias para la unión de tablas de información que faciliten la consulta y consolidación de información entre las distintas entidades del sistema.
- **JavaScript** tendrá un papel clave en el frontend, ya que permitirá validar formularios de manera inmediata y generar interacciones dinámicas que mejoran la experiencia de usuario.
- **CSS y plantillas Bootstrap** se implementarán en la interfaz visual para garantizar un diseño responsivo, moderno y coherente con las necesidades de accesibilidad del proyecto. El uso de Bootstrap permitirá agilizar el desarrollo de las vistas y CSS brindará la posibilidad de personalizar los detalles estéticos de acuerdo con los requerimientos.

5.4 Fase 2: Análisis

En esta etapa se definió la estructura conceptual y técnica del aplicativo, de modo que el desarrollo posterior se llevará a cabo con una base sólida y coherente.

En primer lugar, la arquitectura de la aplicación se fundamenta en el patrón Modelo-Vista-Controlador (MVC) con PHP, ya que este esquema separa la lógica del negocio, la gestión de datos y la presentación en capas independientes, lo cual favorece la organización del código y facilita la escalabilidad del proyecto.

En segundo lugar, se realizó la selección de herramientas más adecuadas para un prototipo académico de bajo costo. Se optó por un servidor con hosting compartido y una base de datos MySQL debido a su disponibilidad y facilidad de integración con PHP. Asimismo, se incorporó el uso de JavaScript y CSS, complementados con plantillas de Bootstrap, con el fin de garantizar una interfaz responsiva, clara y accesible desde diferentes dispositivos. Finalmente, se definió la necesidad de un dominio web que permita acceso multiplataforma y que refuerce la visibilidad del sistema.

En cuanto a los artefactos del negocio, se identificaron los actores y elementos esenciales del proceso de evaluación, entre los que se destacan los jueces, los eventos, las categorías de baile, los criterios de calificación, las puntuaciones y los resultados. Estos componentes permiten estructurar la información de manera lógica y reflejar con fidelidad la dinámica de los concursos.

De manera complementaria, se precisaron los requisitos funcionales que orientan la implementación del sistema. Entre ellos se encuentran el registro y la autenticación de jueces y organizadores, la visualización de criterios de calificación, la carga de puntajes por categoría y la generación automática de resultados, con la opción de exportar reportes consolidados.

Por último, se elaboraron historias de usuario con base en los perfiles de jueces, organizadores y participantes, ya que este recurso metodológico facilita la comprensión de las necesidades específicas de cada actor y permite validar que las funcionalidades respondan a escenarios reales de uso.

5.5 Fase 3: Diseño

En esta fase se consolidaron los lineamientos técnicos y funcionales del aplicativo mediante la elaboración de documentación y diagramas que permiten representar de manera clara la lógica de la información y el comportamiento del sistema.

5.5.1 Requerimientos técnicos y funcionales

Los requerimientos definidos en las fases anteriores fueron detallados en función de las historias de usuario, lo cual permitió identificar las necesidades específicas de cada rol (administrador, juez, participante y academia). Este proceso garantizó que el diseño respondiera a los objetivos de transparencia, seguridad y escalabilidad del sistema, evitando la ambigüedad en la definición de funcionalidades.

5.5.2 Diagramas UML

Para representar el diseño de la aplicación se elaboraron diferentes diagramas UML que cumplieron con la función de analizar la solución desde varias perspectivas, lo cual garantiza la claridad en la definición de la lógica funcional y de la organización de los datos.

En primer lugar, los diagramas de casos de uso se utilizaron con el propósito de describir las interacciones entre los actores definidos en el sistema, entre ellos el super administrador, el administrador de academia, el juez en sus dos roles como jurado y como participante, el participante de academia y el participante independiente, junto con los módulos de la aplicación. Gracias a este recurso fue posible representar de manera gráfica los requerimientos funcionales, lo que facilitó la comprensión de las acciones que cada usuario podía realizar dentro de la plataforma. De esta manera, los casos de uso se constituyeron en la base para la definición de las historias de usuario y, posteriormente, de las validaciones necesarias para cada módulo.

Como se aprecia en la Figura 12, el rol de *superadministrador* reúne las funciones de mayor alcance dentro del sistema, pues desde este perfil se administra la totalidad de las entidades principales del aplicativo. Este actor tiene la facultad de crear, editar y modificar el estado de academias, participantes, jueces, eventos, divisiones y criterios de calificación, además de acceder a los reportes estadísticos globales asociados a cada academia. Gracias a este conjunto de atribuciones, el sistema garantiza la existencia de un usuario responsable de supervisar el funcionamiento general y mantener la consistencia operativa de toda la plataforma. De este modo, se asegura que la información permanezca actualizada y que los procesos administrativos se ejecuten con rigor y seguridad.

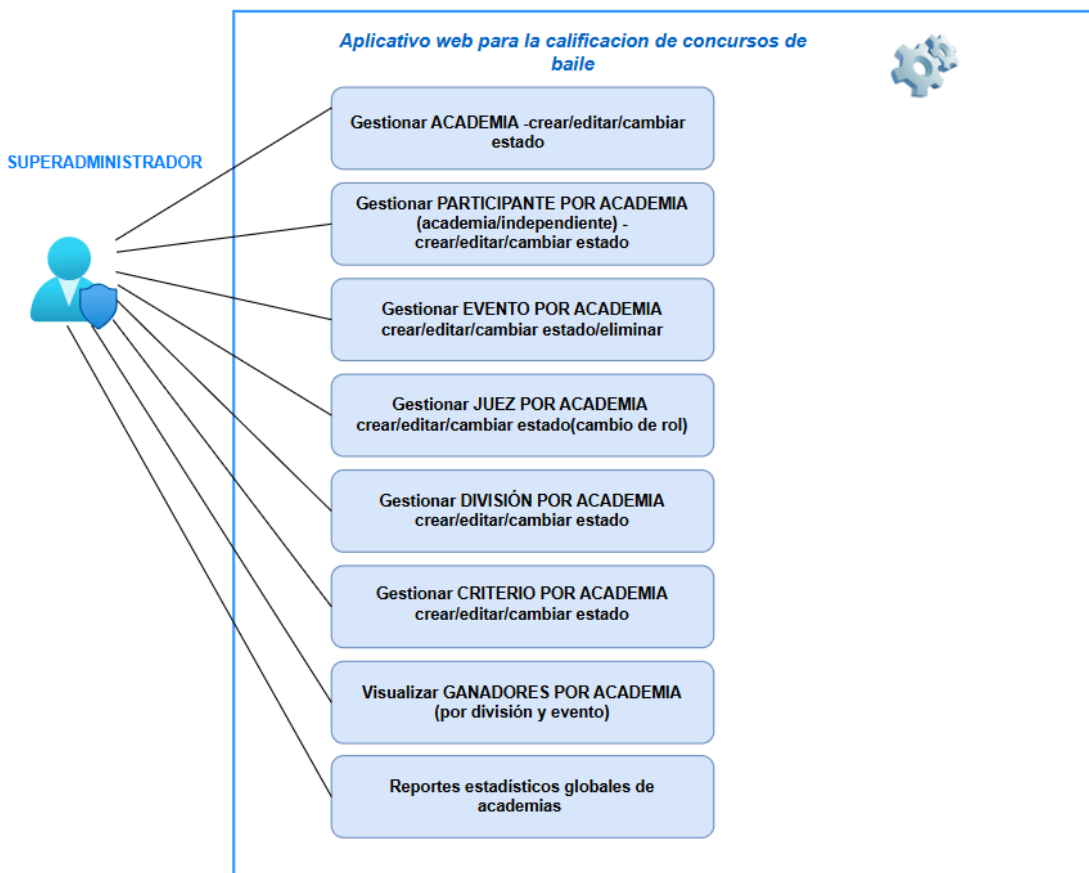


Figura 12. Caso de uso rol super administrador.
Fuente: Elaboración propia.

Como se evidencia en la Figura 13, el *administrador de academia* cuenta con un nivel de gestión enfocado exclusivamente en las operaciones de la institución a la que pertenece. Desde

este rol es posible crear, editar y cambiar el estado de participantes, jueces, divisiones, criterios y eventos propios de su academia, permitiendo un manejo autónomo de los procesos internos. Asimismo, este usuario puede visualizar los ganadores por división y evento, lo que facilita el seguimiento de los resultados obtenidos en cada concurso. La distribución de funciones para este rol permite mantener un equilibrio adecuado entre autonomía local y supervisión global del superadministrador, fortaleciendo la organización y el flujo de trabajo dentro de cada academia.

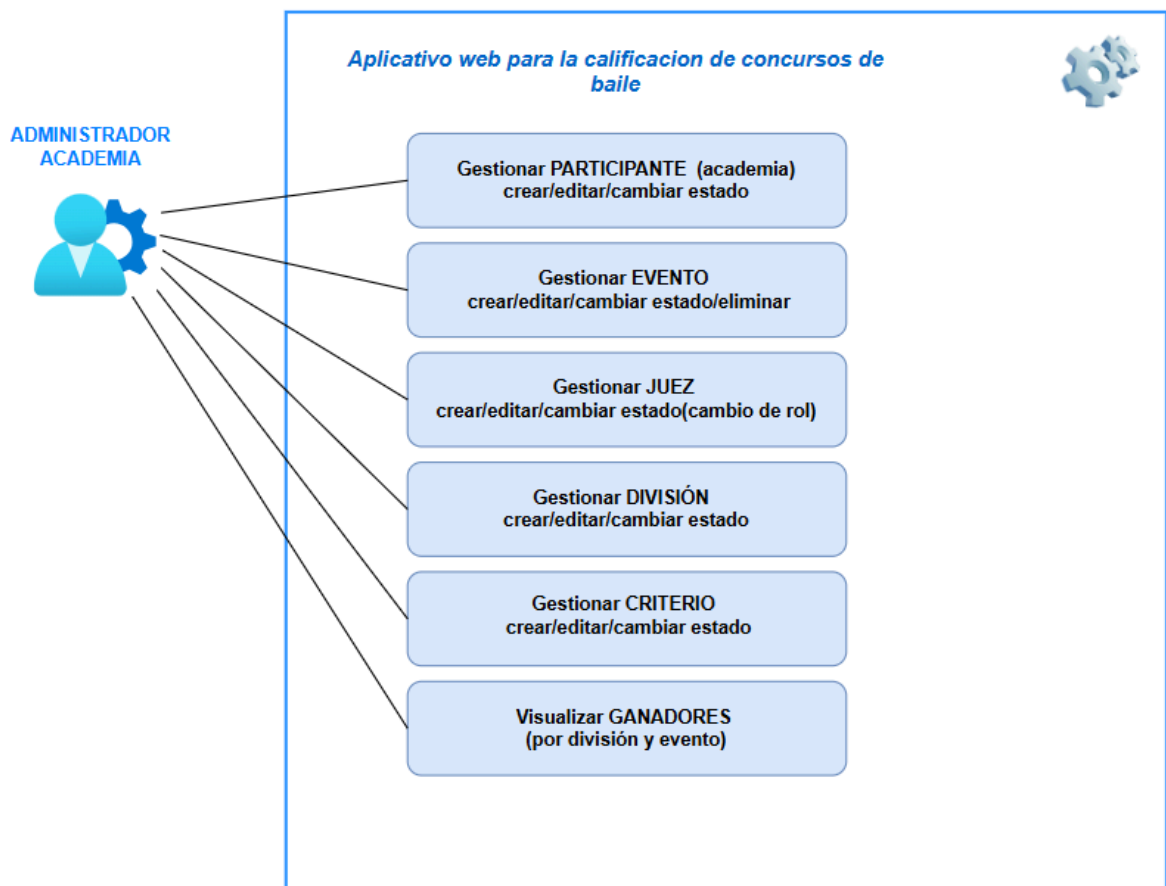


Figura 13. Caso de uso rol administrador academia.
Fuente: Elaboración propia.

En la Figura 14 se muestra el caso de uso correspondiente al *juez*, un actor cuyo enfoque está dirigido exclusivamente a los procesos de evaluación dentro del sistema. Entre sus funciones

destacan calificar participantes según los criterios definidos, consultar el historial de calificaciones, exportar informes y realizar búsquedas filtradas de eventos celebrados. Además, el juez puede visualizar reportes estadísticos relacionados con sus evaluaciones, lo cual proporciona transparencia y trazabilidad en su desempeño. La delimitación de estas funciones permite que el juez se concentre exclusivamente en su labor evaluativa, evitando el acceso a módulos administrativos que no forman parte de su rol.

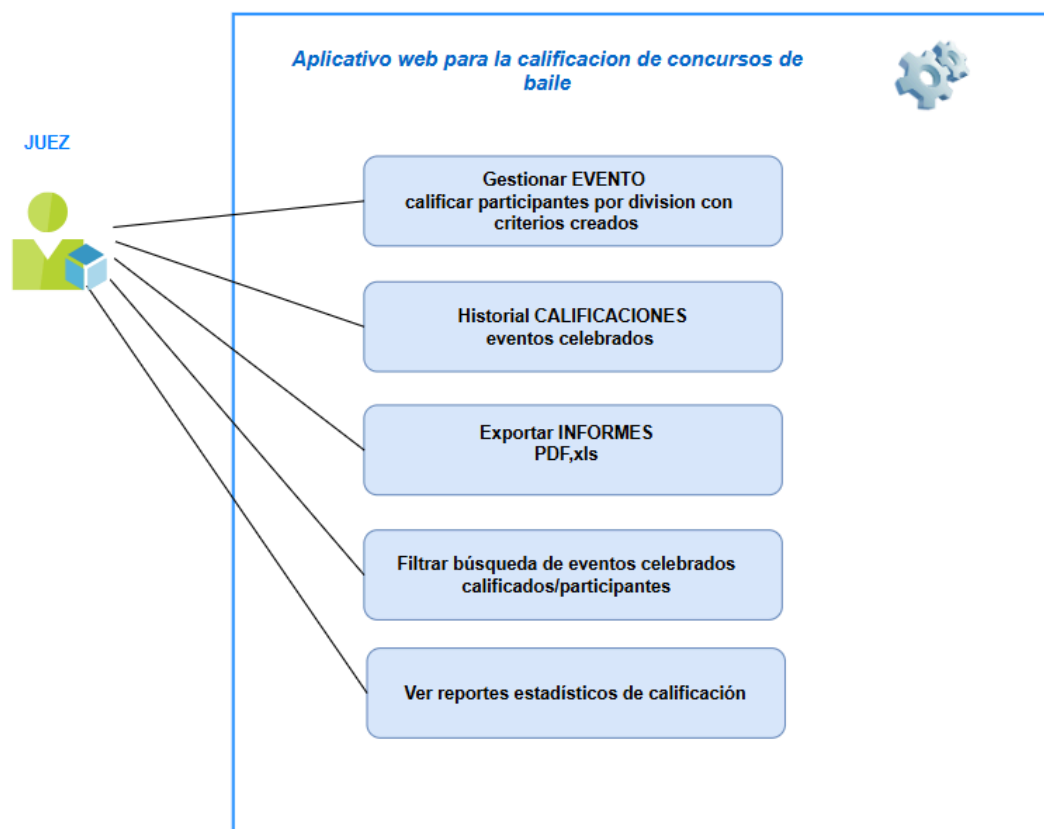


Figura 14. Caso de uso rol juez.
Fuente: Elaboración propia.

De acuerdo con lo ilustrado en la Figura 15, el *participante*, ya sea perteneciente a una academia o inscrito de manera independiente o un juez con su rol como jurado inactivo,

interactúa con los módulos orientados a su experiencia dentro del concurso. Entre sus funciones se incluyen visualizar los eventos disponibles, revisar sus inscripciones, acceder a la pasarela de pagos cuando corresponda y consultar las calificaciones obtenidas en concursos celebrados. También tiene la posibilidad de observar los ganadores según su participación, así como revisar reportes estadísticos básicos que reflejan su desempeño. Este conjunto de funcionalidades ofrece una experiencia clara y accesible para cualquier usuario, garantizando un proceso de navegación simple y orientado a sus necesidades.

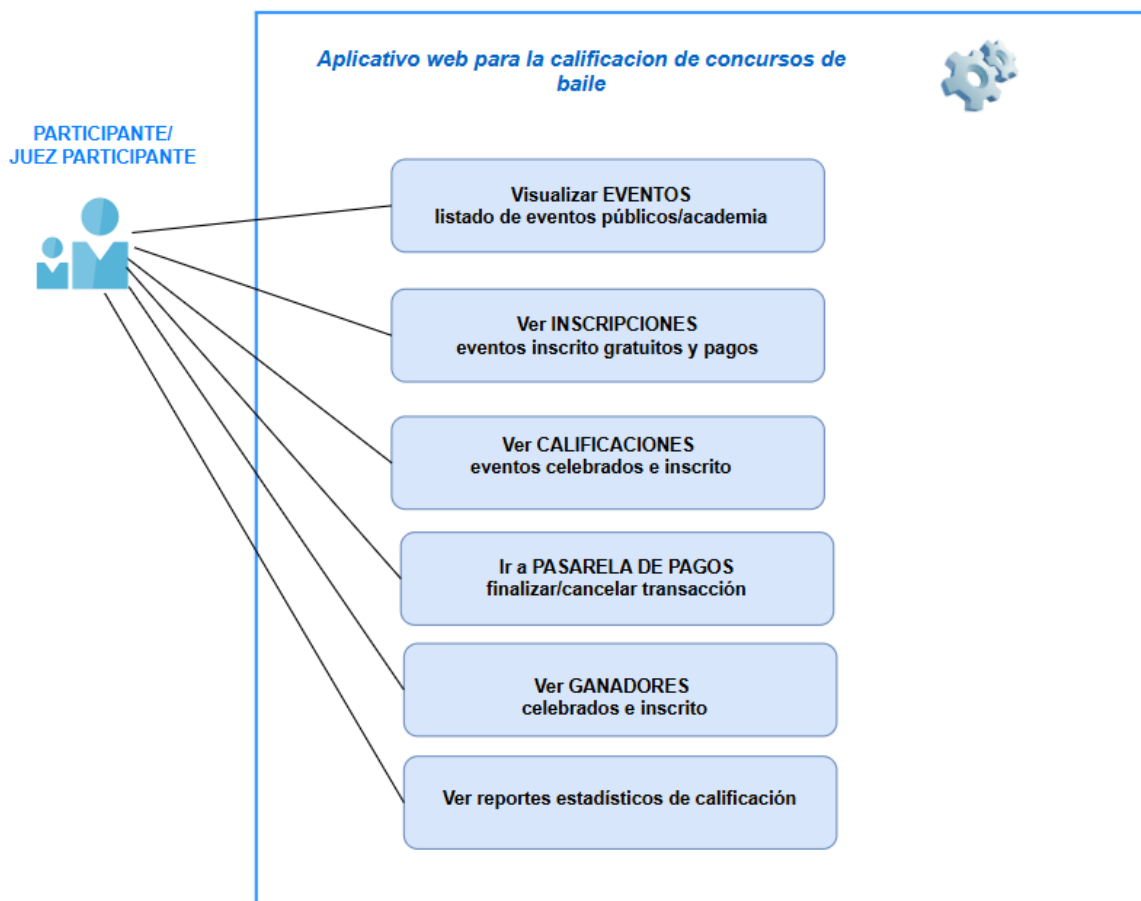


Figura 15. Caso de uso rol participante/juez participante.
Fuente: Elaboración propia.

En segundo lugar, se diseñó el modelo entidad-relación con el fin de mostrar la forma en que se organiza, relaciona y gestiona la información en el aplicativo. Este diagrama fue un insumo fundamental para la construcción de la base de datos, ya que permitió definir las entidades

principales, entre ellas administradores, jueces, participantes, academias, eventos y resultados, además de precisar sus atributos y las relaciones existentes entre ellas. No obstante, el modelo fue objeto de cuatro modificaciones realizadas en el marco de las asesorías técnicas, debido a que la primera versión presentaba limitaciones relacionadas con la seguridad y la escalabilidad.

5.5.3 Diseño del diagrama y modelo entidad–relación

Iteración 1: modelo inicial y motivos de descarte

En la primera versión del modelo entidad–relación, el sistema se diseñó para funcionar con una sola academia. Por esa razón, todos los usuarios quedaban ligados a la misma institución y, además, los roles de administrador y juez tenían permisos muy amplios para registrar y modificar datos sensibles. Incluso los participantes podían editar su información sin controles suficientes. Por otro lado, las divisiones, categorías y criterios de evaluación no estaban definidos previamente: el administrador los ingresaba de forma manual en cada evento, lo que hacía el proceso más lento y aumentaba el riesgo de errores, sesgos y resultados difíciles de comparar. En conjunto, el esquema reproducía la práctica interna de esa academia, pero no ofrecía buenas garantías de seguridad, integridad ni crecimiento.

Como se aprecia en la Figura 16, este primer esquema incluía únicamente las entidades *Administrador*, *Academia*, *Participante*, *Evento*, *Juez* y *Resultado*, representando un diseño básico que se centraba en reproducir el funcionamiento interno de una sola institución. Aunque cumplía con los requisitos preliminares, el modelo no aseguraba niveles adecuados de independencia entre academias, ni restricciones claras por rol, ni un manejo seguro de la información sensible. Esto evidenciaba limitaciones importantes desde la perspectiva de escalabilidad y gobernanza de datos.

A partir de las asesorías técnicas, se propuso un rediseño ya que se identificó que este enfoque no respondía a buenas prácticas de desarrollo ni era sostenible para un proyecto con miras a la escalabilidad. Por tal razón, fue necesario plantear un rediseño que corrigiera estas deficiencias. En este nuevo modelo se buscó garantizar la independencia entre roles, establecer

controles más estrictos en la edición de la información y permitir la posibilidad de integrar múltiples academias dentro de la misma plataforma. Con ello se fortaleció la seguridad del sistema, se favoreció la transparencia en la gestión de datos y se amplió el alcance de la aplicación en correspondencia con los objetivos planteados en el proyecto.

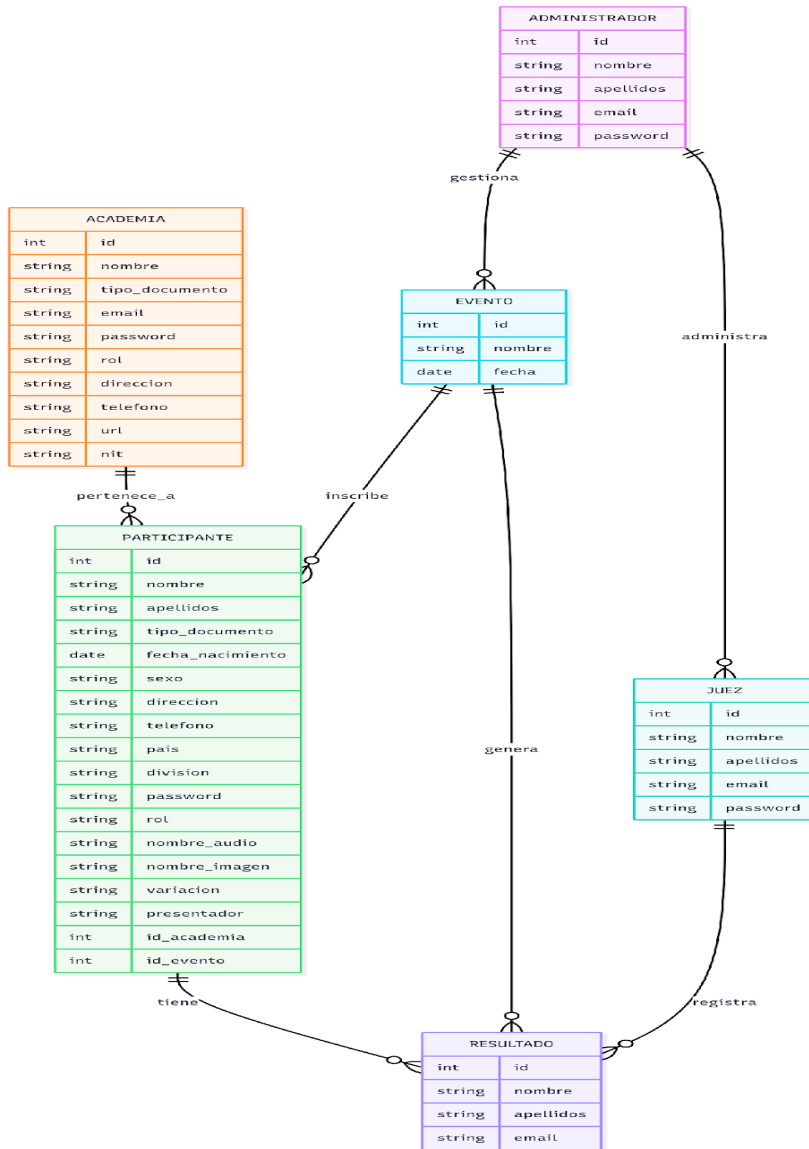


Figura 16. Diseño inicial diagrama entidad-relación con estructura limitada a una sola academia.
Fuente: Elaboración propia.

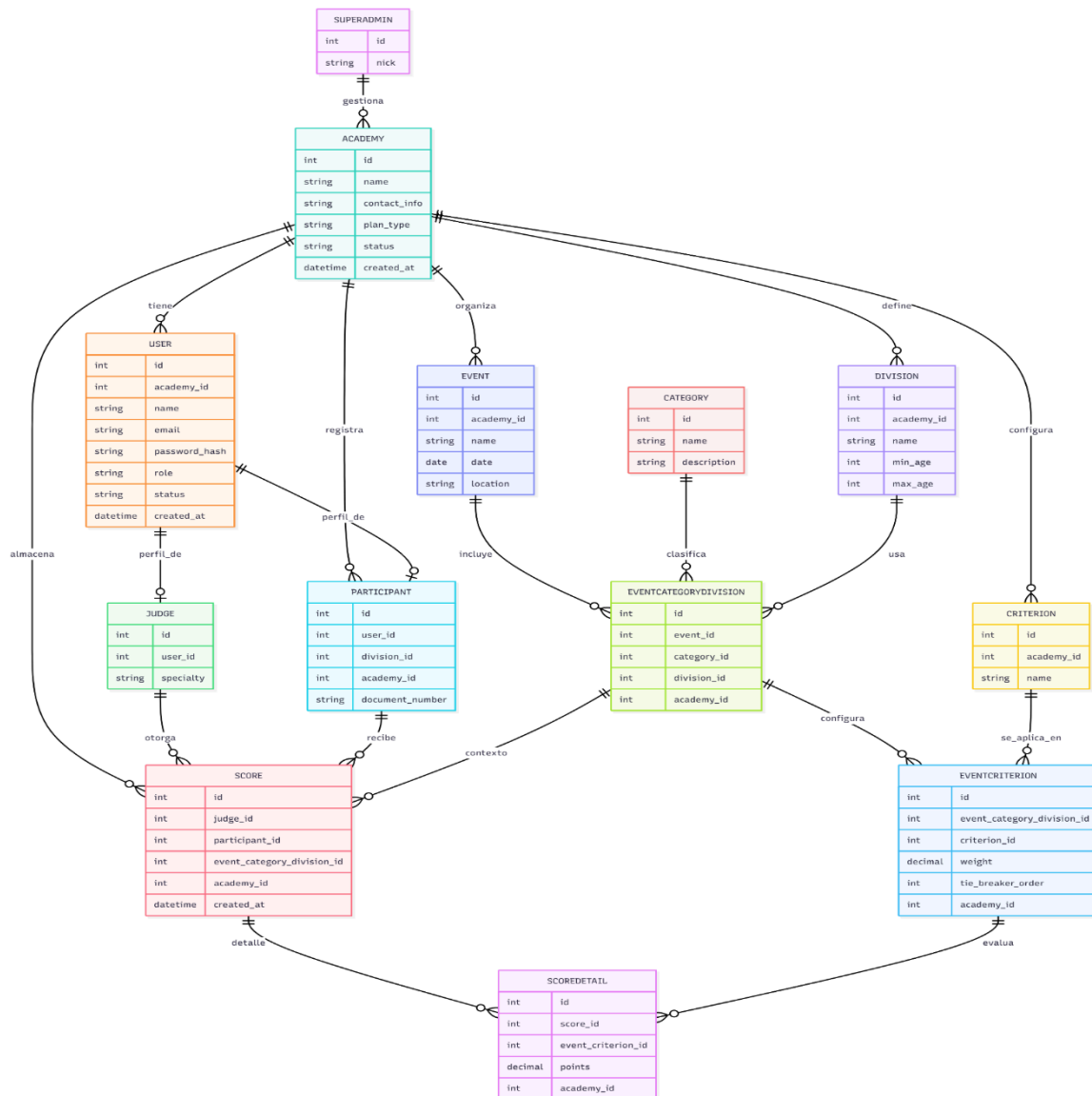
Iteración 2: producto multi academia y normalización de la evaluación

En la segunda versión del modelo entidad–relación, el sistema se estructuró con un enfoque mucho más robusto y adecuado a las necesidades del proyecto. En esta fase se separaron mejor las funciones de cada rol y se aplicó el principio de mínimo permiso, de modo que cada usuario accediera únicamente a las acciones que realmente requería. Asimismo, se establecieron controles claros para la edición de datos y la trazabilidad de cambios, lo que permitió evitar modificaciones no autorizadas y fortalecer la integridad de la información registrada.

Como se muestra en la Figura 17, esta iteración incluyó nuevas entidades que ampliaron considerablemente el funcionamiento del sistema: *Usuario*, *Academia*, *Participante*, *Juez*, *División*, *Criterio*, *Evento* y *judge_score*. Del mismo modo, se incorporaron *event_criterio* y *event_division*, elementos fundamentales para administrar los criterios y divisiones asignados a cada evento. Gracias a ello, cada competencia podía configurarse con diferentes tipos de criterios, puntajes máximos y pesos específicos, lo cual daba respuesta a la diversidad que caracteriza al mundo de los concursos de baile. La entidad *judge_score* permitió relacionar cada calificación con el evento, el participante, el criterio evaluado y el juez responsable, garantizando trazabilidad completa en los procesos calificativos y facilitando la resolución de empates con reglas coherentes y verificables.

A partir de esta reorganización, se definió la figura del superadministrador como responsable de gestionar todas las academias registradas, mientras que cada academia contó con su propio administrador y con roles operativos claramente delimitados. El administrador de academia fue el único autorizado para registrar participantes, crear eventos y, cuando era necesario, añadir divisiones o criterios nuevos basados en catálogos previamente establecidos. Con el uso de *event_criterio* y *event_division*, los eventos pudieron ajustarse a múltiples modalidades, rangos de edad y estilos de baile, manteniendo simultáneamente una estructura estandarizada que reducía inconsistencias y hacía más transparente el proceso de registro y evaluación. En conjunto, el sistema ganó seguridad, flexibilidad y capacidad de comparación entre eventos, al mismo tiempo que fortaleció los cimientos para la escalabilidad del proyecto.

Durante las asesorías técnicas se identificaron falencias que aún afectaban los procesos de inscripción y evaluación. Entre ellas estaban la ausencia de mecanismos para gestionar participantes independientes, la falta de reglas para manejar usuarios asociados a varias academias y la inexistencia de registros históricos que permitieran rastrear cambios en la afiliación de un participante. Por esta razón, se formularon reglas de validación y flujos de verificación que confirmaban el estado del usuario, administraban las múltiples afiliaciones con prioridades claras y registraban los movimientos relevantes en el tiempo. Dichos ajustes



buscaban garantizar que la inscripción, la evaluación y la asignación de roles se mantuvieran coherentes, trazables y justas en todos los eventos celebrados.

Figura 17. Segunda versión del modelo entidad relación con estructura multi academia y normalización del proceso de evaluación.

Fuente: Elaboración propia.

Iteración 3: gobierno datos, reglas de integridad y operación abierta

En la tercera versión del modelo se reorganizó la forma en que se gestionaban los usuarios y sus roles dentro del sistema, ya que se buscó una estructura más clara y ajustada a las necesidades reales del proyecto. Durante esta iteración se definió que, cuando un administrador de academia o el superadministrador registraban a un usuario, este quedaba directamente asociado a una academia específica, lo que permitió mantener un control coherente sobre la pertenencia de quienes sí formaban parte de una institución. Paralelamente se revisó el funcionamiento de los

eventos y se precisó que estos solo podían ser creados por un administrador de academia o por el superadministrador; sin embargo, cada evento podía configurarse con una visibilidad pública o solo academias, lo que abría la posibilidad de recibir participantes externos o restringir la inscripción únicamente a quienes pertenecieran a las academias registradas.

En este punto también se incorporó un mecanismo para integrar a los participantes independientes, dado que muchos concursos de baile admiten competidores sin afiliación. Para ello se habilitó el registro directo desde el aplicativo web mediante un formulario inicial que creaba un usuario con credenciales y acceso a su propio módulo. Una vez completado el proceso, el sistema almacenaba la información en las tablas *user* y *participant*, asignándolo a un identificador especial que representaba la categoría “sin academia”. Gracias a este mecanismo, los participantes independientes quedaron integrados al sistema sin generar excepciones y podían inscribirse en eventos cuya visibilidad fuera pública, lo que permitió su participación sin afectar la coherencia del modelo general.

En la Figura 18 se evidencia que esta reestructuración también permitió resolver un problema crítico que se presentó durante las asesorías técnicas: la coexistencia de roles en un mismo evento. Antes era común que un juez quisiera competir como participante en una competencia puntual, lo cual generaba conflictos lógicos y riesgos para la imparcialidad. Para corregirlo se incorporó la tabla *judge_participant*, que registraba el rol específico que un usuario desempeñaba en cada evento. Con este ajuste, el sistema garantizó que un juez pudiera estar inactivo como juez y activo como participante para un evento determinado, sin generar interferencias. Así, cuando una persona ingresaba al sistema con su rol de juez inactivo, su panel desplegaba un menú idéntico al de un participante —ya fuera de academia o independiente— y podía visualizar eventos públicos o eventos de academia que correspondieran a su contexto, tanto en modalidad gratuita como paga. Además, este diseño aseguró que los eventos pendientes de calificación anteriores a la inactivación no se vieran afectados y que, de igual manera, un juez que se activará después de haberse inscrito previamente como participante mantuviera su inscripción original sin alteraciones. Esto permitió manejar cambios de estado sin romper la lógica interna ni las reglas de equidad.

En paralelo también se mejoró la manera en que se consolidaban los resultados finales, ya que la estructura previa fue reemplazada por la tabla *event_winner*. Esta tabla almacenaba el veredicto por división y evento, permitiendo registrar cuándo la decisión había sido unánime, cuándo se había tomado por mayoría o cuándo se había determinado por puntaje acumulado en los casos en que cada juez seleccionaba un ganador distinto. Con ello, el sistema separó de forma precisa la calificación individual —almacenada en las tablas de puntajes por criterio— del acto final de determinación del ganador, fortaleciendo el gobierno de datos y aumentando la transparencia en los procesos de desempate.

Finalmente, se integraron de manera más coherente las entidades *event_criterio* y *event_division*, que permitieron establecer criterios personalizados y divisiones específicas en cada evento. Esto posibilita asignar pesos diferentes, puntajes máximos y rangos etarios bien definidos, de modo que cada competencia refleja la diversidad de estilos, formatos y modalidades presentes en los concursos de baile actuales. En conjunto, la tercera iteración consolidó un modelo más seguro, flexible y escalable, que representa de manera fiel la complejidad del entorno real y sentó las bases para una gestión más transparente y ordenada.

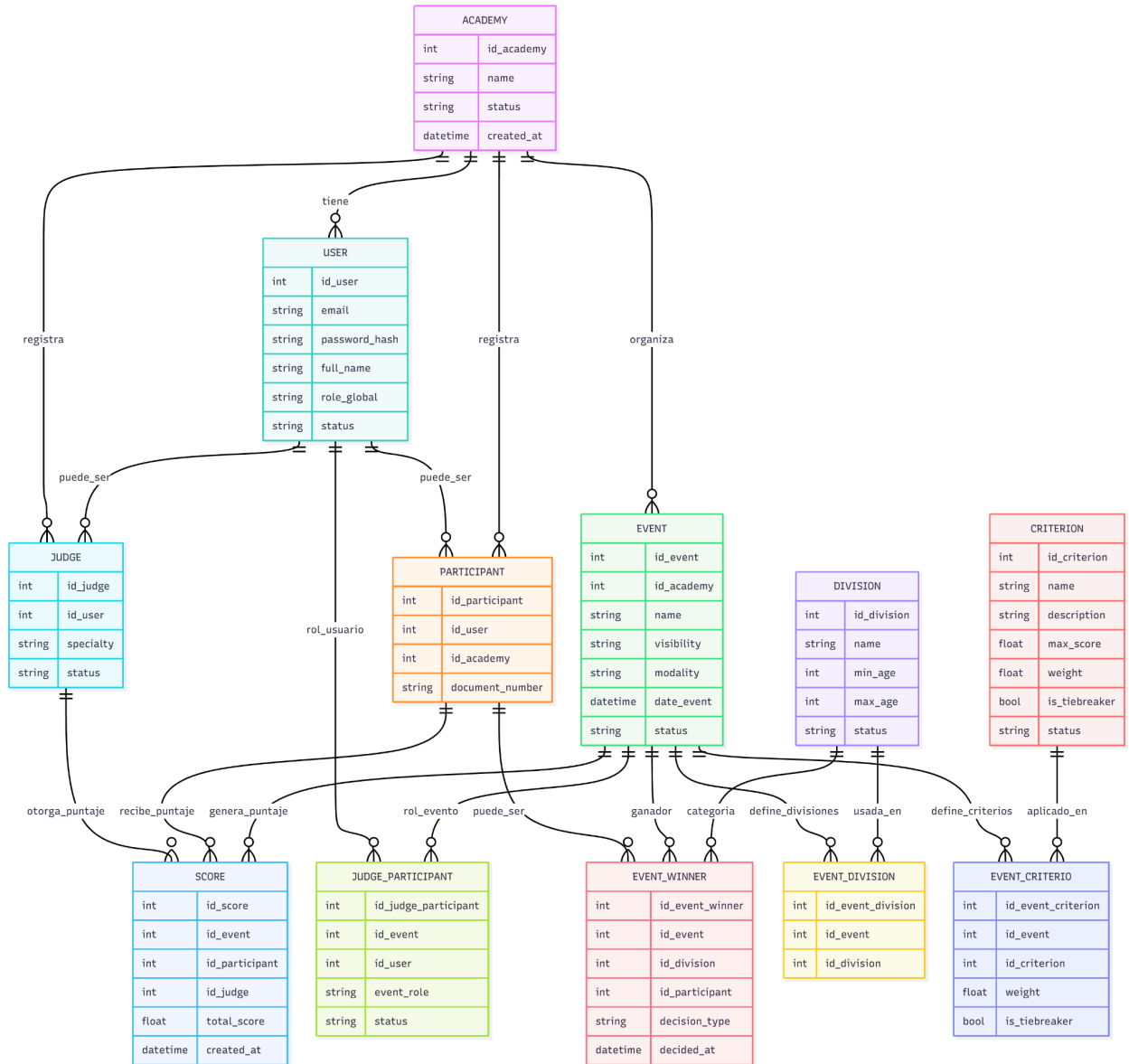


Figura 18. Tercera versión del modelo entidad–relación, con integración de participantes independientes, resolución de roles por evento y estructura final para evaluación y determinación de ganadores.
Fuente: Elaboración propia.

Iteración 4: normalización de listas como datos maestros del sistema y soporte para información geográfica

En la cuarta versión del modelo entidad–relación se decidió fortalecer la forma en que se almacenaban ciertos datos que eran utilizados de manera repetida en la aplicación. Hasta ese momento, la información relacionada con departamentos, ciudades y especialidades de los jueces

se aplicaba principalmente desde la capa de interfaz por medio de listados internos, lo que implicaba que cualquier cambio exigiera modificar el código. Con el fin de mejorar la seguridad, la integridad y la facilidad de mantenimiento del sistema, se incorporaron tres nuevas tablas en la base de datos llamadas *department*, *city* y *specialty*. De este modo, en lugar de depender de archivos estáticos o estructuras internas difíciles de actualizar, la aplicación pasó a apoyarse en listas oficiales de departamentos, ciudades y especialidades centralizadas y administrables desde la propia base de datos.

Como se muestra en la Figura 19, la tabla *department* almacena los departamentos disponibles, mientras que la tabla *city* registra las ciudades y municipios asociados a cada departamento. Esta organización permite que los formularios de registro muestran listas desplegables coherentes, en las que primero se selecciona el departamento y luego se ofrecen únicamente las ciudades que pertenecen a ese departamento, lo que facilita el diligenciamiento correcto de la información. Por su parte, la tabla *specialty* guarda las especialidades que pueden asignarse a los jueces, lo que permite describir de manera más precisa su perfil profesional. Gracias a esta estructura, cuando se requiere añadir una nueva ciudad o especialidad, basta con registrar en la base de datos utilizando herramientas de administración como phpMyAdmin o, en el futuro, consumir un servicio externo que suministre estos catálogos, sin necesidad de alterar el código de la aplicación ni poner en riesgo su estabilidad.

Esta decisión también respondió a una preocupación importante relacionada con el mantenimiento a largo plazo. Si la información geográfica y las especialidades se hubieran dejado “embebidas” en archivos de configuración o en fragmentos de código, cualquier cambio, por pequeño que fuera, habría requerido la intervención de un desarrollador y habría aumentado el riesgo de introducir errores en funcionalidades que ya estaban probadas. Al trasladar estos datos a tablas dedicadas, el sistema ganó flexibilidad y se alineó mejor con los principios de gobierno de datos, ya que los catálogos pueden crecer, actualizarse o adaptarse a nuevas necesidades sin comprometer el correcto funcionamiento del resto de la aplicación.

De esta manera, con la incorporación de *department*, *city* y *specialty* en la cuarta iteración, el modelo entidad–relación quedó completamente refinado y adaptado a las necesidades de los diferentes roles del sistema. Las cuatro etapas de rediseño permitieron pasar de un esquema

inicial limitado a una academia a una solución capaz de gestionar múltiples academias, participantes independientes, reglas de evaluación complejas y determinación transparente de ganadores. En consecuencia, el modelo resultante cumplió con los requisitos definidos para el producto mínimo viable y se consolidó como una base sólida para futuras ampliaciones del sistema de evaluación de concursos de baile.

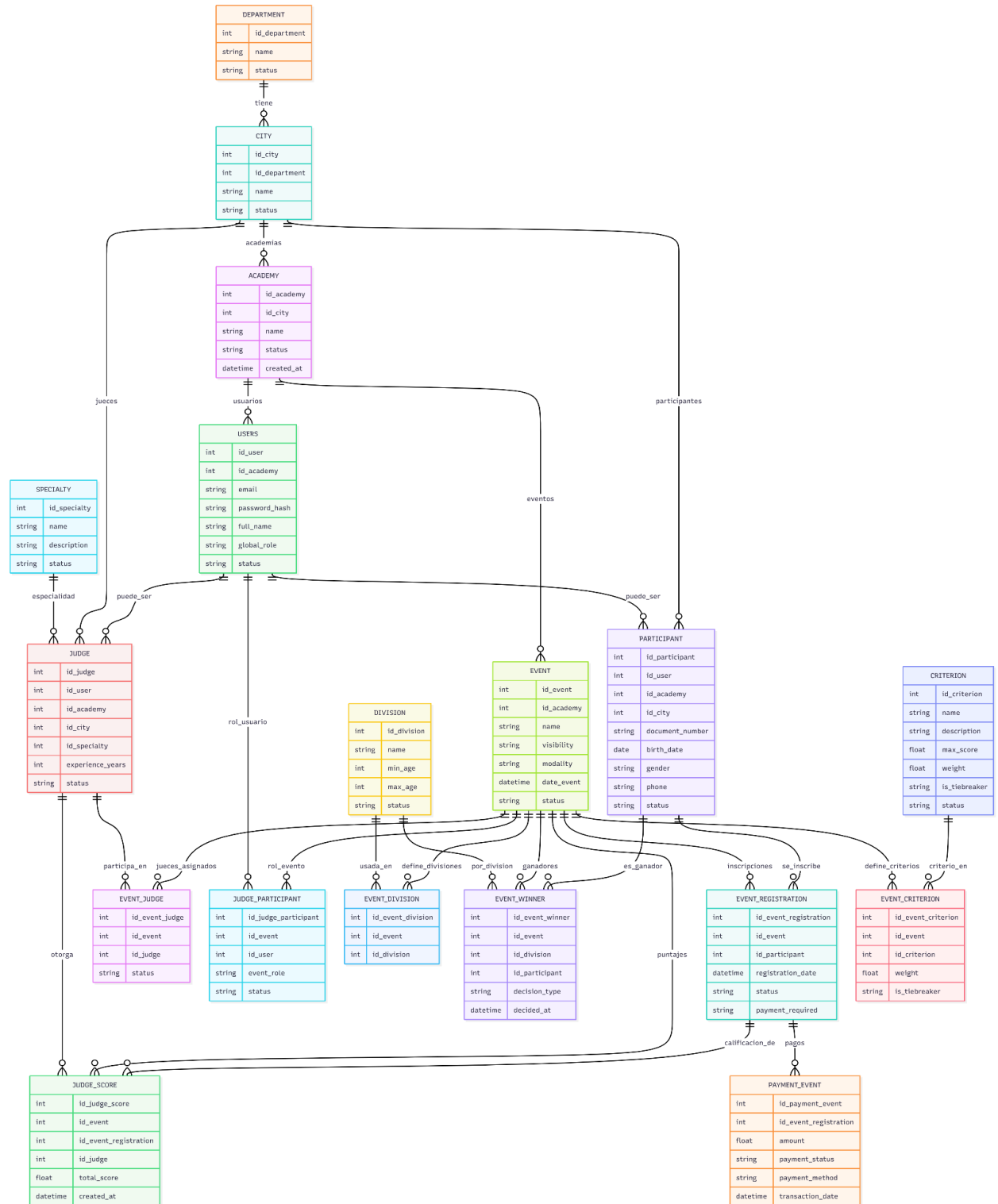


Figura 19. Cuarta versión del modelo entidad-relación, con Incorporación de las tablas de referencia para departamentos, ciudades y especialidades.
Fuente: Elaboración propia.

5.5.4 Diagramas de flujo

En tercer lugar, los diagramas de flujo fueron elaborados para cada uno de los módulos principales, particularmente aquellos asociados a la gestión de participantes, jueces y eventos mediante operaciones de creación, lectura, actualización y eliminación. El objetivo de estos diagramas consistió en describir de manera detallada la lógica de interacción y las validaciones de roles, lo que permitió anticipar posibles fallos, reducir ambigüedades y asegurar la trazabilidad de las operaciones en el sistema.

En el caso de los participantes, el diagrama de flujo correspondiente describe de manera estructurada cómo se desarrollan las operaciones fundamentales de gestión dentro del módulo. Tal como se aprecia en la Figura 20, el proceso inicia con la verificación de las credenciales en la pantalla de acceso; una vez validado el ingreso, el usuario es dirigido al dashboard y desde allí accede a la sección destinada a la administración de participantes. Dentro de este módulo, el flujo contempla las funciones esenciales del sistema: registrar un nuevo participante, editar la información existente, consultar el detalle de un registro y eliminarlo cuando sea necesario.

Cada una de estas acciones pasa por un proceso de validación de datos, lo que garantiza que el sistema identifique oportunamente errores o inconsistencias antes de ejecutar cualquier operación. En caso de detectar información incompleta o incorrecta, se muestra un mensaje de alerta que permite corregir los campos; si los datos son válidos, la acción se completa y la lista de participantes se actualiza de inmediato, reflejando el resultado de la creación, actualización o eliminación.

El diagrama también integra los puntos de retorno al dashboard y la opción de cerrar sesión, lo que evidencia la estructura lógica del recorrido completo dentro del módulo. De esta manera, la Figura 20 representa un flujo coherente, ordenado y alineado con las responsabilidades y permisos establecidos para cada rol, asegurando que la gestión de participantes se realice de manera segura, consistente y completamente trazable.

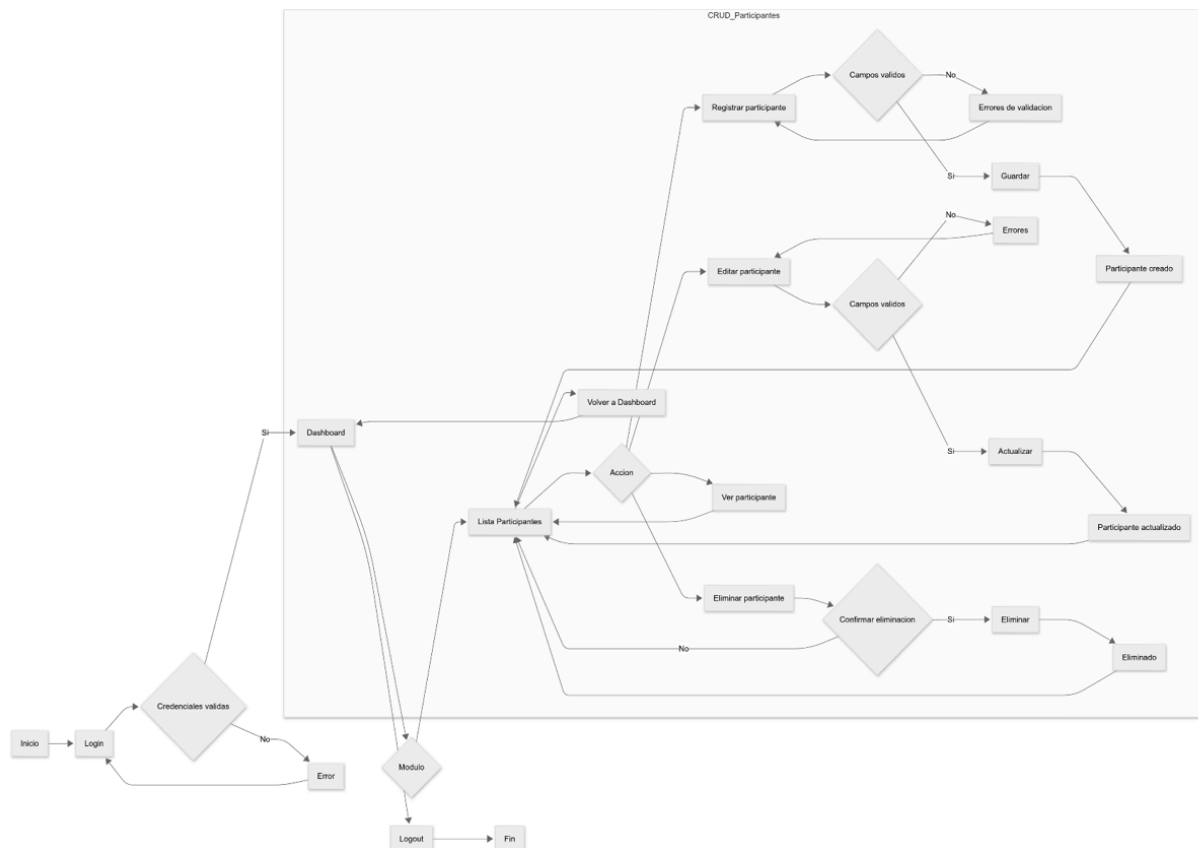


Figura 20. Diagrama de flujo CRUD jueces
Fuente: Elaboración propia.

CRUD Participantes

Secuencia principal

1. **Ingreso y validación.** El usuario inicia sesión. Si las credenciales son válidas, el sistema redirige al *Dashboard* y, desde allí, al módulo de participantes; en caso contrario, se muestra un mensaje de error.
2. **Lista de participantes.** El sistema carga *Lista de participantes* como pantalla base; desde aquí se inician todas las acciones.
3. **Registrar participante.**
 - a) Se abre el formulario *Registrar participante*.
 - b) Se ejecuta la validación de campos.

- o **Sí:** el sistema guarda y notifica al Participante *creado*.
 - o **No:** el sistema muestra errores de validación y permite corregir.
 - c) Al finalizar, se retorna a *Lista de participantes* y, si el usuario lo decide, al *Dashboard*.
4. **Ver participantes.** Desde la lista, la opción *Ver* muestra el detalle sin modificar y luego regresa a la lista.
5. **Editar participante.**
- a) Se abre *Editar participante*.
 - b) Se valida nuevamente la integridad de los campos.
 - o **Sí:** el sistema actualiza y confirma *Participante actualizado*.
 - o **No:** el sistema presenta errores para corrección.
 - c) Se retorna a la lista y, opcionalmente, al *Dashboard*.
6. **Eliminar participante.**
- a) Se elige *Eliminar* y se solicita *Confirmar eliminación*.
 - o **Sí:** el sistema elimina y muestra *Eliminado*.
 - o **No:** el sistema cancela y vuelve a la lista.
7. **Salida.** En cualquier punto, el usuario puede cerrar sesión (*Logout*) y finalizar.

Controles y reglas visibles

- Validación de campos obligatorios y formato de datos.
- Mensajes de error en edición con indicación del campo.
- Confirmación de eliminación con advertencia de impacto.
- Reglas de unicidad cuando aplique (documento, correo).
- Trazabilidad de cambios y retorno consistente a la lista.

Resultado esperado

- La creación y actualización se realizan solo si los datos cumplen las reglas.
- La consulta no altera información y la eliminación requiere confirmación.
- Se preserva la integridad de los datos y la navegación es predecible.

En el caso de la gestión de jueces, se elaboró un diagrama de flujo específico que permitió visualizar el recorrido completo que realiza el administrador de academia o el super administrador al crear, editar, consultar o eliminar registros de este rol. Como se observa en la Figura 21, el proceso se inicia con la validación de credenciales y el acceso al *dashboard* correspondiente; desde allí, el usuario ingresa al listado de jueces, donde puede seleccionar la acción que desea ejecutar. Si opta por registrar un juez, el sistema presenta un formulario en el que se validan los campos obligatorios antes de guardar la información; de manera similar, cuando se edita un juez, los datos se actualizan solo si las validaciones son correctas, mientras que, para la eliminación, se muestra una confirmación previa que evita borrar registros por error. En todos los casos, el flujo retorna al listado de jueces y, posteriormente, al menú principal, lo que asegura una navegación consistente y controlada dentro del aplicativo web.

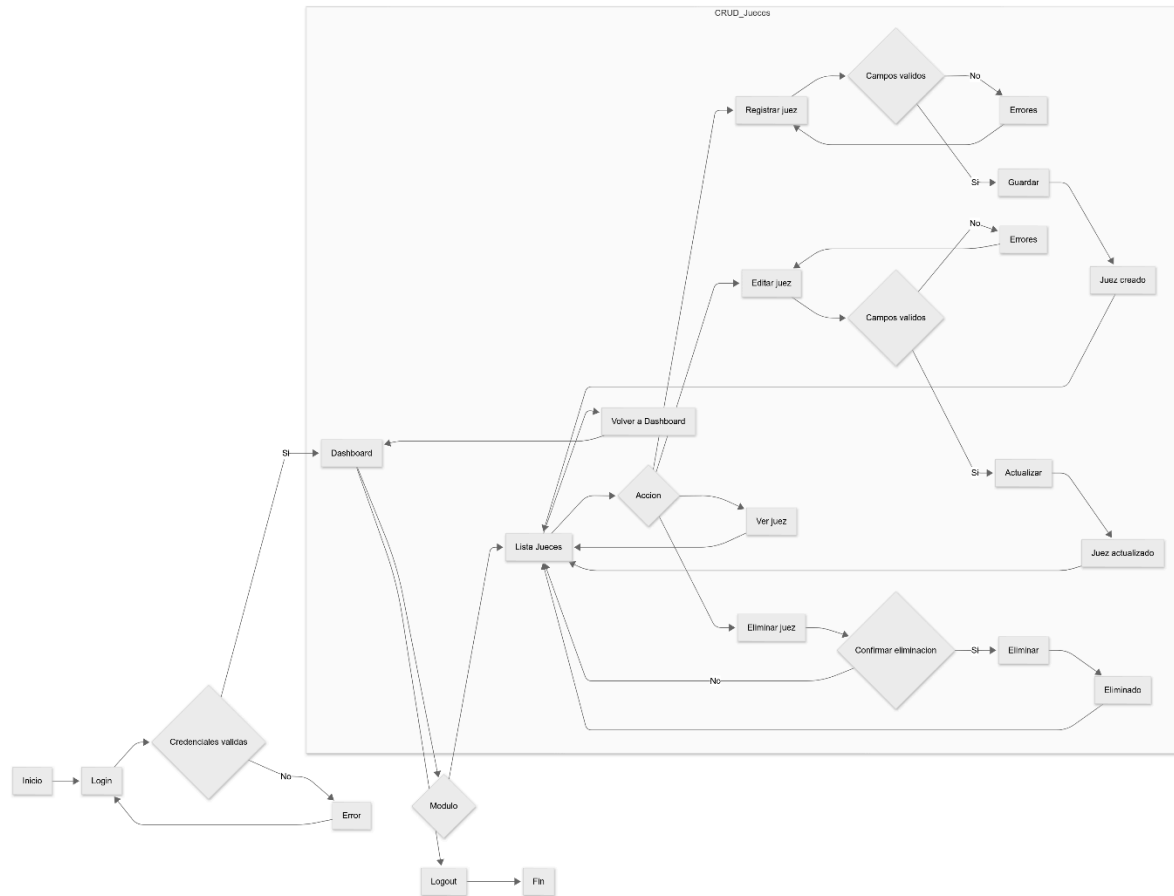


Figura 21. Diagrama de flujo CRUD jueces.
Fuente: Elaboración propia.

CRUD Jueces

Secuencia principal

1. **Ingreso y validación.** El usuario inicia sesión. Si las credenciales son válidas, el sistema redirige al *Dashboard* y, desde allí, al módulo de jueces; en caso contrario, se muestra un mensaje de error.
2. **Lista de jueces.** El sistema carga *Lista de jueces* como pantalla base; desde aquí se inician todas las acciones.
3. **Registrar juez.**
 - a) Se abre el formulario *Registrar juez*.
 - b) Se ejecuta la validación de campos.

- o **Sí:** el sistema guarda y notifica al Juez *creado*.
 - o **No:** el sistema muestra errores de validación y permite corregir.
 - c) Al finalizar, se retorna a *Lista de jueces* y, si el usuario lo decide, al *Dashboard*.
- 4. **Ver juez.** Desde la lista, la opción *Ver* muestra el detalle sin modificar y luego regresa a la lista.
- 5. **Editar juez.**
 - a) Se abre *Editar juez*.
 - b) Se valida nuevamente la integridad de los campos.
 - o **Sí:** el sistema actualiza y confirma *Juez actualizado*.
 - o **No:** el sistema presenta errores para corrección.
 - c) Se retorna a la lista y, opcionalmente, al *Dashboard*.
- 6. **Eliminar juez.**
 - a) Se elige *Eliminar* y se solicita *Confirmar eliminación*.
 - o **Sí:** el sistema elimina y muestra *Eliminado*.
 - o **No:** el sistema cancela y vuelve a la lista.
- 7. **Salida.** En cualquier punto, el usuario puede cerrar sesión (*Logout*) y finalizar.

Controles y reglas visibles

- Validación de campos obligatorios y formato de correo.
- Reglas de unicidad para identificador o correo.
- Control de rol y estado por evento para evitar conflictos de intereses.
- Confirmación de eliminación con impacto en evaluaciones asociadas.
- Trazabilidad de cambios y bloqueo de campos críticos cuando exista vínculo con evaluaciones.

Resultado esperado

- Gestión segura y consistente del registro, consulta, edición y eliminación de jueces.
- Prevención de conflictos de rol y mantenimiento de un historial auditable.
- Navegación clara con *Lista de jueces* como estado base.

El diagrama de flujo presentado en la Figura 22 representó la lógica completa del manejo de eventos dentro del aplicativo web, desde el inicio de sesión hasta las operaciones de creación, edición, visualización y eliminación. Este flujo permitió estructurar de manera clara las decisiones que debían tomarse en cada paso, validando campos, confirmando acciones críticas y orientando al usuario hacia las rutas adecuadas según su selección. Asimismo, el diagrama evidenció cómo el sistema gestionaba el acceso al módulo correspondiente una vez autenticado el usuario, garantizando que cada operación del CRUD de eventos se ejecutara siguiendo un recorrido coherente y controlado. Gracias a esta representación, fue posible comprender con precisión el comportamiento del aplicativo ante distintas acciones del usuario, asegurando que la interacción con los eventos se desarrollara de forma ordenada, segura y alineada con los requerimientos del proyecto.

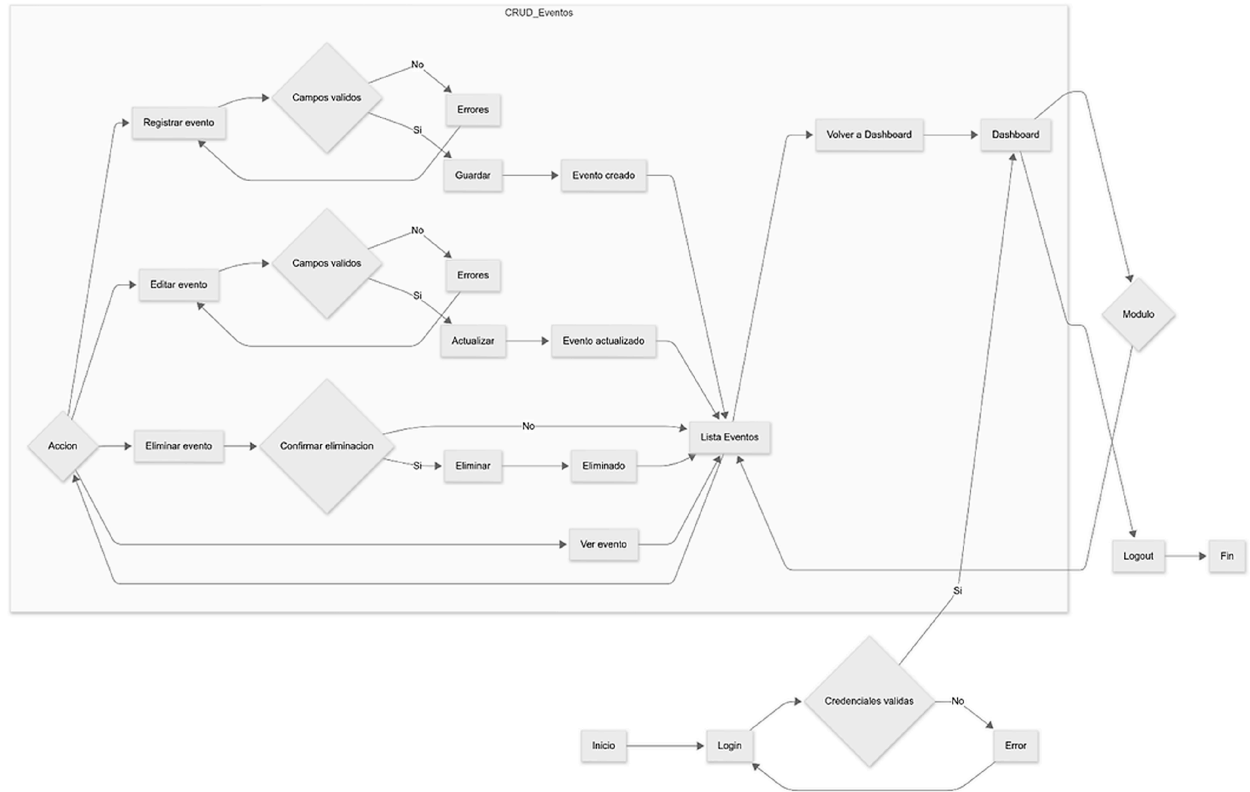


Figura 22. Diagrama de flujo CRUD eventos
Fuente: Elaboración propia.

CRUD Eventos

Secuencia principal

1. **Ingreso y validación.** El usuario inicia sesión. Si las credenciales son válidas, accede al *Dashboard* y, desde allí, al módulo de eventos; en caso contrario, el sistema muestra error.
2. **Lista de eventos.** El sistema carga *Lista de eventos* como pantalla base; desde aquí se inician todas las acciones.
3. **Registrar evento.**
 - a) Se abre el formulario *Registrar evento*.
 - b) Se ejecuta la validación de campos.
 - o **Sí:** el sistema guarda y notifica *Evento creado*.

- o **No:** el sistema muestra errores de validación y permite corregir.
- c) Al finalizar, se retorna a *Lista de eventos* y, si el usuario lo decide, al *Dashboard*.

4. Editar evento.

- a) Se abre *Editar evento*.
- b) Se valida nuevamente la integridad de los campos.
 - o **Sí:** el sistema actualiza y confirma *Evento actualizado*.
 - o **No:** el sistema presenta errores para corrección.
- c) Se retorna a la lista y, opcionalmente, al *Dashboard*.

5. Eliminar evento.

- a) Se elige *Eliminar evento* y se solicita *Confirmar eliminación*.
 - o **Sí:** el sistema elimina y muestra *Eliminado*.
 - o **No:** el sistema cancela y vuelve a la lista.

6. **Ver evento.** Desde la lista, la opción *Ver evento* muestra el detalle sin modificar y luego regresa a la lista.

7. **Salida.** En cualquier punto, el usuario puede cerrar sesión (*Logout*) y finalizar.

Controles y reglas visibles

- Validación de campos del evento: nombre, fechas, sede, tipo, estado, reglas de inscripción.
- Coherencia de fechas y no solapamiento cuando aplique.
- Reglas de cupo y fechas límite de inscripción.
- Consistencia con catálogos activos (divisiones, categorías, criterios).
- Control de visibilidad: evento público o propio de una academia.
- Mensajes de error, confirmación de eliminación y trazabilidad de cambios.

Resultado esperado

- Configuración de eventos segura y alineada con las reglas de negocio y de calendario.
- Protección de inscripciones y evaluaciones asociadas.
- Flujo predecible con *Lista de eventos* como estado base y operación trazable.

5.5.5 Arquitectura MVC

En cuarto lugar, el proyecto se organizó sobre la arquitectura Modelo–Controlador–Vista, por lo que el sistema resultó claro y ordenado. El Modelo conservaba la información esencial y aplicaba las reglas del dominio; por ejemplo, gestionaba usuarios, academias, eventos, participantes e inscripciones, además calculaba la división por edad y validaba la pertenencia activa antes de permitir un registro. El Controlador, a su vez, actuaba como puente, pues recibía la acción del usuario, verificaba los permisos según su rol y estado, coordinaba la operación en el Modelo y devolvía una respuesta consistente. Por su parte, la Vista presentaba formularios y listados para crear eventos, administrar miembros de academia, inscribir participantes y registrar calificaciones, todo mediante validaciones simples y mensajes claros que guiaban la interacción.

Como se muestra en la Figura 23, esta arquitectura permitió representar de forma visual cómo cada componente cumplía una función específica dentro del flujo general del sistema. En la ilustración se aprecia que la Vista interactuaba con los usuarios a través del navegador, mientras que el Controlador se encargaba de procesar las solicitudes y enlazarlas con las entidades internas, tales como eventos, participantes, academias, administradores y jueces. Asimismo, el Modelo agrupaba los elementos asociados a los datos y su lógica, lo que ayudaba a garantizar la integridad del sistema y la consistencia entre los procesos.

Gracias a esta organización, el sistema ganó mantenibilidad y transparencia. Si alguna regla cambiaba, se ajustaba en el Modelo y, en consecuencia, todas las pantallas la respetaban; por otra parte, si se modificaba un diseño, los datos permanecían íntegros. Además, el Controlador registraba acciones con sus fechas y reforzaba la coherencia del proceso, por lo que cada usuario

entendía qué podía hacer y en qué contexto. Finalmente, la Vista guiaba el trabajo diario y el Modelo garantizaba la calidad de la información, lo que permitió una evaluación confiable y una gestión ordenada de los eventos, asegurando que la plataforma funcionara de manera consistente en todos sus módulos.

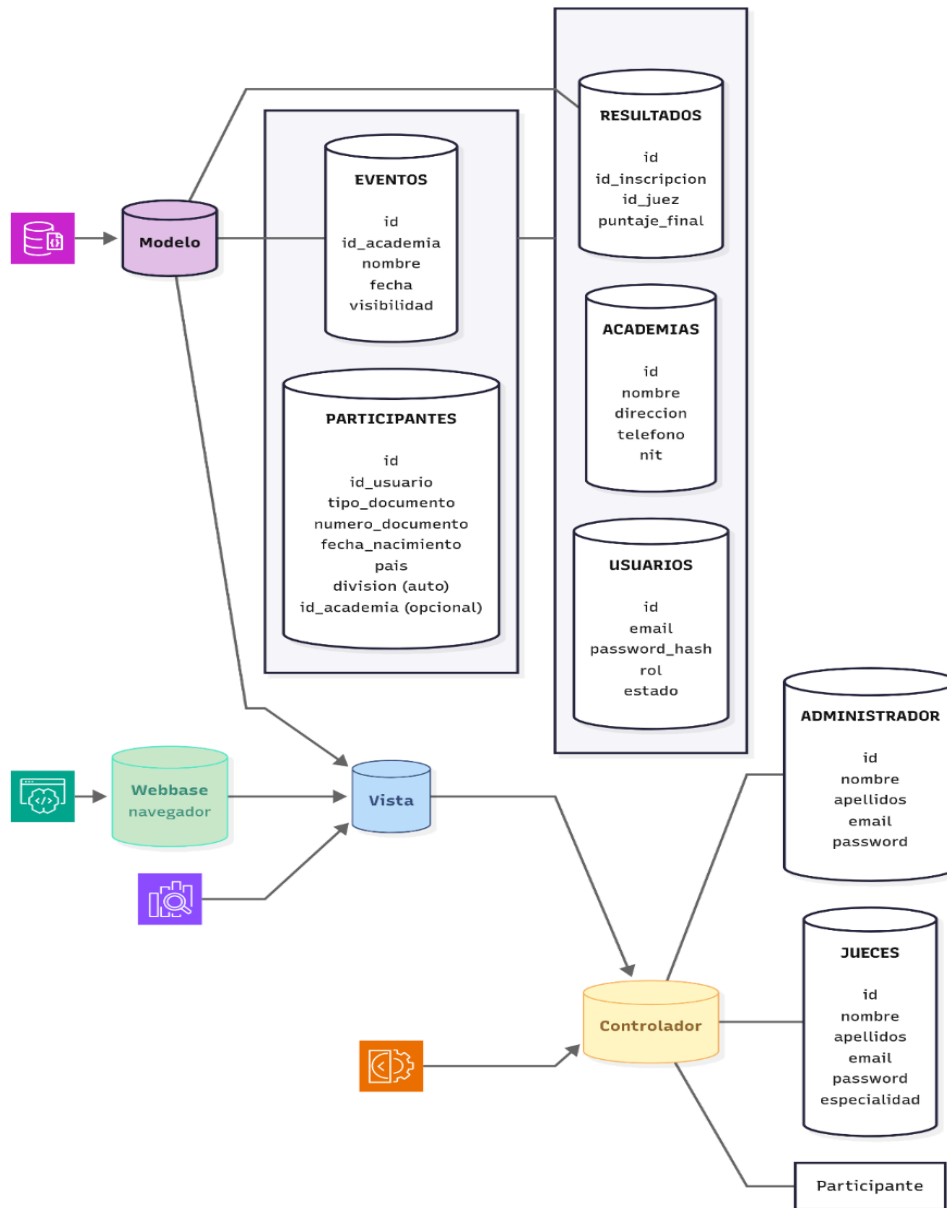


Figura 23. Prototipo modelo MVC (modelo- vista- controlador)
 Fuente: Elaboración propia.

Finalmente, en la etapa inicial del diseño arquitectónico, se elaboró un primer prototipo de diagrama de paquetes que permitiera organizar la aplicación en grupos funcionales y visualizar de manera general cómo interactuaban los diferentes componentes del sistema. Tal como se aprecia en la *Figura 24*, este esquema preliminar dividió la plataforma en módulos básicos de presentación, negocio y datos, con el fin de establecer una estructura inicial para la comunicación entre interfaces, servicios y acceso a la información. Aunque este modelo tenía un enfoque

introdutorio y no representaba aún la arquitectura final, resultó útil para comprender las responsabilidades principales de cada capa, anticipar posibles necesidades de integración y orientar las decisiones posteriores de refinamiento. Gracias a esta visualización temprana fue posible identificar relaciones, dependencias y límites entre módulos que más adelante se fortalecerían en la versión completa del diagrama.

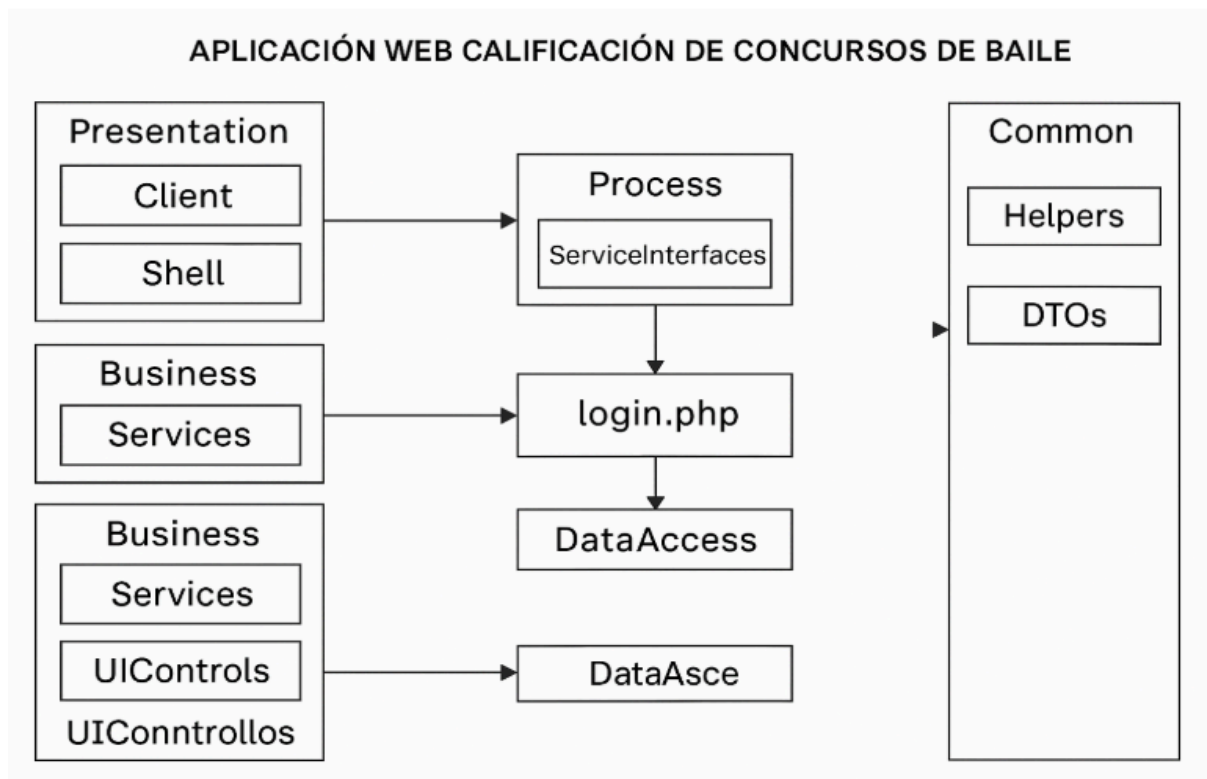


Figura 24. Prototipo modelo MVC (modelo- vista- controlador)
Fuente: Elaboración propia.

En la Figura 25 se presentó el diagrama de paquetes definitivo, el cual resumió de manera clara cómo quedó organizada la arquitectura del sistema después del proceso de refinamiento. En esta versión final se distinguieron las capas de presentación, negocio y datos, cada una con responsabilidades bien delimitadas que favorecieron el orden interno, la trazabilidad y el mantenimiento del software. La capa de presentación gestionó la interacción con el usuario mediante formularios, dashboards e interfaces visuales; la capa de negocio concentró la lógica

que coordinó procesos, aplicó validaciones y controló la ejecución de las reglas del sistema; y la capa de datos administró las operaciones de acceso, consultas y persistencia en la base de datos. Asimismo, se incorporaron paquetes de apoyo como la API, los modelos, la configuración y los servicios externos, que complementaron funcionalidades como pagos, notificaciones o manejo de sesiones. Gracias a esta estructura modular, cada componente cumplió un rol específico, lo que redujo errores, mejoró la escalabilidad y aseguró que el desarrollo permaneciera ordenado y comprensible para cualquier equipo técnico responsable de continuar con el proyecto.

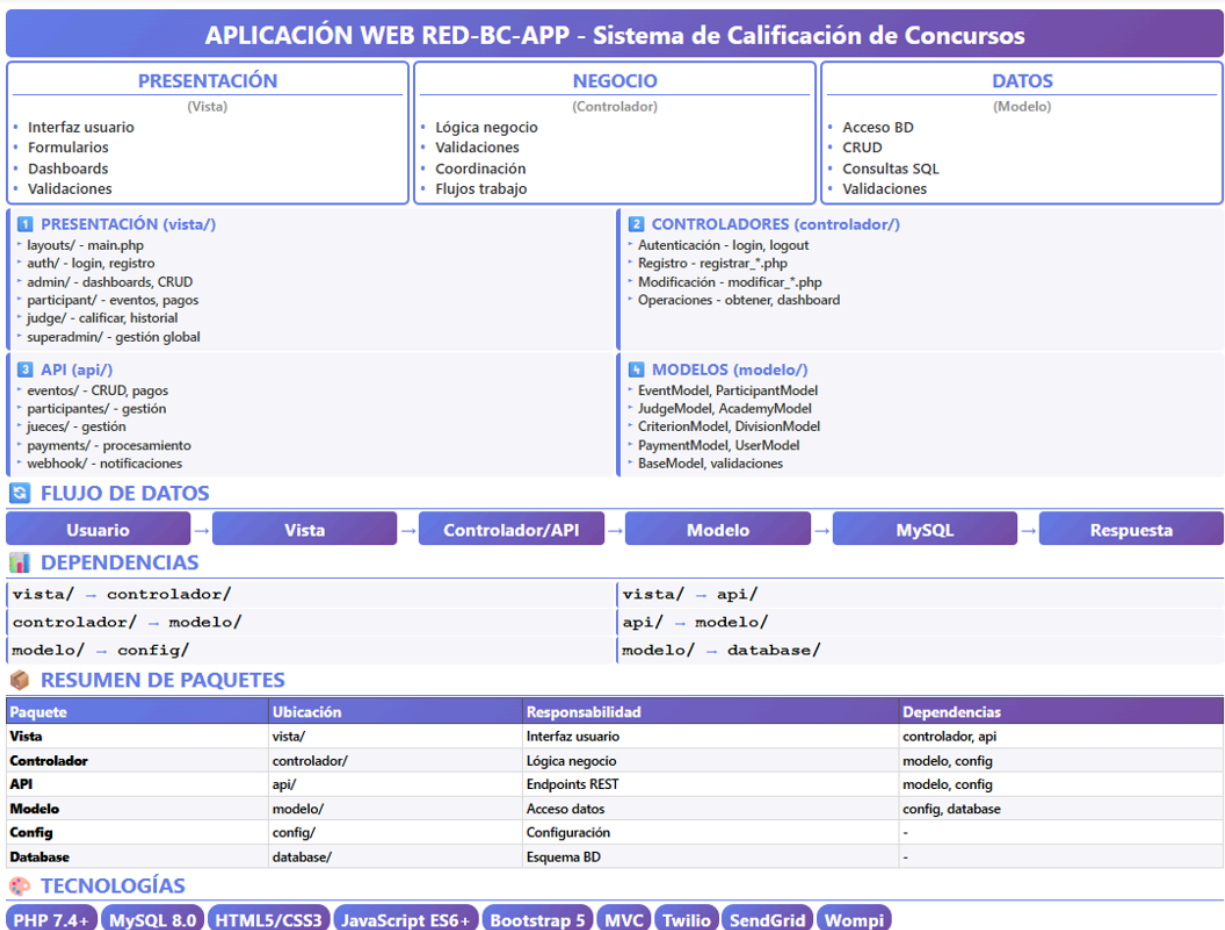


Figura 25. Diagrama de paquetes y dependencias del aplicativo web
Fuente: Elaboración propia.

En términos generales, la construcción de los diagramas UML representó una fase esencial dentro del diseño del sistema, ya que permitió depurar los requerimientos funcionales, validar las decisiones arquitectónicas y comprender de manera precisa el comportamiento previsto para cada módulo, validar las decisiones de arquitectura, anticipar posibles problemas de diseño y

asegurar que el sistema final resultara seguro, escalable y alineado con los objetivos planteados en el proyecto.

5.5.6 Población y muestra

La población considerada en este proyecto correspondió a los potenciales usuarios del sistema web de calificación de concursos de baile, agrupados según los roles definidos dentro de la plataforma: participantes (tanto de academia como independientes), jueces, administradores de academia y el superadministrador que dirige la operación global del sistema. Esta población reflejaba con fidelidad los perfiles involucrados en un concurso real, dado que comprendía desde usuarios jóvenes que iniciaban su formación artística hasta adultos con experiencia en procesos culturales y de evaluación. En este marco, la amplitud etaria coincidía con el rango permitido por la plataforma, que contemplaba participantes entre los 5 y los 100 años y jueces mayores de edad a partir de los 18 años.

La muestra se seleccionó mediante un muestreo intencionado y estuvo conformada por cinco personas cuyas edades oscilaron entre los 16 y los 60 años. Este rango permitió representar de manera adecuada la diversidad de usuarios que interactúan con el sistema y, al mismo tiempo, evaluar su comportamiento en condiciones cercanas a un escenario real. Cada integrante de la muestra realizó la prueba de forma autónoma, con una contextualización inicial sobre el propósito del ejercicio y acceso al manual correspondiente a su rol, sin incluir instrucciones detalladas para completar las tareas. Esta decisión metodológica permitió observar la capacidad del sistema para guiar al usuario mediante su diseño, navegación y mensajes internos.

Para la evaluación se asignó un rol distinto a cada persona con el fin de cubrir el funcionamiento general del Producto Mínimo Viable. El asesor técnico participó con el perfil de superadministrador, lo que le permitió explorar la totalidad del sistema: academias, participantes independientes, participantes de academia, jueces, eventos, divisiones, criterios, reportes globales, ganadores y manuales. Los demás usuarios accedieron únicamente a las secciones vinculadas con sus permisos, lo cual facilitó validar tanto la experiencia de cada rol como la correcta delimitación de opciones en los menús y el aislamiento entre módulos.

Durante las sesiones se asignaron tareas específicas que correspondían a las acciones clave de cada rol, tales como inscribirse a un evento, calificar participantes, actualizar información personal, crear registros, realizar pagos, filtrar eventos, consultar reportes o publicar actividades. Esto permitió evaluar aspectos esenciales en pruebas de usuario, entre ellos que el flujo de uso sea intuitivo, la claridad de la interfaz, la adecuación del contenido visual, la coherencia entre pantallas y la facilidad para identificar botones, formularios e indicadores de estado. Asimismo, se analizaron factores relacionados con la velocidad de respuesta, la estabilidad de las funciones CRUD, la comprensión de los mensajes del sistema, la accesibilidad para personas con diferente nivel de experiencia digital y la percepción de confianza durante la navegación.

El instrumento utilizado para registrar la experiencia de cada participante fue un formulario de evaluación estructurado en cinco secciones que incluían acciones asignadas, criterios funcionales, elementos visuales y de usabilidad, valoración global y observaciones abiertas. Este formulario incorporaba un encabezado con información fundamental para el análisis, incluyendo la edad del usuario, su profesión, su nivel de experiencia digital o tecnológica, el conocimiento previo respecto a aplicaciones de concursos y el rol evaluado. Estos datos permitieron interpretar con mayor precisión la variabilidad de la muestra y contextualizar de manera adecuada los resultados obtenidos. El anexo A y B, presentan el formato utilizado para el rol de superadministrador, el cual se tomó como referencia general por su alcance más amplio dentro del sistema y por incluir todos los módulos disponibles para la validación.

En conjunto, la muestra permitió observar el desempeño del sistema desde perspectivas diversas, considerando tanto usuarios con experiencia digital alta como participantes que enfrentaban por primera vez una herramienta de este tipo. Esta variabilidad aportó una visión completa para evaluar la claridad del diseño, la facilidad de uso, la estabilidad operativa y la coherencia funcional del prototipo. De esta manera, las pruebas realizadas proporcionaron una base sólida para validar el Producto Mínimo Viable sin modificar su alcance y sin incorporar nuevas funciones, cumpliendo así el objetivo metodológico planteado para esta etapa del proyecto.

5.5.7 Mockups y Wireframes

Durante esta fase se elaboraron los wireframes y mockups que sirvieron como guía visual para la construcción del aplicativo web. Estas representaciones permitieron anticipar la distribución de los elementos en pantalla, definir la estructura de navegación y validar la coherencia entre los diferentes módulos antes de avanzar al desarrollo. Además, facilitaron la comunicación con el asesor técnico, ya que mostraban de manera sencilla y comprensible cómo se esperaba que funcionaran procesos clave como el inicio de sesión, la calificación, la administración de eventos y la visualización de resultados. Gracias a estas maquetas, diseñadas en herramientas como Figma, fue posible detectar ajustes necesarios en la interfaz y asegurar que las pantallas respondieran a los requerimientos funcionales establecidos para cada rol del sistema.

Interfaz de Inicio de Sesión

Pantalla de acceso al sistema de concursos de baile. Permite a administradores, jueces y participantes ingresar con su correo y contraseña para acceder a sus respectivos módulos conocidos como dashboard. La figura 26 ilustra la vista de ingreso y validación de credenciales de roles y entrada segura a la plataforma.



Figura 26. Interfaz de inicio de sesión.
Fuente: Elaboración propia (2025).

Interfaz de Registro de Participantes desde el dashboard del rol administrador academia

Como se ilustra en la figura 27 es la vista donde se despliega el formulario destinado a registrar participantes del sistema de concursos de baile, pertenecientes a una academia de baile. Permite capturar datos personales, información de contacto, división, estado de membresía como usuario perteneciente a la academia.

The image shows a web browser window with the address bar displaying "http://localhost/app/registro". The main content is a registration form titled "Registro de Participante". The form is organized into two columns of input fields. The left column contains: "Nombre completo *" with a placeholder "Escribe tu nombre"; "Contraseña *" with a placeholder "Entre 6 y 15 caracteres"; "Número de documento *" with a placeholder "Cédula o ID"; and "Evento *" with a placeholder "Selecciona un evento". The right column contains: "Email *" with a placeholder "ejemplo@correo.com"; "Confirmar contraseña *" with a placeholder "Repite la contraseña"; "Fecha de nacimiento" with a placeholder "aaaa-mm-dd"; and "División *" with a placeholder "Selecciona una división". At the bottom of the form, there is a note: "Campos marcados con * son obligatorios."

Figura 27. Formulario de registro de participante desde la interfaz de administrador de academia
Fuente: Elaboración propia (2025).

Interfaz de Registro de Participantes independientes desde el módulo de Login

Formulario redireccionado desde la vista de inicio de sesión, destinado a nuevos participantes del sistema de concursos de baile, que no pertenecen a ninguna academia. Permite capturar datos personales, información de contacto, categoría, división, y la inscripción a eventos públicos o por invitación como muestra la figura 28, adicional almacena los datos asociados a un evento específico para habilitar la participación en el concurso.

The image shows a registration form titled "Registro de Participante" with the subtitle "Únete a Red BC como participante". The form is divided into two main sections: "Información Personal" and "Información de Participante".

Información Personal

- Nombre completo * (input field with 'xxx')
- Correo electrónico * (input field with 'xxx')
- Contraseña * (input field with 'xxx')
- Confirmar contraseña * (input field with 'xxx')

Información de Participante

- Número de documento * (input field with 'xxx')
- Fecha de nacimiento * (input field with 'xxx')
- País (input field with 'xxx')
- Teléfono * (input field with 'xxx')
- Ciudad (input field with 'xxx')
- Dirección (input field with 'xxx')
- Género (input field with 'xxx')
- Tipo de documento * (input field with 'xxx')

Acepto términos de uso y tratamiento de datos

Crear cuenta

Mockup b/n · baja/mediana fidelidad · campos con 'xxx'

Figura 28. Formulario de registro de participantes individuales.
Fuente: Elaboración propia (2025).

Interfaz Dashboard administrador de academias para gestionar participantes.

La figura 29 ilustra la vista que contiene el menú para gestionar participantes para el rol administrador de academias y super administrador global y diseñada para visualizar, editar o eliminar la información personal del participante y su estado de actividad.

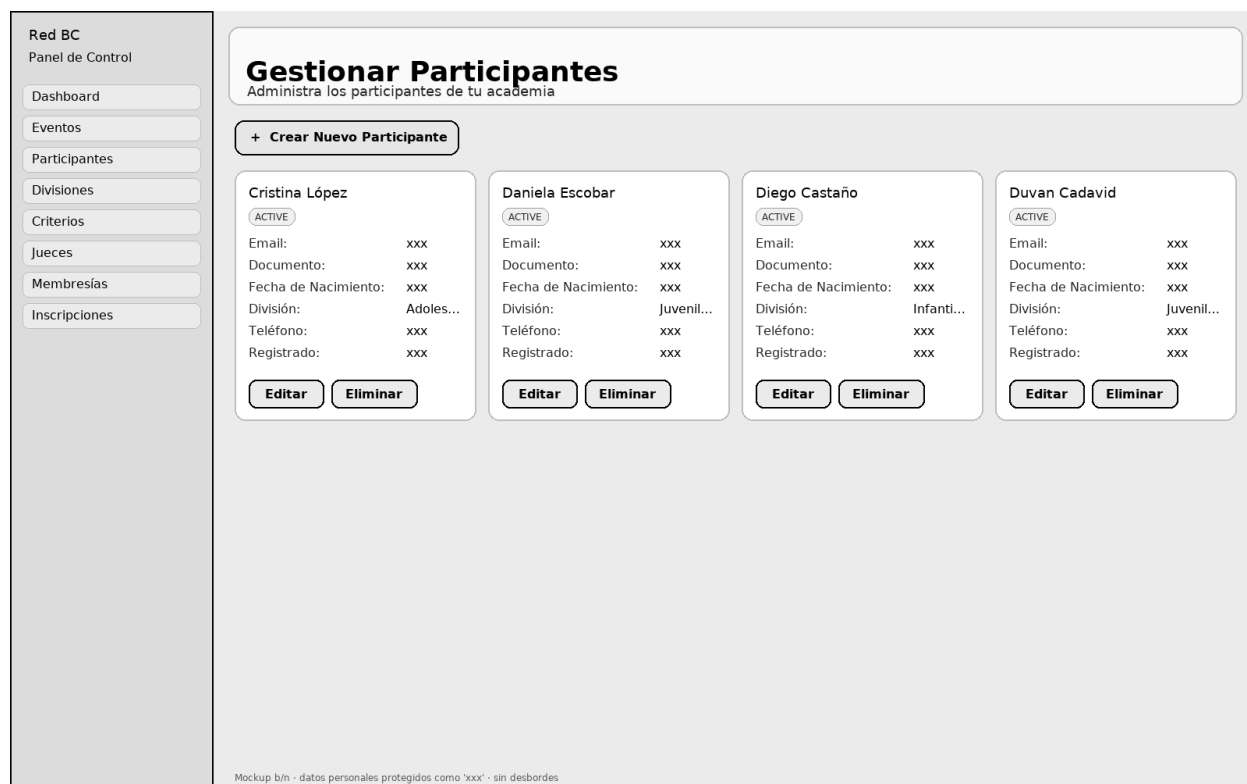


Figura 29. Interfaz de gestión y listado de participantes.
Fuente: Elaboración propia (2025).

Interfaz de Registro de Academias

Vista diseñada para registrar academias participantes en el sistema de concursos de baile. Como presenta la figura 30, el formulario de la interfaz recoge información personal y propia de la academia. Permite modificar y eliminar academia, editar el estado de la academia y su actividad, también vincular academias a eventos y mantener un control estructurado de las entidades responsables de los participantes.

Red BC
Super Administrador Global
SUPER ADMIN

- Dashboard
- Gestionar Academias
- Gestionar Usuarios
- Ver Todos los Eventos
- Ver Todos los Participantes
- Ver Todos los Jueces
- Divisiones
- Criterios
- Todas las Inscripciones
- Mi Perfil

+ Crear Nueva Academia

[Volver](#)

Información de la Academia

Nombre de la Academia * NIT *

Email *

Teléfono *

Dirección *

Tipo de Plan * Estado *

Información de Contacto

[Cancelar](#) [Crear Academia](#)

Mockup bin · botones ajustados dentro del contenedor

Figura 30. Interfaz de registro de academias
Fuente: Elaboración propia (2025).

Interfaz Dashboard super administrador para gestionar academias, eventos, participantes y jueces

La figura 31 detalla gráficamente la vista que contiene el menú para gestionar academias, usuarios administradores, participantes, eventos y jueces con todos los permisos para el rol super administrador global y diseñada para visualizar, editar o eliminar la información de estos menús, este módulo redirige a los formularios que permite editar, eliminar, cambiar el estado de cada una de las opciones del menú.

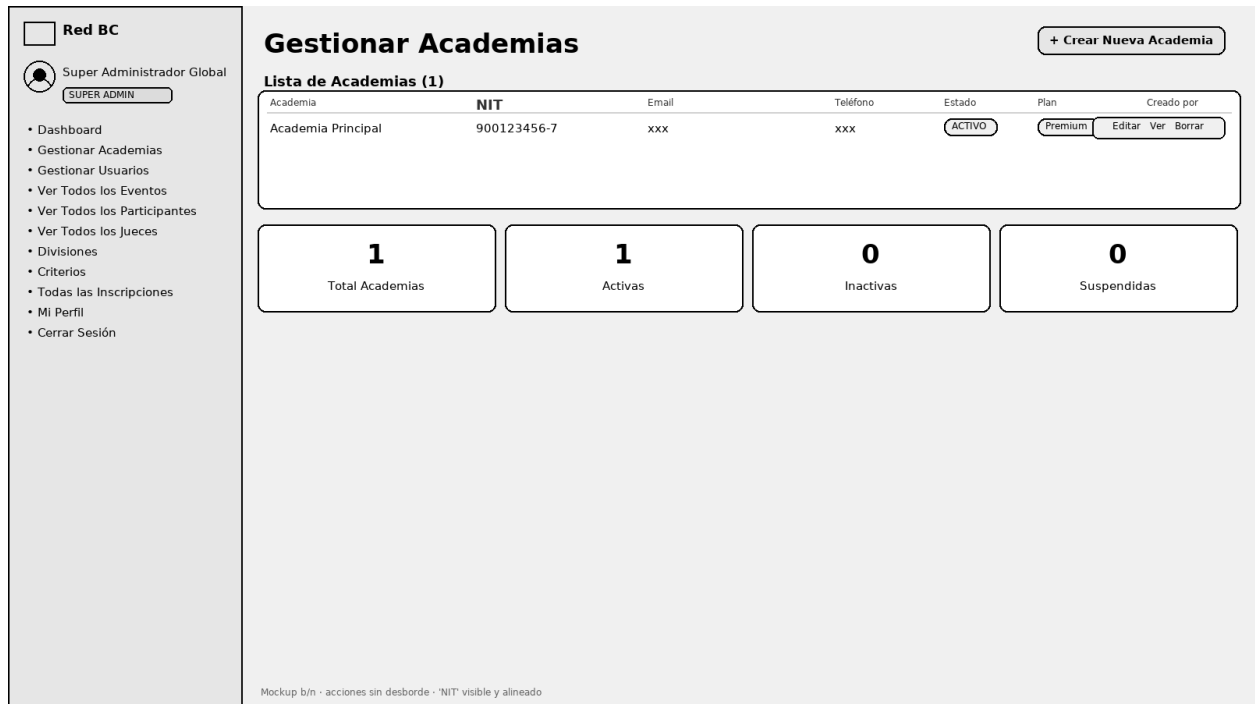


Figura 31. Interfaz de gestión y listado de academias.
Fuente: Elaboración propia (2025).

Interfaz de Registro de Eventos

Este formulario está habilitado para el dashboard y rol con permisos de administrador de academias y super administrador global, el cual permite crear nuevos eventos ingresando datos como nombre, ciudad, fecha del evento, inicio y fin de inscripciones, estado, visualización, jueces activos para el evento es decir si solo lo pueden ver los participantes pertenecientes a alguna academia o si es para todo público, o es un borrador creado de manera momentánea. Su diseño simplificado representado en la figura 32 facilita la gestión eficiente de múltiples competencias de baile, garantizando que el evento quede correctamente vinculado a una academia y disponible para participantes y jueces.

The image shows a web interface titled "Gestión de Eventos" with a sub-modal window titled "Crear Nuevo Evento". The form contains the following fields:

- Nombre del Evento *
- Fecha del Evento *
- Inicio de Inscripciones
- Fin de Inscripciones
- Ubicación *
- Visibilidad *
- Política de Inscripción *
- Estado Inicial *

Each field contains the placeholder text "xxx". At the bottom of the form, there is a note: "Nota: El año de edición se calculará automáticamente con base en la fecha del evento. Puedes cambiar el estado del evento después de crearlo, si es necesario." and two buttons: "Cancelar" and "Crear Evento".

Figura 32. Formulario de registro de eventos.
Fuente: Elaboración propia (2025).

Interfaz de registro de criterios

Este formulario está habilitado para el dashboard y rol con permisos de administrador de academias y super administrador global. La figura 33 detalla de forma gráfica el formulario el cual permite crear nuevos criterios con peso, puntaje y nivel de puntaje de primer, segundo o tercer nivel para facilitar el proceso de desempate de participantes con puntuaciones similares dentro de la base de datos y ser asignados a los eventos que se requieran.



Crear Nuevo Criterio

Nombre del Criterio * Peso (%) *

Descripción

Estado *

Mockup: 0711 - sin recordos - placeholders y botones con ancho fijo

Figura 33. Formulario de registro de criterios de evaluación.
Fuente: Elaboración propia (2025).

Interfaz de registro de divisiones

Este formulario está habilitado para el dashboard y rol con permisos de administrador de academias y super administrador global. La figura 34 ilustra el formulario el cual permite crear nuevas divisiones que no se superpongan a los rangos predefinidos en las divisiones preestablecidas dentro de la base de datos y ser asignados a los eventos que se requieran.

El formulario, titulado "Crear Nueva División", contiene los siguientes campos:

- Nombre de la División ***: Campo de texto con el placeholder "xxx".
- Edad Mínima ***: Campo de texto con el valor "0". Debajo del campo se indica "Mínimo: 1 año".
- Edad Máxima ***: Campo de texto con el valor "100". Debajo del campo se indica "Máximo: 100 años".
- Descripción**: Campo de texto con el placeholder "Opcional...".
- Estado ***: Campo de selección con el placeholder "Seleccionar estado".

En la parte inferior derecha del formulario hay dos botones: "Cancelar" y "Crear División". En la parte inferior izquierda, se encuentra el texto "Mockup b/n - sin desbordes - inputs con placeholders".

Figura 34. Formulario de registro de divisiones para participantes en eventos.
Fuente: Elaboración propia (2025).

Interfaz Dashboard administrador de academias para gestionar eventos

Esta vista está habilitada para el dashboard y rol con permisos de administrador de academias y super administrador, La figura 35 evidencia el formulario el cual permite visualizar, editar, eliminar la información del evento con respecto a fecha del evento, de inicio y cierre de inscripción, modificar su estado de actividad, aforo, su visibilidad, política de visibilidad y privacidad y modalidad si es pago o gratuito y los jueces que harán parte del evento.

Este formulario está habilitado para el dashboard y rol con permisos de administrador de academias y super administrador global. La figura 34 ilustra el formulario el cual permite crear nuevas divisiones que no se superpongan a los rangos predefinidos en las divisiones preestablecidas dentro de la base de datos y ser asignados a los eventos que se requieran.



















Gestión de Eventos							Volver al Dashboard	+ Crear Evento
Eventos de la Academia								
Nombre	Fecha	Ubicación	Estado	Visibilidad	Inscripciones	Acciones		
Competencia de Ballet Clásico	10/12/2025	Teatro Nacional	REG OPEN	Solo Acade...	15/09/2025 - 01/12/2025	  		
Festival Juvenil de Danza	30/11/2025	Centro Cultural	REG OPEN	Público	10/09/2025 - 20/11/2025	  		
Festival de Danza Nacional	15/11/2025	Teatro Colón	REG OPEN	Público	01/10/2025 - 11/11/2025	  		
Competencia Regional de Baile	20/10/2025	Centro de Convenciones	REG OPEN	Público	01/09/2025 - 15/10/2025	  		
Gala de Danza Contemporánea	15/10/2025	Teatro Municipal	REG OPEN	Público	06/09/2025 - 10/10/2025	  		
Encuentro de Academias	30/09/2025	Auditorio Principal	REG OPEN	Solo Acade...	06/09/2025 - 25/09/2025	  		

Figura 35. Interfaz de gestión, CRUD y listado de eventos.
Fuente: Elaboración propia (2025).

Interfaz de Registro de Jueces

Este formulario está habilitado para el dashboard y rol con permisos de administrador de academias y super administrador global, el cual permite crear nuevos eventos ingresando datos personales y estado de actividad en su rol de juez para impartir calificaciones en eventos o registrarse como participante en eventos de bailes. Su diseño simplificado facilita la gestión eficiente de cada juez con un estado claro (activo como juez o participante), así no hay confusiones ni sesgos dentro del proceso de calificación o de registro como participante a un evento. La figura 36 detalla visualmente el formulario donde se ingresan datos personales, especialidad de géneros musicales del juez y la experiencia en el campo.

El juez queda vinculado al evento y a la academia que lo invita, y solo evalúa las categorías/divisiones que se le asignan y las notas se registran con criterios definidos, por eso son comparables y trazables. Si el juez quiere competir, pasa por inscripción como participante, separado de su rol de juez.

Gestionar Jueces

Crear Nuevo Juez [X]

Nombre Completo *
xxx

Email *
xxx

Teléfono *
xxx

Especialización
Seleccionar...

Contraseña *
xxx

Confirmar Contraseña *
xxx

Estado *
Seleccionar estado

Los datos se guardarán con cifrado básico en esta etapa del prototipo.

Cancelar Crear Juez

Figura 36. Formulario de registro de jueces.
Fuente: Elaboración propia (2025).

Interfaz Dashboard jueces para gestionar calificaciones en eventos

Vista diseñada para visualizar el listado de participantes por evento y su información personal, asignar puntajes sobre criterios por divisiones de participantes en eventos, ver el historial de calificación de participantes filtrado por academia y división que se evaluó, generar reportes en pdf o formato xls y visualizar las inscripciones realizadas con el estado activo como juez a eventos e inactivo como participante de eventos como representa visualmente la figura 37.

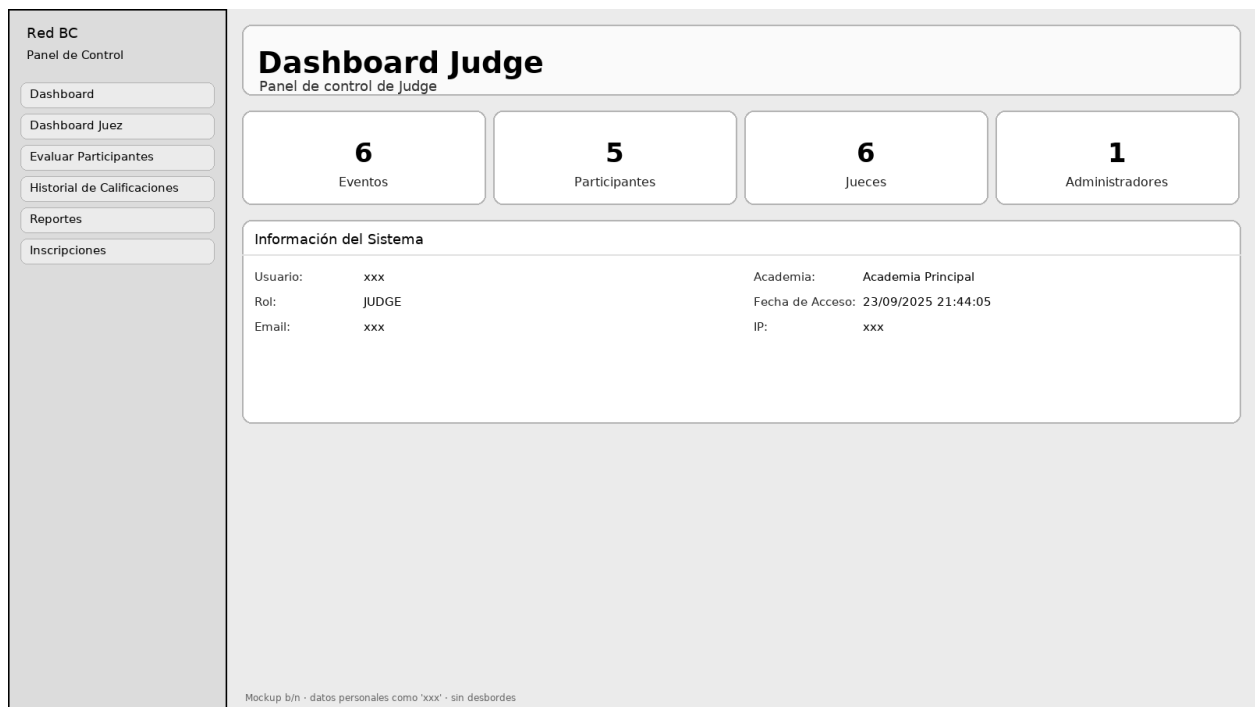


Figura 37. Interfaz de gestión de jueces y calificación de participantes.
Fuente: Elaboración propia (2025).

Interfaz de Resultados

La figura 38 expone de manera visual la interfaz de resultados está dirigida al rol de jueces activos, el cual permite visualizar únicamente las calificaciones por criterios, división y evento que han asignado cada uno de forma individual sin tener conocimiento de las calificaciones impartidas por otros jueces asignado al evento, asegurando imparcialidad en el proceso. Al mismo tiempo, el administrador tiene acceso al consolidado general de puntajes. Facilita la validación de resultados y la posterior publicación de ganadores de forma clara y controlada.

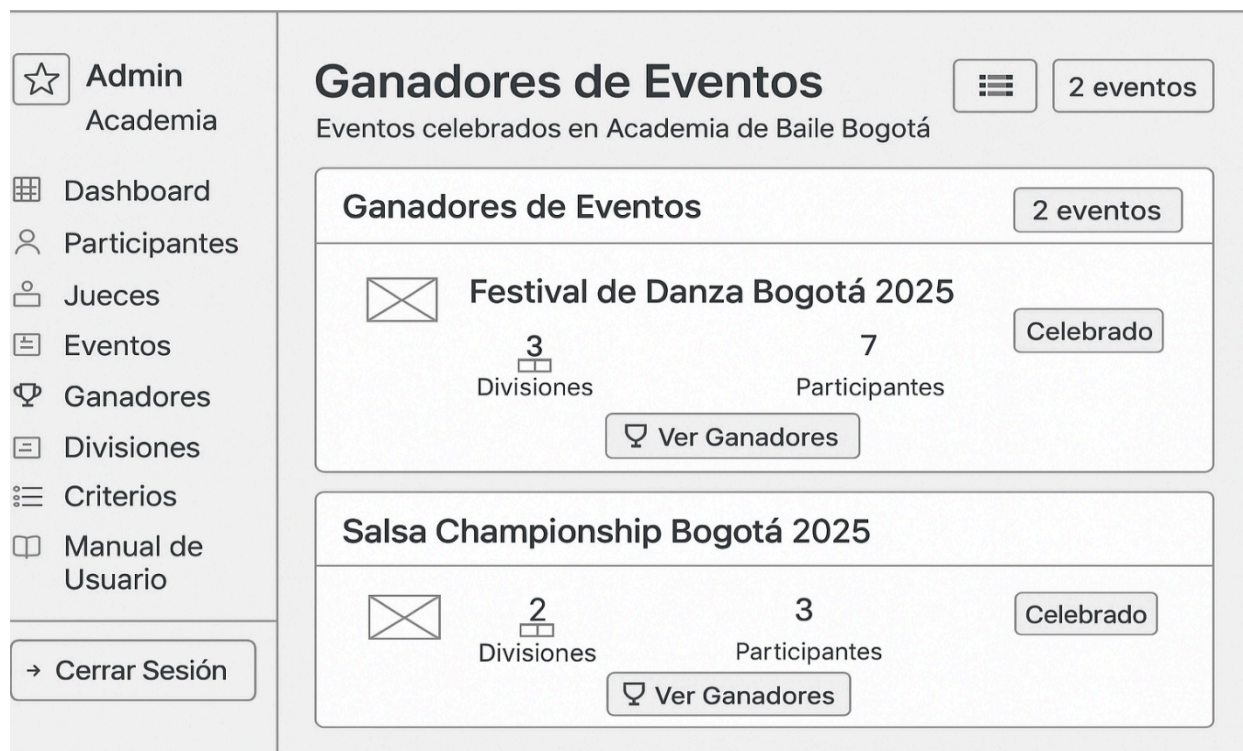


Figura 38. Interfaz de gestión y listado de resultados.

Fuente: Elaboración propia (2025).

Interfaz dashboard Administrador academia del Módulo Ganadores

La figura 39 permite observar la vista está diseñada exclusivamente para los roles de super administrador o administrador de academias. Presenta en tiempo real el listado de participantes con los puntajes consolidados, ordenados de mayor a menor, permitiendo identificar fácilmente a los ganadores por división y categoría.

El sistema actualiza automáticamente los resultados conforme se registran y consolidan las calificaciones emitidas por los jueces, garantizando una visualización precisa y transparente del ranking final, los ganadores se muestran una vez todos los jueces asignados al evento hayan deliberado su calificación completa a cada participante por división en cada evento.

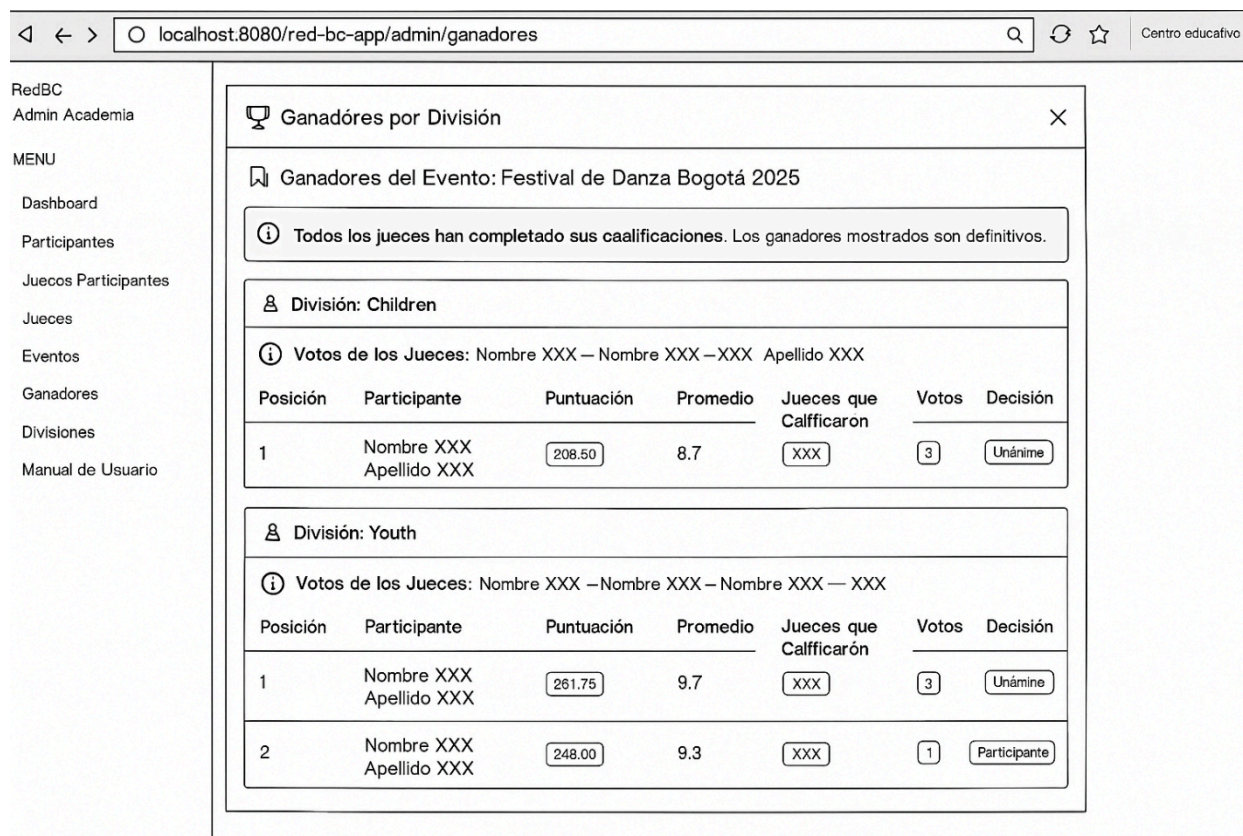


Figura 39. Interfaz de gestión y listado de ganadores por división en eventos.
Fuente: Elaboración propia (2025).

Interfaz Historial de calificaciones

Esta vista está dirigida al rol de jueces activos y permite a los jueces visualizar únicamente las calificaciones que solo él cómo juez ha asignado a los participantes en las divisiones y en los eventos que estuvo activo como rol de juez, asegurando imparcialidad en el proceso. Como se ejemplifica en la figura 40 la vista puede filtrar la búsqueda por fecha de evento o por nombre o identificación del participante.

La figura 39 permite observar la vista está diseñada exclusivamente para los roles de super administrador o administrador de academias. Presenta en tiempo real el listado de participantes

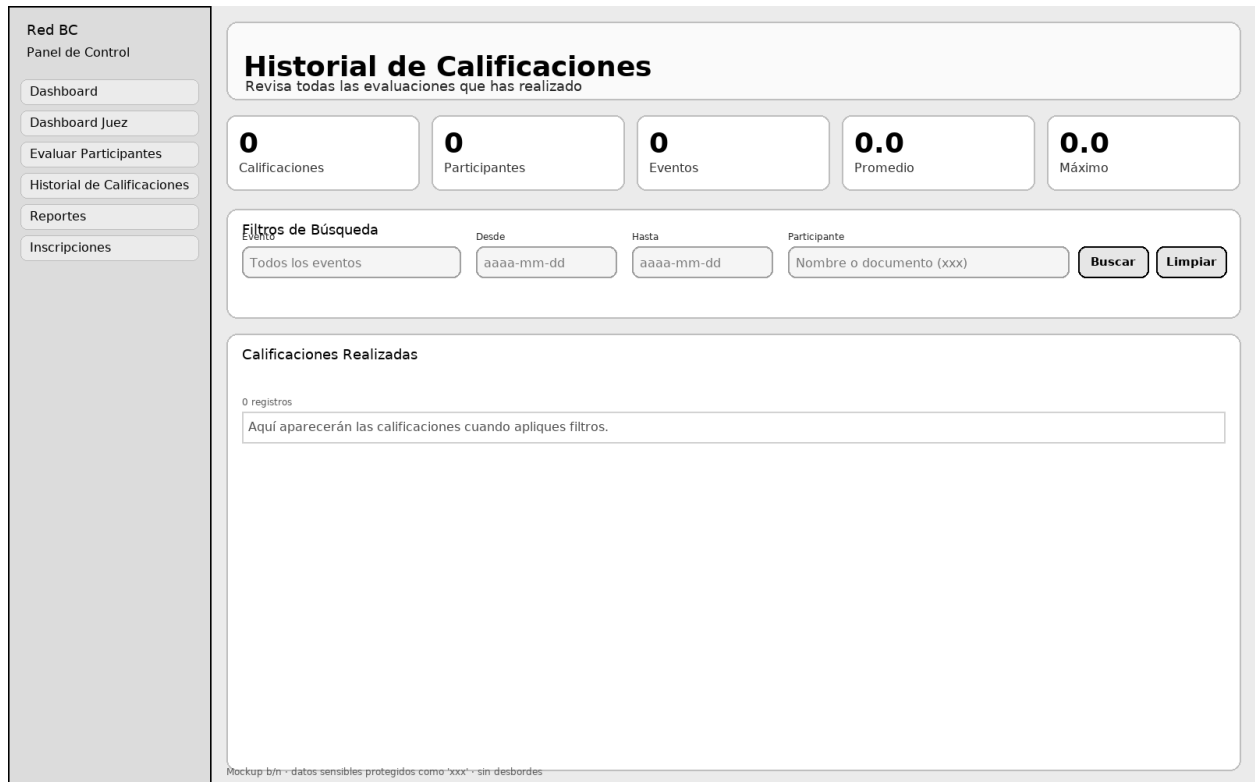


Figura 40. Interfaz de gestión y listado de calificaciones.
Fuente: Elaboración propia (2025).

5.6 Fase 4: Arquitectura

En esta fase se definió la estructura interna del aplicativo web y se organizó el proyecto siguiendo un orden lógico que facilitara su comprensión y mantenimiento. Para ello, se adoptó una organización por carpetas donde se separaron los archivos según la función que cumplían dentro del sistema. Esta distribución permitió diferenciar los elementos responsables de mostrar la interfaz, los encargados de procesar la lógica del sistema y los componentes destinados a manejar la información almacenada. Como se observa en la Figura 41, esta estructura ofreció una visión clara de cómo estaban organizados los archivos principales del proyecto.

Adicional, se estableció una arquitectura basada en una separación por capas que permitió gestionar adecuadamente la lógica del negocio, la interacción con la base de datos y la presentación visual. El aplicativo fue desarrollado en PHP dentro de Visual Studio Code, lo que facilitó la integración con herramientas de depuración, extensiones para el manejo de código y

control de versiones. En esta arquitectura, la capa de presentación estuvo conformada por archivos HTML, CSS y plantillas reutilizables que definieron la interfaz del sistema; la capa lógica incluyó los módulos encargados de procesar operaciones como el registro de calificaciones, la validación de usuarios y la administración de eventos; y la capa de datos incorporó las conexiones hacia la base de datos, los métodos para realizar consultas SQL y los controles para garantizar la integridad de la información. Esta organización también incluyó carpetas específicas para servicios, configuraciones, librerías y componentes reutilizables, lo que permitió mantener un proyecto ordenado, escalable y fácil de modificar. Gracias a esta arquitectura, el sistema pudo gestionar correctamente las operaciones de los diferentes roles, mejorar la mantenibilidad y asegurar que el flujo de datos desde la interfaz hasta la base de datos se ejecutara de manera consistente y segura.

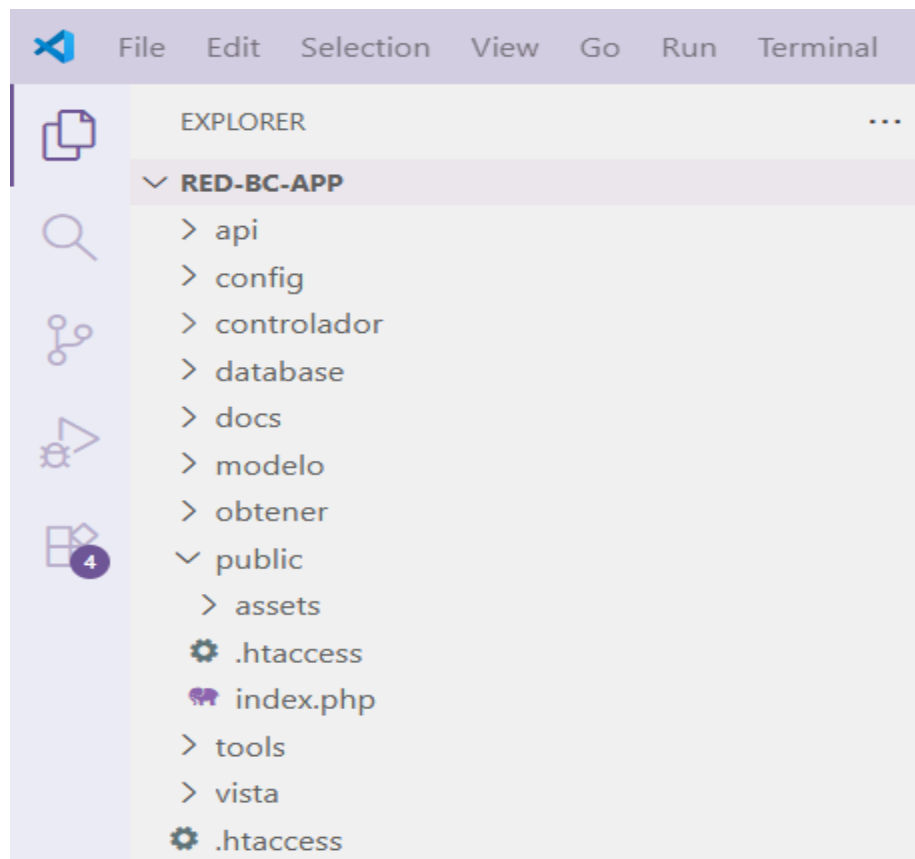


Figura 41. Mapa de archivos de la aplicación y su separación por capas.
Fuente: Elaboración propia.

De manera complementaria, la figura 42 ilustra cómo la arquitectura incluyó directorios destinados a configuraciones, recursos compartidos, manejo de sesiones, validaciones, hojas de estilo, scripts y plantillas reutilizables. Esta separación permitió que el desarrollo fuera más seguro y que los cambios pudieran aplicarse sin afectar el funcionamiento general del sistema. Así mismo, esta organización facilitó la identificación rápida de errores, la actualización de componentes y la incorporación de mejoras, dado que cada parte del sistema se encontraba ubicada en un lugar específico y cumplía una función claramente definida.

The image shows a code editor interface. On the left, the 'EXPLORER' pane displays the directory structure of a project named 'RED-BC-APP'. The structure includes folders for 'api', 'controlador', 'modelo', 'public', and 'vista', along with files like '.htaccess', 'bd_baile.sql', 'DIAGRAMA_FLUJO_COMPLETO.md', 'DOCUMENTACION_SISTEMA.md', 'index.php', and 'REGLAS_MEMBRESIA_DINAMICA.md'. On the right, the editor shows the code for 'crear_criterio.php'. The code is a PHP script that starts with a session, checks if the user is an administrator, includes a connection file, and processes a form submission with validation. The terminal at the bottom shows the command prompt 'PS C:\xampp\htdocs\red-bc-app>'.

```

1 <?php
2 session_start();
3
4 // Verificar que sea Admin Academia
5 if (!isset($_SESSION['user_id']) || $_SESSION['rol'] !== 'ADMIN') {
6     header('Location: ../login/login.php');
7     exit();
8 }
9
10 // Incluir archivo de conexión
11 require_once '../modelo/conexion.php';
12
13 $academy_id = $_SESSION['academy_id'] ?? null;
14
15 if (!$academy_id) {
16     $_SESSION['mensaje'] = 'No tienes academia asignada';
17     $_SESSION['tipo_mensaje'] = 'danger';
18     header('Location: vista_dashboard.php');
19     exit();
20 }
21
22 // Procesar el formulario
23 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
24     $nombre = trim($_POST['nombre']);
25     $descripcion = trim($_POST['descripcion']);
26
27     $errores = [];
28
29     // Validaciones
30     if (empty($nombre) || strlen($nombre) > 100) {
31         $errores[] = 'El nombre del criterio es obligatorio y debe tener máximo 100 caracteres';

```

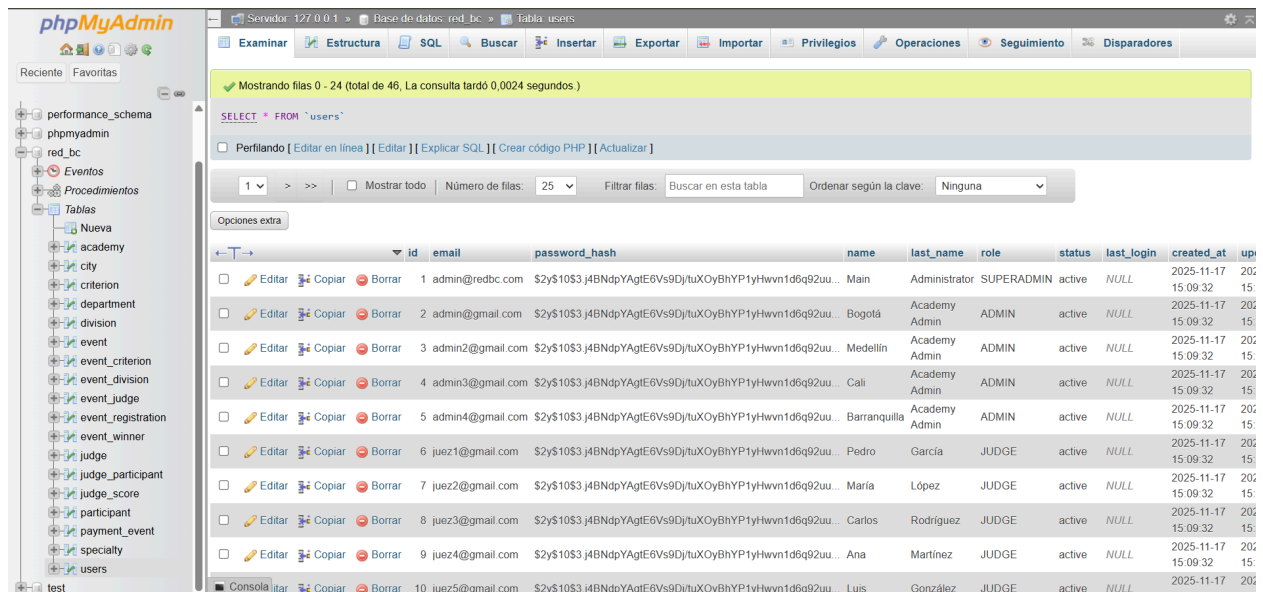
Figura 42. Estructura de carpetas del proyecto en arquitectura (MVC).
Fuente: Elaboración propia.

5.6.1 Base de datos y configuración en phpMyAdmin

En paralelo, durante la fase de arquitectura se definió y creó la base de datos que soportó toda la información del sistema, lo que incluyó tablas para usuarios, participantes, jueces, academias, criterios, divisiones, eventos, calificaciones y registros de inscripción. Este diseño permitió

almacenar la información de forma ordenada y asegurar que las consultas realizadas por el aplicativo fueran rápidas, coherentes y seguras. En la Figura 43 se aprecia la estructura general de la base de datos en phpMyAdmin, donde cada tabla quedó relacionada según las necesidades del proyecto y se configuraron los campos requeridos para garantizar la integridad de los datos.

Además, se llevaron a cabo ajustes iniciales en el entorno de phpMyAdmin, como la creación de índices, la verificación de claves primarias y foráneas, y la revisión de los tipos de datos utilizados en cada columna. Esto permitió asegurar que las operaciones de registro, actualización, consulta y eliminación se ejecutaran adecuadamente durante las pruebas posteriores. En esta misma etapa se configuraron usuarios con permisos específicos para evitar accesos no autorizados y proteger la información del aplicativo, lo que fortaleció la seguridad general del sistema.



	id	email	password_hash	name	last_name	role	status	last_login	created_at	updated_at
<input type="checkbox"/>	1	admin@redbc.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Main	Administrator	SUPERADMIN	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	2	admin@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Bogotá	Academy Admin	ADMIN	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	3	admin2@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Medellin	Academy Admin	ADMIN	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	4	admin3@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Cali	Academy Admin	ADMIN	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	5	admin4@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Barranquilla	Academy Admin	ADMIN	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	6	juez1@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Pedro	García	JUDGE	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	7	juez2@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	María	López	JUDGE	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	8	juez3@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Carlos	Rodríguez	JUDGE	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	9	juez4@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Ana	Martínez	JUDGE	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32
<input type="checkbox"/>	10	juez5@gmail.com	\$2y\$10\$3j4BNdpYAgtE6Vs9DjtuXOyBhYP1yHwv1d6q92uu...	Luis	González	JUDGE	active	NULL	2025-11-17 15:09:32	2025-11-17 15:09:32

Figura 43. Arquitectura de base de datos en localhost phpMyAdmin
Fuente: Elaboración propia.

5.7 Fase 5: Desarrollo

Durante la fase de desarrollo se consolidó la construcción del aplicativo web a partir de los entregables obtenidos en el análisis y el diseño. En primer lugar, se configuró el entorno de

trabajo mediante la instalación de XAMPP, un paquete que integra un servidor web (Apache) y un gestor de base de datos (MySQL). Esta herramienta permitió ejecutar el sistema en un computador local, simular su funcionamiento real y realizar pruebas sin necesidad de un servidor en la nube. En la Figura 44 se observa cómo Apache y MySQL fueron activados para alojar el aplicativo y permitir que las páginas web, los formularios y las consultas de datos funcionen correctamente dentro del entorno de desarrollo.

Posteriormente, se implementó la interfaz del sistema utilizando HTML5, CSS3, JavaScript y Bootstrap, tecnologías responsables de presentar las pantallas al usuario, organizar los formularios, definir estilos visuales y garantizar que la aplicación se adapta a distintos tamaños de pantalla. De manera complementaria, la lógica funcional del sistema se programó en PHP, empleando un enfoque orientado a objetos para administrar registros, validar información, procesar solicitudes y comunicarse con MySQL, que almacena los datos de academias, jueces, participantes, criterios y eventos.

Finalmente, se estructuró de manera organizada el directorio del proyecto, el cual alberga los archivos necesarios para el funcionamiento del aplicativo. Allí se incluyen los formularios que utilizan los usuarios, los controladores encargados de procesar las acciones, los modelos encargados de gestionar los datos, las hojas de estilo, los recursos multimedia y los scripts que permiten la interacción dinámica con la plataforma. Esta organización favoreció la claridad en el desarrollo, facilitó la localización de cada componente y permitió mantener un control adecuado sobre los elementos que conforman el sistema.

Index of /red-bc-app/vista

Name	Last modified	Size	Description
Parent Directory		-	
academias.php	2024-11-21 13:02	5.0K	
anfitrion.php	2025-08-21 10:03	22K	
assets/	2025-09-19 02:11	-	
calificar.php	2025-08-21 09:13	28K	
calificar_copy.php	2024-10-07 19:15	13K	
crear_academia.php	2025-09-19 03:28	20K	
crear_criterio.php	2025-09-04 16:59	15K	
crear_division.php	2025-09-04 16:59	20K	
crear_evento.php	2025-09-04 20:33	24K	
crear_juez.php	2025-09-04 15:59	19K	
crear_participante.php	2025-09-04 16:12	22K	
dashboard_simple.php	2025-08-25 03:23	2	
detalles_evento.php	2025-09-19 02:19	9.3K	
editar_academia.php	2025-09-19 03:28	21K	
editar_evento.php	2025-09-04 22:27	9.4K	
eventos.php	2024-10-07 16:57	5.8K	
examples/	2025-09-19 08:10	-	
ganadores.php	2025-02-11 18:12	7.7K	
includes/	2025-09-19 08:11	-	
inicio.php	2025-08-21 10:42	1.0K	
jueces.php	2024-10-07 17:16	12K	
layout/	2025-09-04 13:07	-	
login/	2025-09-04 15:29	-	
modales/	2025-09-18 22:45	-	
modificar_participan_>	2025-08-21 10:03	32K	
obtener_resultados_p_>	2025-02-11 17:31	795	
participantes.php	2025-08-21 10:03	9.6K	
perfil_superadmin.php	2025-08-25 06:36	15K	
registro_academias.php	2025-08-21 10:03	9.1K	

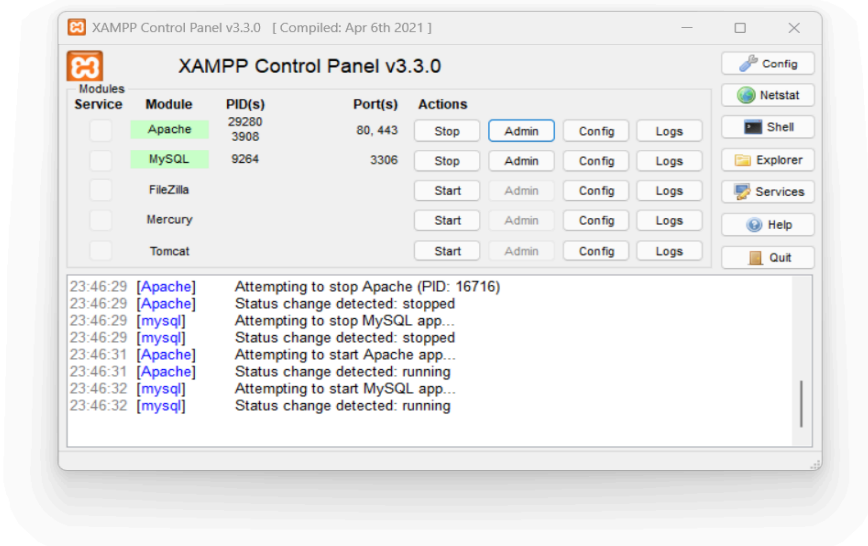


Figura 44. Entorno local en servidor Apache con índice de vistas del sistema.

Fuente: Elaboración propia.

5.8 Fase 6: Implementación

La fase de implementación correspondió a la puesta en marcha del sistema desarrollado y estuvo orientada a su despliegue en un entorno real, con el fin de validar su funcionamiento y garantizar que los objetivos del proyecto se cumplieran de manera efectiva. Para tal fin, se utilizó un dominio gratuito disponible en línea, lo que permitió alojar el aplicativo en un servidor básico y facilitar su acceso desde distintos dispositivos sin incurrir en costos adicionales, aspecto pertinente dado que el presente trabajo fue realizado por una sola estudiante. Además, se llevaron a cabo configuraciones iniciales relacionadas con la base de datos, la seguridad de accesos y los permisos por rol, lo que aseguró que los procesos de registro, inscripción, calificación y validación pudieran ejecutarse de manera controlada.

Posteriormente, se verificó que las rutas internas, los formularios, los controladores y las consultas de la base de datos funcionaran correctamente fuera del entorno local. Esto implicó

ajustar parámetros de conexión, comprobar la integridad de las tablas, revisar la carga de recursos estáticos y validar que las sesiones de usuario permanecieran estables en un servidor externo. Como se observa en la Figura 45, la pantalla de inicio del aplicativo desplegada en el servidor permitió corroborar que el formulario de autenticación cargara adecuadamente y que el ingreso mediante credenciales redirigiera de forma correcta hacia el módulo correspondiente. De igual manera, la Figura 46 muestra el dashboard del superadministrador ya en el entorno real, donde se confirmó que los indicadores y tarjetas de resumen se visualizaran sin errores y que el menú lateral ofreciera acceso estable a cada sección del sistema. En complemento, la Figura 42 ilustra el formulario de gestión de academias, con el que se validó que las operaciones de crear, actualizar, inactivar y listar registros se ejecutaran de manera adecuada como parte de las funciones CRUD del aplicativo.

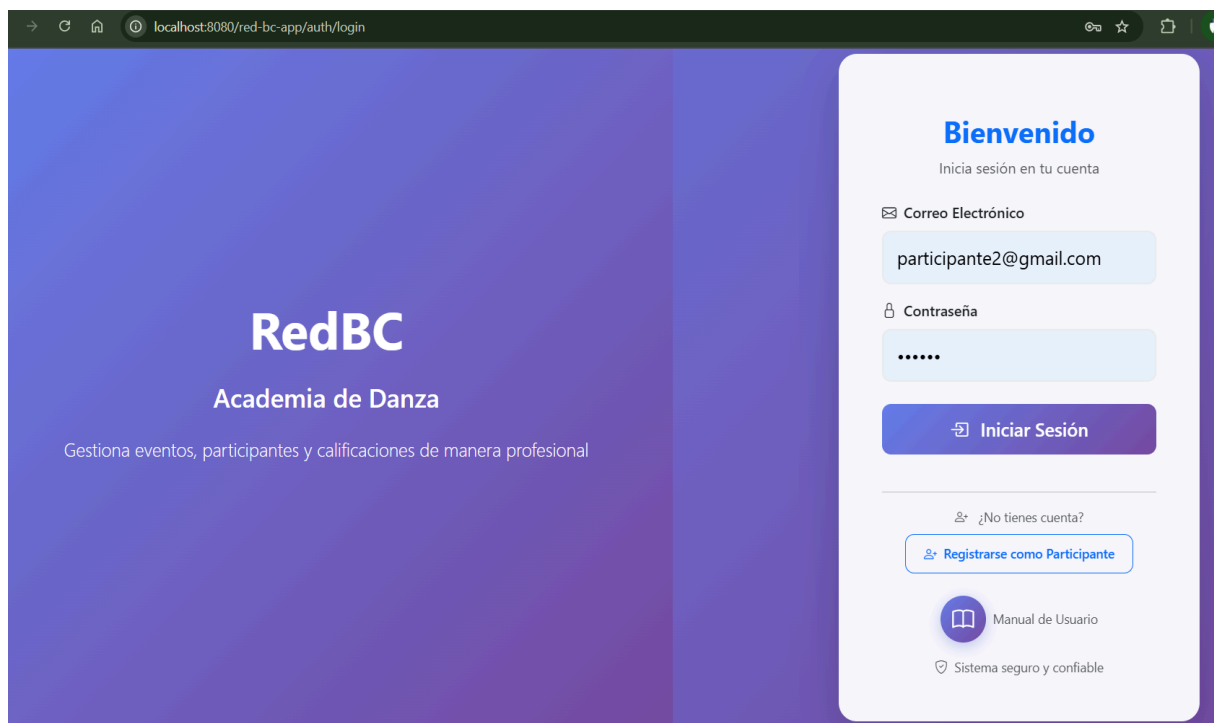


Figura 45. Pantalla de inicio del aplicativo web desplegado en el entorno real.
Fuente: Elaboración propia.

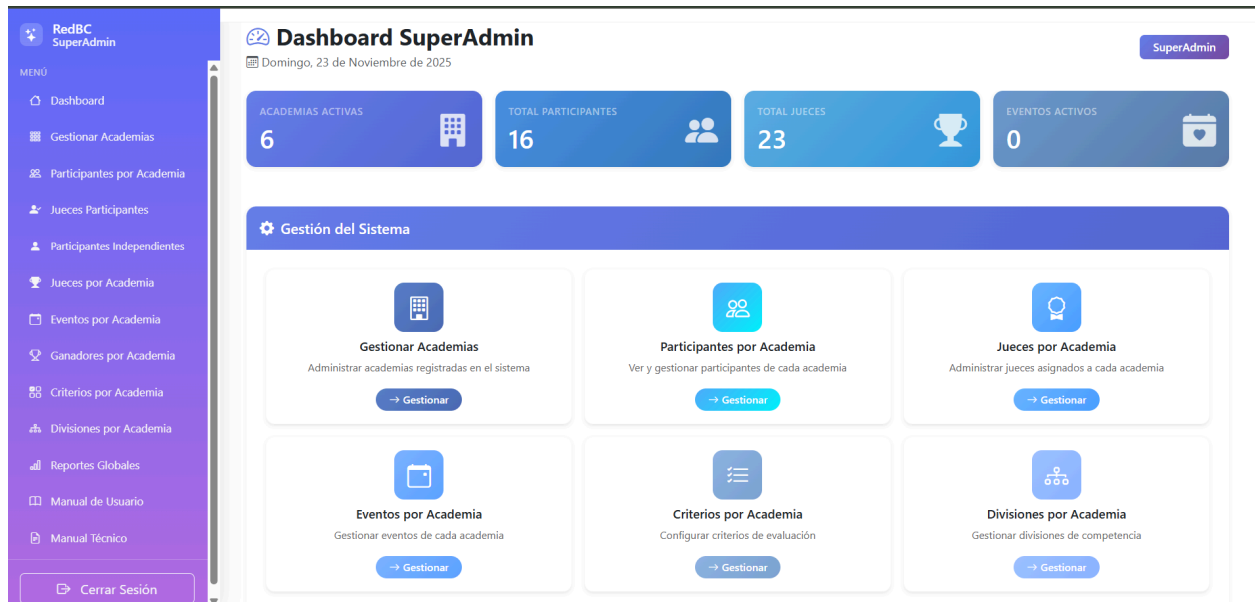


Figura 46. Dashboard superadmin desplegado en el entorno real.
Fuente: Elaboración propia.

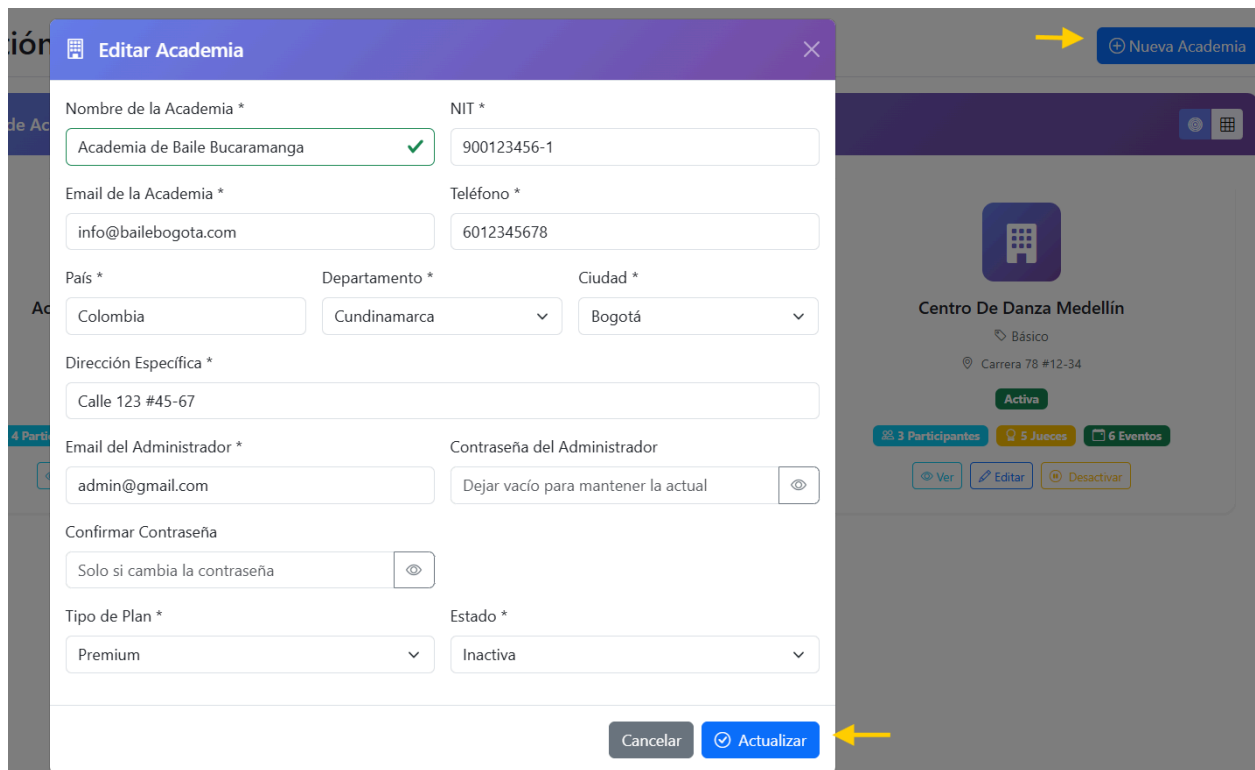


Figura 47. Formulario CRUD academia desplegado en el entorno real.
Fuente: Elaboración propia.

De manera complementaria, se comprobó el correcto funcionamiento del flujo de pagos en el entorno de producción, aspecto clave para la inscripción a eventos. Tal como se aprecia en la Figura 48, la pasarela de pagos permitió seleccionar el método de pago, ingresar los datos requeridos y procesar la transacción sin fallos visibles para el usuario. A su vez, la Figura 49 presenta el comprobante de pago exitoso, mediante el cual se verificó que el sistema generara un recibo con los datos esenciales de la operación, el valor cancelado y el estado aprobado, además de ofrecer accesos directos para revisar las inscripciones o volver al panel principal. Estas evidencias permitieron constatar que el ciclo completo de inscripción y pago se ejecutó de forma coherente y trazable.

En perspectiva, la implementación se constituyó en un espacio de verificación práctica que permitió confirmar la escalabilidad del aplicativo y asegurar que las funcionalidades diseñadas respondieron adecuadamente a las necesidades identificadas en las fases previas. Gracias a este despliegue preliminar, el sistema quedó preparado para su uso operativo en competencias reales, garantizando una experiencia consistente para administradores, jueces y participantes, tanto en los procesos de gestión interna como en la inscripción y el pago de los concursos de baile. Gracias a este despliegue preliminar, el sistema quedó preparado para su uso operativo en competencias reales, garantizando una experiencia consistente para administradores, jueces y participantes.

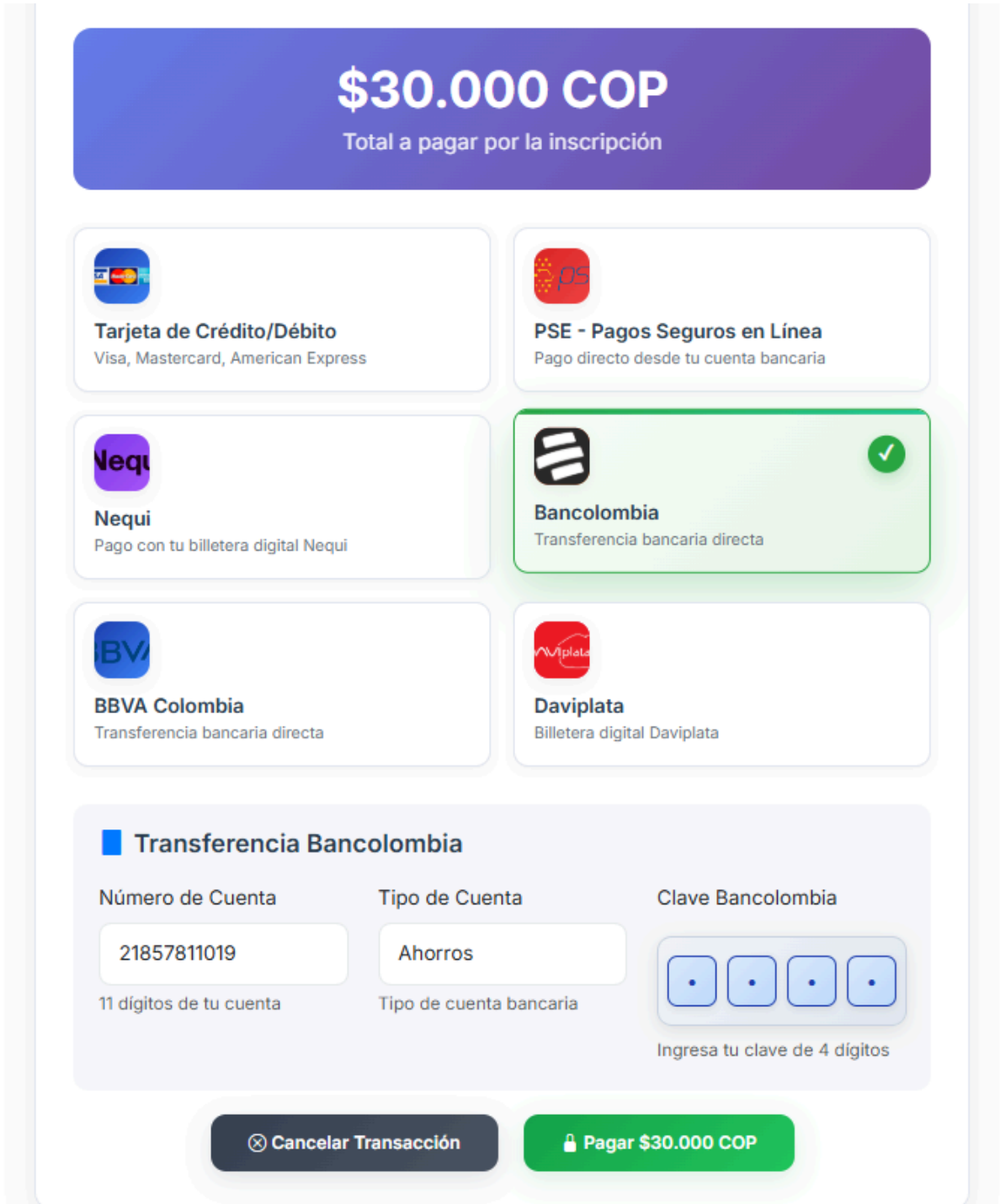


Figura 48. Pasarela de pagos del aplicativo web desplegado en el entorno real.
Fuente: Elaboración propia.

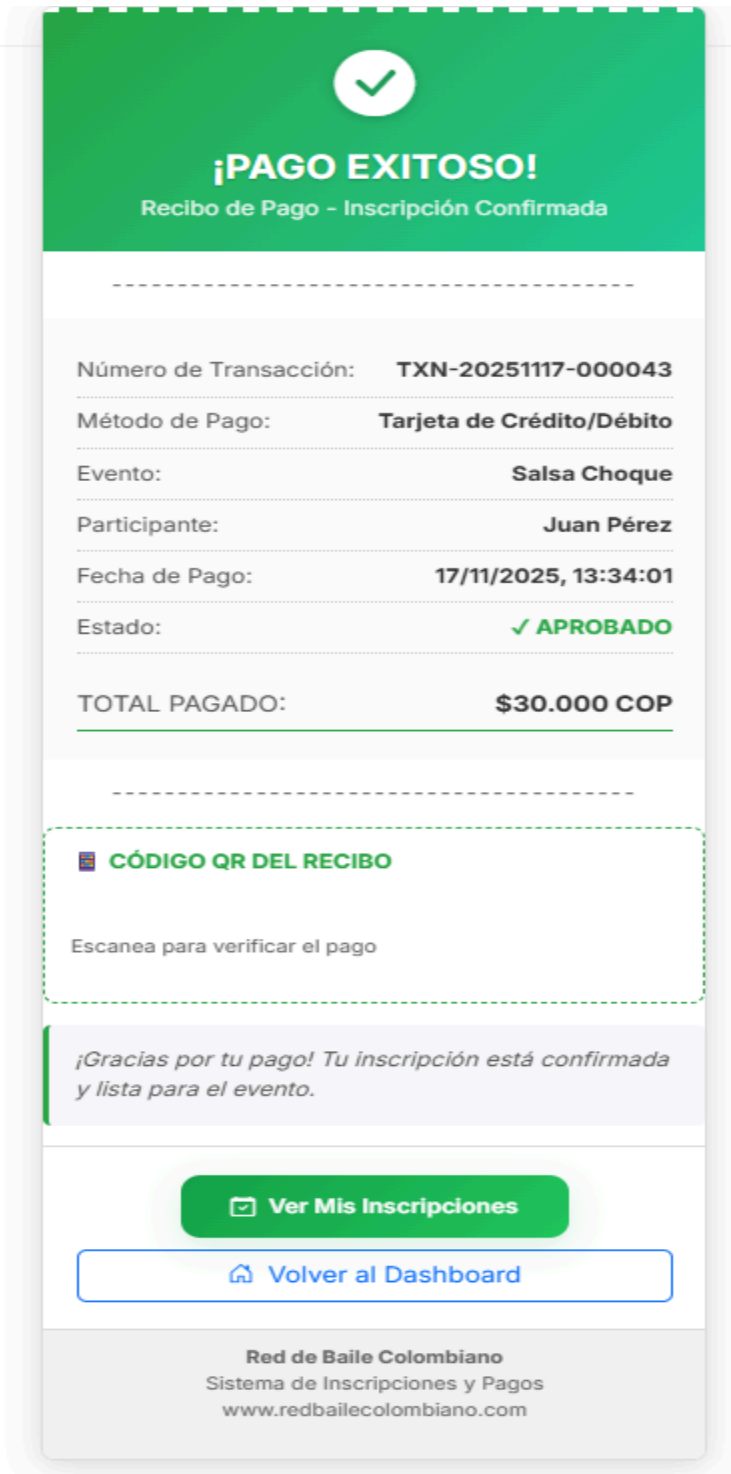


Figura 49. Ticket de pago generado de un evento inscrito por un participante desplegado en el entorno real.
Fuente: Elaboración propia.

5.9 Fase 7: Pruebas

Durante esta fase se realizaron pruebas unitarias con el propósito de verificar que cada módulo crítico del aplicativo funcionara de manera aislada y sin errores. Estas validaciones se ejecutaron sobre los componentes principales del sistema, entre ellos el registro y autenticación de usuarios, la gestión de jueces, la administración de academias, el almacenamiento de puntajes y la consolidación de reportes. Tal como se observa en la Figura 50, el entorno de desarrollo mostró la ejecución satisfactoria de todas las pruebas asociadas a los controladores y funcionalidades clave, lo que evidenció que los métodos evaluados responden correctamente a las entradas previstas y producían salidas consistentes. Gracias a estas verificaciones fue posible confirmar que la lógica interna del sistema se comportaba de forma estable y que los módulos podían avanzar con seguridad hacia las pruebas de integración y las pruebas funcionales realizadas posteriormente.

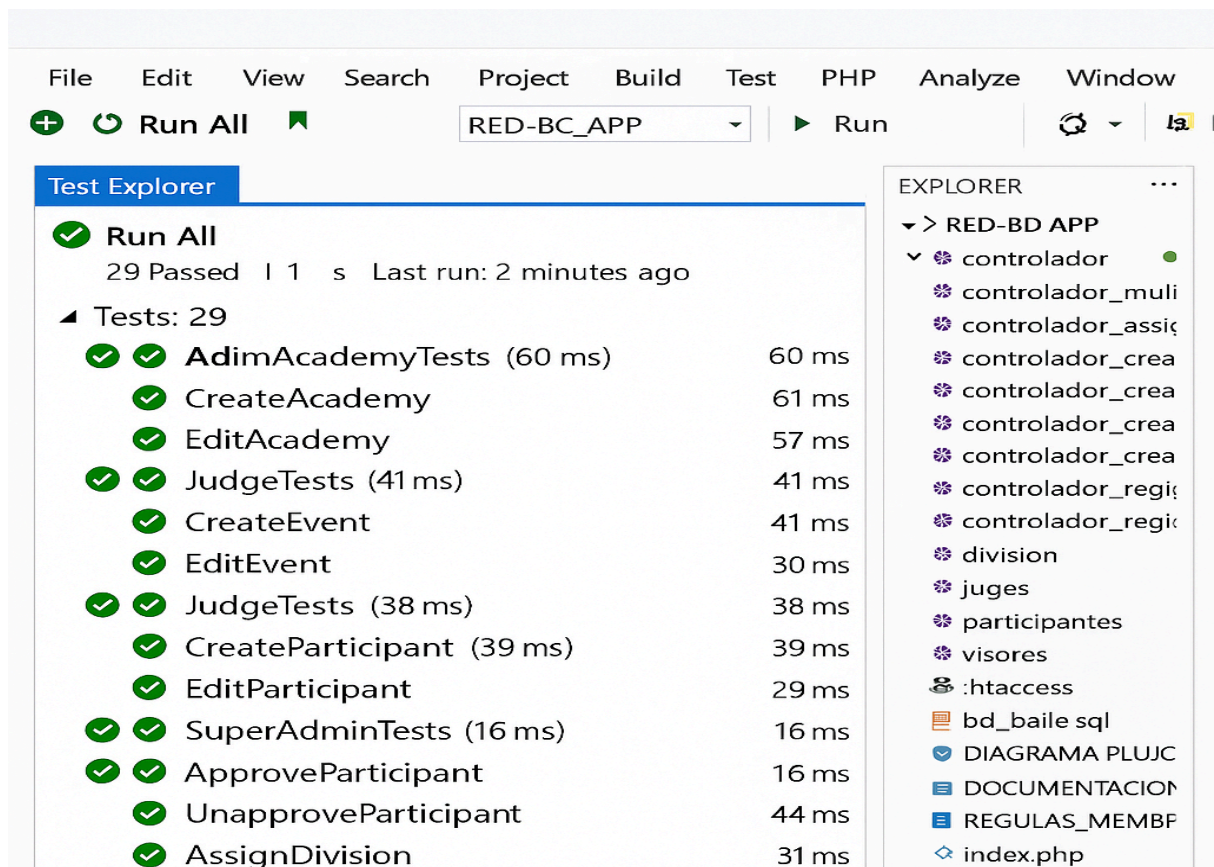


Figura 50. Resultados de pruebas unitarias por rol en el Test Explorer de Visual Studio
Fuente: Elaboración propia.

En adición se llevaron a cabo validaciones orientadas a comprobar que el sistema cumpliera con los objetivos planteados y garantizará su correcto funcionamiento. En primer lugar, se aplicaron pruebas unitarias, las cuales consistieron en verificar de manera individual los módulos esenciales del aplicativo, como el inicio de sesión, el registro de puntajes por parte de los jueces y la generación de reportes consolidados. Con estas validaciones se aseguró que cada componente funcionará de manera independiente y sin errores evidentes. En segundo lugar, se realizaron pruebas de integración, que tuvieron como propósito comprobar el flujo completo de uso, desde el acceso de un juez al sistema hasta la obtención de los resultados finales. De esta forma, se confirmó que los módulos pudieran interactuar de manera correcta y que no se presentaran inconsistencias en la transmisión de datos entre las distintas capas de la aplicación. Finalmente, se ejecutó una prueba piloto en un escenario controlado, lo que permitió simular el uso real del sistema en un evento de baile. Con ello se identificaron aspectos de usabilidad y se corroboró la pertinencia de las funcionalidades implementadas, validando así que la solución fuera práctica y adecuada para el contexto previsto.

De manera complementaria, se llevaron a cabo pruebas de usuario con el fin de evaluar qué tan intuitiva y funcional resultaba la plataforma para personas que interactuaron por primera vez. Para este proceso se seleccionaron usuarios con distintos niveles de experiencia digital y con roles variados dentro del sistema, como participantes de academia, participantes independientes, jueces, administradores de academia y superadministrador. A cada uno se le asignaron tareas específicas acordes con su perfil, y se registró su desempeño sin ofrecer instrucciones directas, de modo que se pudiera observar la claridad de la interfaz, la coherencia del flujo de navegación y la capacidad del usuario para completar acciones sin acompañamiento. Gracias a estas pruebas fue posible medir aspectos como la facilidad de uso, la percepción visual, la comprensión de los mensajes del sistema, los tiempos de respuesta y la efectividad de las validaciones implementadas. Los resultados permitieron identificar fortalezas del diseño y confirmar que el aplicativo ofrecía una experiencia consistente y accesible para diferentes tipos de usuarios, lo que reforzó la pertinencia del prototipo en contextos reales de competencia.

5.10 Fase 8: Entrega y operación

La fase de entrega y operación se enfocó en la puesta a disposición del aplicativo para su implementación en el entorno real. Para ello, se elaboró la documentación técnica, que incluye la descripción de la arquitectura, los requisitos de instalación y la explicación de las funciones principales, de modo que el sistema pudiera mantenerse en el tiempo y adaptarse a futuras actualizaciones. Asimismo, se elaboró un manual de usuario redactado en un lenguaje sencillo, con capturas de pantalla y procedimientos básicos que facilitaron la interacción de jueces y organizadores con la plataforma. Además, esta entrega incluyó manuales específicos para cada rol del sistema, junto con la explicación detallada de las secciones del menú y las funcionalidades asociadas a cada módulo, lo que permitió que los usuarios comprendieran con claridad el alcance de sus permisos y las acciones disponibles.

De manera complementaria, el manual técnico documentó los pasos necesarios para preparar el entorno de ejecución. En él se indicó la instalación adecuada de XAMPP, la configuración inicial del servidor Apache y la activación de los servicios requeridos para ejecutar la aplicación. También se describió el proceso para importar la base de datos en phpMyAdmin y la forma correcta de alojar el proyecto en el servidor local mediante el uso del puerto localhost, garantizando así que cualquier equipo autorizado pudiera poner en funcionamiento el sistema sin contratiempos. Gracias a este conjunto de entregables, el aplicativo quedó listo para operar en entornos reales, con lo cual se aseguró su disponibilidad para competencias de baile y se cumplió con el propósito de mejorar la transparencia y la eficiencia en los procesos de evaluación.

6. Resultados

Las pruebas de usuario se realizaron con el propósito de evaluar qué tan intuitiva, funcional y estable resultaba la aplicación web para distintos perfiles de uso. Para ello, se asignaron tareas específicas que permitieran recorrer por completo las secciones principales de cada módulo, de acuerdo con los permisos y responsabilidades asociados a cada rol dentro del sistema. Estas tareas incluyeron acciones esenciales como iniciar sesión, navegar por el menú, inscribirse a un

evento, registrar calificaciones, consultar históricos, gestionar participantes o jueces, y finalizar procesos como inscripciones, pagos y exportación de reportes. La selección de estas actividades respondió a la necesidad de validar que cada persona pudiera desplazarse de manera autónoma por su dashboard y comprender sin dificultad la distribución de funciones dentro del aplicativo.

En cuanto a los participantes de las pruebas, se le asignó un rol de forma aleatoria, excepto en el caso del usuario Superadmin, que fue otorgado al asesor técnico del proyecto. Esta decisión permitió que el docente acompañante realizara la validación integralmente la aplicación desde la perspectiva del rol con mayor nivel de acceso, lo que incluyó la gestión de academias, criterios, divisiones, participantes, jueces, eventos y reportes globales. Además, su participación permitió corroborar aspectos claves como la coherencia de la arquitectura interna, el cumplimiento de las reglas de negocio, la trazabilidad de los datos, la consistencia visual entre módulos, la seguridad en el manejo de credenciales y la estabilidad del sistema frente a operaciones críticas.

En cuanto a los resultados generales, todos los usuarios reportaron que la aplicación fue intuitiva, amigable y de fácil comprensión. La carga de las vistas se ejecutó sin retrasos perceptibles y cada persona identificó con facilidad los íconos, botones, etiquetas y secciones del menú. Del mismo modo, la estructura visual permitió reconocer rápidamente dónde realizar cada acción, lo que facilitó completar tareas como acceder al perfil, navegar por los formularios, revisar eventos o consultar los historiales. Los participantes destacaron, además, que los mensajes emergentes eran claros, oportunos y útiles para validar acciones exitosas o corregir errores.

En relación con los procesos transaccionales, todos los roles que requerían interacción con la pasarela de pagos lograron completar la operación sin dificultades. La redirección fue estable, la confirmación se visualizó correctamente y la información se actualizó de manera inmediata en cada módulo. Así mismo, los participantes pudieron cerrar sesión sin inconvenientes, volver a iniciar sesión con sus credenciales asignadas y confirmar que sus acciones quedaban registradas en tiempo real en la base de datos.

Los usuarios también lograron completar las tareas principales propuestas para cada rol, lo que permitió confirmar que la distribución del menú, los permisos específicos y la lógica interna del sistema estaban alineados con las necesidades funcionales previstas. En el caso de los jueces,

la calificación de participantes, la consulta del historial y la aplicación de filtros se ejecutaron correctamente. Para los participantes independientes, la inscripción, el pago, la consulta de eventos y la visualización de calificaciones se desarrollaron sin errores. Por su parte, los administradores de academia pudieron gestionar eventos, jueces, participantes, divisiones y criterios con total claridad, mientras que el Superadmin validó satisfactoriamente los procesos de creación y modificación de academias, gestión global de usuarios y exportación de reportes agregados.

los resultados obtenidos en la aplicación de las pruebas de usuario permitieron identificar patrones consistentes en el desempeño del aplicativo y en la percepción de quienes interactuaron con él. Para evaluar la experiencia se utilizaron escalas tipo Likert (1 a 5) y actividades guiadas por tareas, lo que facilitó el análisis tanto cuantitativo como cualitativo del comportamiento del sistema frente a diferentes perfiles de uso. La distribución de los participantes cubrió rangos de edad entre los 18 y los 60 años, con niveles variados de experiencia digital, lo que aportó heterogeneidad a las valoraciones y aseguró que la prueba reflejara situaciones reales de uso.

En términos generales, los resultados mostraron que la mayoría de los participantes calificó positivamente todos los criterios evaluados, especialmente en los aspectos relacionados con la facilidad de uso, la estabilidad del sistema y la claridad visual. En la escala del 1 al 5, la tendencia predominante se concentró en la valoración máxima (5), que fue seleccionada de manera recurrente en los apartados de claridad del diseño, comprensión de acciones, confiabilidad del sistema, fluidez de navegación y velocidad de carga. De hecho, no se registraron puntuaciones bajas ni reportes de fallos críticos durante la interacción, lo que evidenció un funcionamiento estable del prototipo en cada uno de los módulos probados, confirmando que la aplicación cumplió con los criterios de usabilidad, claridad, estabilidad, funcionalidad y coherencia establecidos para el Producto Mínimo Viable. Todos los participantes pudieron navegar el sistema sin supervisión directa, completar las tareas asignadas y comprender la lógica del aplicativo desde su rol correspondiente.

7. Conclusiones

Tomando todo lo anterior en cuenta, los hallazgos encontrados como resultado de las pruebas unitarias respaldaron que el sistema desarrollado alcanzó un nivel satisfactorio de usabilidad, accesibilidad y estabilidad técnica. Los participantes interactuaron de manera natural con las funciones principales sin requerir supervisión directa, lo que demuestra que la interfaz fue clara, comprensible y coherente con las expectativas de funcionamiento propias de un aplicativo orientado a concursos de baile. Este comportamiento fue consistente incluso en usuarios con baja experiencia digital, lo que amplía el alcance potencial de la herramienta en escenarios reales.

En vista de lo observado, la tendencia general hacia valoraciones máximas en la escala de satisfacción indica que el prototipo cumplió con los estándares de desempeño para un Producto Mínimo Viable. La robustez en los módulos de inscripción, calificación, pagos, historial y administración interna confirma que la estructura técnica implementada fue adecuada para soportar el flujo de trabajo planteado desde la fase de diseño. Además, la retroalimentación positiva de los usuarios respalda la pertinencia del modelo de interacción, los tiempos de respuesta, la organización del contenido y la coherencia visual del sistema.

De manera global, las pruebas permitieron concluir que el aplicativo web se consolidó como una solución viable para gestionar procesos de evaluación en concursos de baile, aportando claridad, transparencia y eficiencia en actividades que tradicionalmente presentan riesgos de error, subjetividad y problemas de trazabilidad. El desempeño del prototipo sugiere que la herramienta está en condiciones de ser utilizada en eventos reales y constituye una base sólida para futuras mejoras orientadas a ampliar su escalabilidad y su capacidad de integración con sistemas externos.

8.Referencia Bibliográficas

Boen, F., Van Hoye, K., Vanden Auweele, Y., Feys, J., & Smits, T. (2008). Open feedback in gymnastic judging causes conformity based on informational influencing. *Journal of Sports Sciences*, 26(6), 621–628. <https://doi.org/10.1080/02640410701670393>

Carvajal, K. P. (2019). Diseño de un prototipo de aplicativo web para la gestión de semilleros (tesis de pregrado, Universidad Nacional Abierta y a Distancia – UNAD). <https://repository.unad.edu.co/bitstream/handle/10596/27486/%20kpcarvajalb.pdf?sequence=1>

Departamento Administrativo Nacional de Estadística (DANE). (2020). Encuesta de Consumo Cultural (ECC) 2020. <https://microdatos.dane.gov.co/index.php/catalog/691>

Furnham, A., & Boo, H. C. (2011). A literature review of the anchoring effect. *Journal of Socio-Economics*, 40, 35–42. <https://doi.org/10.1016/j.socec.2010.10.008>

García, A., Carvajal, J., & Sanabria, R. (2025). Desarrollo de una base de datos para la gestión de eventos (tesis, Universidad de Cundinamarca). <https://repositorio.ucundinamarca.edu.co/handle/20.500.12558/49>

Medellín.travel. (2025). Medellín Sí Danza 2025. Medellín.travel. <https://www.medellin.travel/>

Mercado, D. A. (2024, June 23). Medellín: comenzó el XVIII Festival Internacional de Tango. *El Tiempo*. <https://www.eltiempo.com/colombia/medellin/todo-lo-que-debe-saber-del-xviii-festival-internacional-de-tango-de-medellin-3355364>

Ministerio de Cultura de Colombia. (2021). Revisión conceptual.
https://www.mincultura.gov.co/despacho/plan-nacional-de-cultura/Documents/13.%20Revisio%CC%81n%20Conceptual_compressed.pdf

Panko, R. R. (2000). What we know about spreadsheet errors. Proceedings of the EuSpRIG 2000 Conference. Retrieved from <https://eusprig.org/wp-content/uploads/0802.3457.pdf>

Panko, R. R. (2016). What we don't know about spreadsheet errors today: The facts, why we don't believe them, and what we need to do. arXiv. <https://arxiv.org/abs/1602.02601>

Turban, E., Pollard, C., & Wood, G. (2018). Information Technology for Management: On-Demand Strategies for Performance, Growth and Sustainability (11th ed.). John Wiley & Sons.

Universidad EAN. (2022). Propuesta de un sistema de información unificado para eventos culturales. Biblioteca Digital Minerva.
https://repository.universidadean.edu.co/server/api/core/bitstreams/c8ed63a1-6c59-4da8-84b1-1032b74777be/content?utm_source=chatgpt.com

Zitzewitz, E. (2014). Does transparency reduce favoritism and corruption? Evidence from the reform of figure skating judging. *Journal of Sports Economics*, 15(1), 3–30.
<https://doi.org/10.1177/1527002512441479>

9.Anexos

Anexo A. Formulario aplicado para las pruebas de usuario (Página 1)

Usuario 5 – Superadmin

Fecha de prueba: 12/11/2025

Datos del participante:

Campo	Valor
Usuario	Usuario 5
Edad	
Rol evaluado	Superadmin
Fecha y hora	
Profesión	
Nivel de experiencia digital	Alta
Conocimiento previo de apps de concursos	Sí

Acciones realizadas

Nº	Tarea asignada	Descripción	¿Logró completarla?
1	Crear academia	Registrar nueva academia con datos completos.	
2	Editar academia y credenciales	Modificar datos de contacto y credenciales del administrador.	
3	Inactivar academia	Cambiar estado sin eliminar datos.	
4	Crear evento pago público	Publicable globalmente o por academia.	
5	Crear evento gratuito sólo academia	Restringido a miembros de la academia.	
6	Crear juez por academia	Registrar juez vinculado a academia seleccionada.	
7	Crear participante independiente	Registrar participante sin academia. ▾	
8	Inactivar participante independiente	Cambiar estado sin borrar histórico.	
9	Editar criterio de academia	Ajustar ponderación/obligatoriedad/desempate.	
10	Exportar reportes globales PDF	Generar reportes agregados de todas las academias.	

Anexo B. Formulario aplicado para las pruebas de usuario (Página 2)

Evaluación funcional

Nº	Criterio	Resultado
1	Creación/edición de academias y credenciales sin errores.	
2	Filtros por academia y eventos funcionales.	
3	Restricción de eliminación en eventos borrador/pendiente.	
4	El cambio de estado mantiene la trazabilidad.	
5	Gestión de criterios por academia correcta.	
6	Gestión de divisiones y participantes correcta.	
7	Reportes globales correctos y exportables.	

Evaluación visual y de usabilidad

Nº	Criterio	Resultado
1	Menú multi-academia comprensible.	
2	Etiquetas de activo/inactivo claras.	
3	Colores distinguen estados correctamente.	
4	Tipografía y jerarquías adecuadas.	
5	Reportes globales legibles.	
6	Navegación coherente y fluida.	

Evaluación general (Escala 1-5)

Aspecto	1	2	3	4	5
Facilidad de uso					
Velocidad del sistema					
Claridad del diseño					
Cumplimiento de tareas					
Experiencia general					