PROTOTIPO DE SISTEMA PARA LA AGENDA CULTURAL DE LA CIUDAD DE MEDELLÍN

JUANITA RAMÍREZ SÁNCHEZ

ASESOR
JAIME ERNESTO SOTO URDANETA

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2026

Tabla de contenido

1. Planteamiento del problema 4
    1.2 Descripción 4
    1.2 Formulación 4
2. Justificación 5
3. Objetivos 6
    3.1 Objetivo general 6
    3.2 Objetivos específicos 6
3. Marco teórico 7
    3.1. Antecedentes 7
    3.2 Cultura Digital y Gestión Cultural 7
    3.3 Software orientado a la cultura 8
    3.4 Marcos de Trabajo (Frameworks) 8
        3.4.1 Laravel 9
        3.4.2 Spring Boot 9
        3.4.3 Otros framework 10
    2.5 Metodologías de desarrollo de software 11
        3.5.1 Waterfall (Modelo de cascada) 11
        3.5.2 Scrum 12
        3.5.3 Double Diamond (Doble Diamante) 13
4. Marco Metodológico 16
    4.1 Investigación aplicada 16
    4.2 Metodología de desarrollo de sistema de información. 16
    4.3 Relación de las fases con los objetivos específicos 18
5. Resultados 19
    5.1. Fase 1: Descubrir 19
        5.1.1 Requerimientos 21
        5.1.2 Requerimientos del Prototipo 22
        5.1.3 Requerimientos NO Funcionales del Prototipo 22
        5.1.4 Inventario de Módulos 24
        5.1.5 Requerimientos funcionales de los módulos del prototipo 24
        5.1.6 Requerimientos no funcionales 27
    5.2 Fase 2: Definir 28
        5.2.1 Arquitectura del sistema 28
        5.2.2 Modelo conceptual 31
        5.2.3 Diagrama de Chen entidad - Relación 33
        5.2.4 Modelo Físico 34
    3.3 Fase 3: Desarrollar 34

3.3.1 Base de datos PostgreSQL 35
3.3.2 Desarrollo con el Framework Laravel 36
3.4 Fase 4: Entregar 40
3.4.1 Testing 40
3.4.2 Despliegue 50
3.4.3 Documentación del proyecto 51
6. Conclusiones 51
7. Recomendaciones 54
8. Referencias bibliográficas 55
8.1 Bibliografía 55

1. Planteamiento del problema

1.2 Descripción

La oferta cultural en Medellín es amplia y diversa, sin embargo, en su difusión puede presentar inconvenientes relacionados con el acceso al internet y la centralización de la información. Actualmente, los eventos se divulgan a través de diversos canales como redes sociales, correos electrónicos, páginas web, panfletos, entre otros. Aun así, estos canales de difusión pueden presentar limitaciones asociadas como la falta de claridad en la información del evento o que la información no llegue a personas interesadas. Igualmente, la efectividad de la promoción depende del público objetivo al que está dirigido el evento, los recursos de este público y el tipo de evento.

De acuerdo con el "Plan Decenal de Cultura 2025-2035" donde una de sus acciones a diez años es "1.1.4. Diversificación de medios y canales de interacción para la participación de la ciudadanía en los asuntos de interés público de la cultura", alineándose con el propósito del presente proyecto. El desarrollo del presente trabajo plantea el desarrollo de un software donde se busca fortalecer el acceso a la cultura, mejorar la visibilidad de los eventos de la ciudad, fomentando la participación del público y la sostenibilidad de las entidades culturales.

1.2 Formulación

¿Es factible construir un prototipo de aplicación Web que permita el registro, consulta y promoción de los eventos culturales realizados por la Alcaldía de Medellín para un mayor y mejor acceso a la información del arte y la cultura de la ciudad?

2. Justificación

El desarrollo del proyecto se basa en la necesidad de fortalecer la oferta y el acceso cultural en la ciudad de Medellín. El proyecto busca mejorar la experiencia del ciudadano al momento de consultar los eventos disponibles y facilitar su asistencia. Para esto, se propone un prototipo de aplicación web amigable con el usuario, que incluya información clara y precisa de la ubicación de los eventos (dirección y comuna), entre otros.

Con el desarrollo de este prototipo los ciudadanos que viven en las periferias de la ciudad de Medellín podrán acceder a eventos cercanos si así lo desean. Mejorando la participación ciudadana en la oferta cultural de la ciudad. Además, se centraliza la información, la cual se actualizará constantemente según la demanda de los eventos mejorando la eficiencia en la gestión cultural.

3. Objetivos

3.1 Objetivo general

Desarrollar un prototipo de aplicación web que permita el registro, consulta y promoción de los eventos culturales patrocinados por la Alcaldía de Medellín

3.2 Objetivos específicos

1. Levantar los requerimientos de información necesarios para definir el alcance del prototipo.
2. Diseñar la arquitectura funcional y de interfaz del prototipo que garantice un flujo de información eficiente y experiencia amigable.
3. Desarrollar el prototipo según los requerimientos del diseño.
4. Evaluar el funcionamiento del prototipo.
5. Desplegar el prototipo y la documentación en un repositorio de acceso público.

3. Marco teórico

Este capítulo establece las bases tecnológicas para el desarrollo del prototipo de la agenda cultural para la ciudad de Medellín. Más allá de unas simples de teorías, es una guía que conecta conocimientos existentes con nuevas ideas para crear un producto innovador y sólido.

3.1. Antecedentes

Se centran en la necesidad de centralizar la información cultural a través de plataformas digitales. La dispersión de datos en múltiples canales dificulta el acceso de la ciudadanía a los eventos, lo que limita el impacto de la oferta cultural. El presente proyecto busca resolver esta problemática, alineándose con el "Plan Decenal de Cultura 2025-2035" de Medellín, que promueve la diversificación de medios para la participación ciudadana.

Se ha demostrado que la implementación de estos sistemas mejora la visibilidad de los eventos y fomenta la participación del público. Experiencias previas, como la agenda cultural de Bogotá y la plataforma anterior de la Secretaría de Cultura Ciudadana de Medellín, sirven como referencia. Sin embargo, la plataforma de Medellín presentaba limitaciones en la accesibilidad y el enfoque administrativo, lo que la hacía menos participativa para la ciudadanía.

3.2 Cultura Digital y Gestión Cultural

La cultura digital ha cambiado completamente cómo se gestionan los eventos culturales, mezclando lo físico con lo virtual. Gracias a la tecnología, los gestores culturales ya no solo

organizan eventos, sino que crean experiencias completas que van más allá de un lugar físico.

Con herramientas digitales, pueden entender mejor a su público y crear estrategias que lo conecten con la cultura de formas nuevas y emocionantes. Así, la gestión cultural se convierte en una mezcla de creatividad y tecnología, que hace que la cultura llegue a más personas y tenga un impacto más grande.

3.3 Software orientado a la cultura

El desarrollo de software para la cultura requiere una comprensión única de sus necesidades específicas. Este software debe ser más que una base de datos; debe ser una plataforma que refleje el dinamismo y la diversidad de la cultura que representa. Sus funcionalidades pueden incluir: la gestión de eventos , la administración de perfiles de artistas y gestores culturales y la ubicación de actividades. La funcionalidad del software se mide no sólo por su estabilidad técnica, sino por su capacidad para convertirse en un ecosistema vivo que facilite el encuentro entre la creación artística y su público.

3.4 Marcos de Trabajo (Frameworks)

Un Sistema de Información Web es una herramienta que usa hardware y software para recolectar, procesar, almacenar y compartir información a través de Internet. Su propósito es solucionar un problema específico. Para construir un SIW sólido y eficiente, se usan "frameworks" que facilitan el desarrollo y aseguran que se sigan las mejores prácticas. (Pressman & Maxim, 2020).

3.4.1 Laravel

Es un framework de desarrollo web basado en el lenguaje de programación PHP, que adopta el patrón arquitectónico Modelo-Vista-Controlador (MVC) para organizar de manera estructurada y eficiente la creación de aplicaciones. Mediante herramientas predefinidas y una sintaxis clara, simplifica procesos complejos como la gestión de bases de datos y la autenticación de usuarios, acelerando el desarrollo de plataformas digitales escalables y mantenibles. Ideal para proyectos culturales, permite a desarrolladores y gestores concentrarse en diseñar experiencias significativas para los usuarios, democratizando el acceso a soluciones tecnológicas robustas sin requerir expertise técnico avanzado. (Otwell, 2023).

3.4.2 Spring Boot

Es un framework de desarrollo basado en Java que simplifica la creación de aplicaciones empresariales robustas y escalables mediante un enfoque de convención sobre configuración, reduciendo la complejidad técnica al minimizar ajustes manuales. Ideal para plataformas de alta demanda (como agendas culturales masivas), incluye herramientas integradas para seguridad, gestión de bases de datos y APIs, facilitando la conexión con servicios externos como mapas o pasarelas de pago. Optimizado para entornos cloud, garantiza alto rendimiento y estabilidad, permitiendo desarrollar soluciones confiables sin sacrificar flexibilidad. (Johnson et al., 2022).

3.4.3 Otros framework

La elección de un framework de desarrollo depende de factores como los requisitos del proyecto, la escalabilidad esperada, el expertise del equipo de desarrollo y los recursos computacionales disponibles. Junto a Laravel y Spring Boot, otros frameworks relevantes en el panorama del desarrollo web moderno incluyen:

.NET (ASP.NET Core): Un framework de Microsoft multiplataforma y de alto rendimiento, que permite construir desde servicios web hasta aplicaciones web modernas. Es conocido por su robustez y amplia adopción en el ámbito corporativo.

Angular: Una plataforma y framework frontend basado en TypeScript, mantenido por Google. Es usado predominantemente para construir Single-Page Applications (SPA) dinámicas, complejas y con una excelente experiencia de usuario, funcionando como el cliente ideal para consumir APIs construidas con Spring Boot o Laravel.

Django: Es un framework de desarrollo web basado en Python que sigue el principio "baterías incluidas", ofreciendo una amplia gama de herramientas integradas para el desarrollo rápido y seguro de aplicaciones. Su arquitectura Modelo-Template-Vista (MTV) permite crear aplicaciones escalables con una administración automática de interfaces, ideal para proyectos que requieren robustez y rapidez en su implementación.

Express.js: Es un framework minimalista para Node.js diseñado para construir aplicaciones web y APIs de manera ágil y flexible. Como tecnología backend, permite crear

servicios RESTful eficientes con un enfoque simplificado y de alto rendimiento, siendo particularmente adecuado para proyectos que priorizan la ligereza y la escalabilidad horizontal.

2.5 Metodologías de desarrollo de software

Las metodologías de desarrollo de software constituyen el conjunto de procesos, prácticas y principios que guían la planificación, diseño, implementación, prueba y mantenimiento de sistemas informáticos. Su propósito es garantizar la calidad, eficiencia y alineación del producto con los requerimientos del usuario o de la organización. Estas metodologías proporcionan un marco estructurado que facilita la organización del trabajo en equipo, la gestión de tiempos y recursos, y la reducción de riesgos durante el ciclo de vida del proyecto. (Pressman & Maxim, 2020).

3.5.1 Waterfall (Modelo de cascada)

El modelo de cascada es una metodología lineal y secuencial propuesta por Winston W. Royce en 1970, en la que cada fase del desarrollo debe completarse antes de iniciar la siguiente. Es ideal para proyectos con requisitos bien definidos y estables, ya que prioriza la documentación exhaustiva y el control estricto de avances. Sin embargo, su rigidez dificulta incorporar cambios durante el desarrollo, lo que la hace menos adaptable a entornos dinámicos como los proyectos culturales (Royce, 1970).

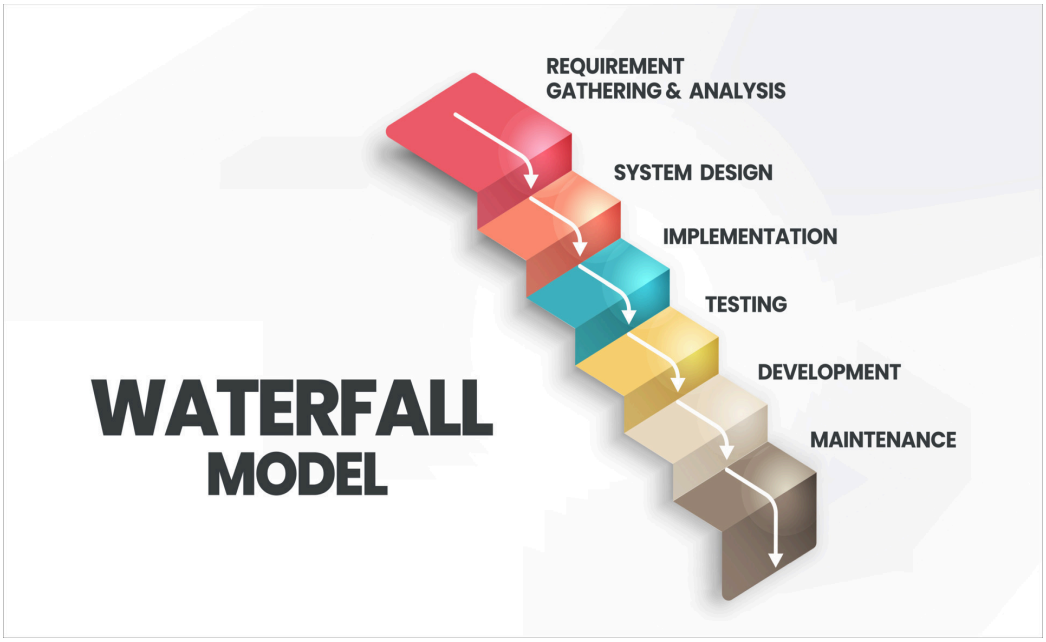


Figura 1. *Modelo Waterfall.*

Nota. De BAP Software, sin fecha. https://bap-software.net/

3.5.2 Scrum

Framework ágil iterativo e incremental centrado en la entrega de valor en ciclos cortos (sprints). Roles clave incluyen el Product Owner (define prioridades), el Scrum Master (facilita el proceso) y el Equipo de Desarrollo. Mediante reuniones diarias, planificación de sprints y revisiones continuas, permite adaptarse rápidamente a cambios y feedback del usuario. Es especialmente útil en proyectos culturales donde los requisitos pueden evolucionar, fomentando la colaboración y la mejora continua.

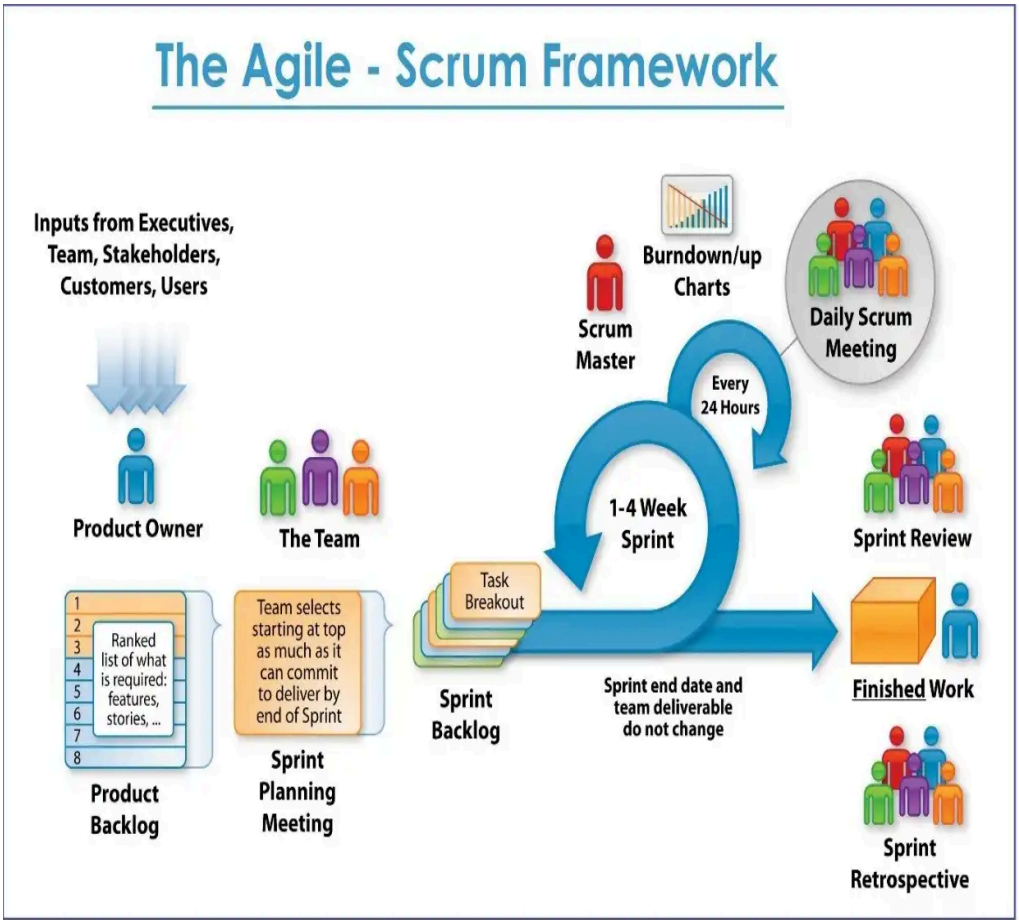


Figura 2. *Scrum Framework*

Nota. De "¿How complex is the Scrum Framework?" por A. Vargas, 2023, LinkedIn. https://www.linkedin.com/pulse/how-complex-scrum-framework-antonio-vargas/

3.5.3 Double Diamond (Doble Diamante)

El modelo Double Diamond es una metodología de diseño centrada en el ser humano, desarrollada por el Design Council del Reino Unido en 2005. Se estructura en cuatro fases: Descubrir (investigar las necesidades reales), Definir (sintetizar los hallazgos), Desarrollar (prototipar soluciones) y Entregar (implementar y validar los resultados). Su enfoque alterna momentos de divergencia y convergencia, lo que permite explorar múltiples perspectivas antes de concretar soluciones. Es especialmente útil en proyectos culturales que requieren una

comprensión profunda de las audiencias y la creación de experiencias significativas (Design Council, 2005).

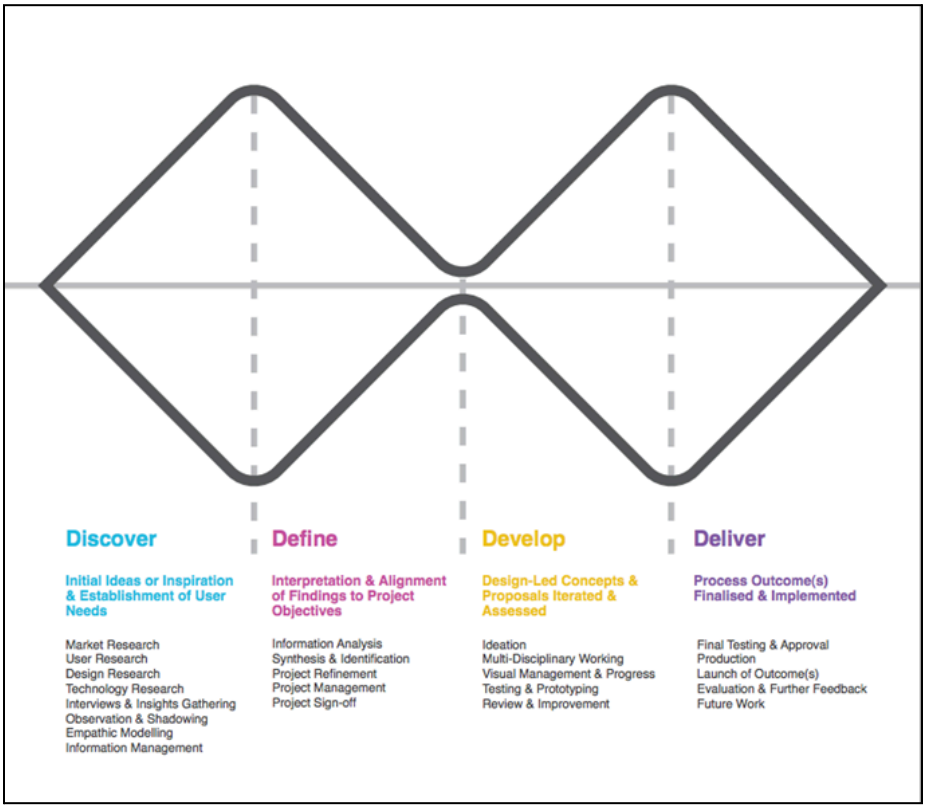


Figura 3. *Double Diamond - Doble diamante*

Nota. De "GUIA DOBLE DIAMANTE" por R. Sanchez, sin fecha. Contacto grafico freelance. https://contactograficofreelance.wordpress.com/otros/

Discover (Descubrir).

En esta primera fase, el objetivo es comprender el problema y las necesidades reales de los usuarios. Implica investigar, observar y recopilar información mediante métodos como entrevistas, grupos focales, encuestas, etnografía o análisis de contexto. Es un proceso divergente, donde se busca generar la mayor cantidad posible de información y perspectivas para identificar oportunidades o desafíos. En el ámbito cultural, esta etapa permite reconocer los intereses, comportamientos y barreras de las audiencias.

Define (Definir)

Durante esta etapa se sintetiza y analiza la información recopilada en la fase anterior para delimitar el problema de diseño. Se seleccionan los hallazgos más relevantes, se agrupan patrones y se formulan los insights clave que orientarán las decisiones. El objetivo es transformar los datos en una declaración clara del desafío a resolver. Aquí se pasa de la exploración a la concreción, cerrando el primer diamante.

Develop (Desarrollar)

En esta fase se promueve la creatividad y generación de soluciones potenciales. A partir del problema definido, los equipos diseñan, prototipan y experimentan diversas alternativas. Es nuevamente un proceso divergente, que fomenta la colaboración interdisciplinaria y la iteración rápida. Se utilizan herramientas como lluvia de ideas, maquetas, diagramas de flujo prototipos funcionales.

Deliver (Entregar)

Finalmente, se selecciona, implementa y evalúa la solución más efectiva. Esta fase implica probar los prototipos con usuarios reales, ajustar detalles y validar que el resultado responda a las necesidades iniciales. Concluye con la entrega del producto o servicio final, acompañado de métricas de impacto y aprendizaje para futuras mejoras. En proyectos culturales, esta etapa asegura que las soluciones sean sostenibles y significativas para la comunidad.

4. Marco Metodológico

4.1 Investigación aplicada

El presente proyecto se enmarca dentro de un tipo de investigación aplicada. Según Hernández Sampieri, Fernández Collado y Baptista Lucio (2014), este tipo de investigación se caracteriza por buscar la aplicación práctica de los conocimientos para resolver problemas específicos o tomar decisiones concretas. Su finalidad no es solo la generación de teoría, sino la puesta en marcha de soluciones efectivas en contextos reales. Este enfoque es el más adecuado para el proyecto, ya que su propósito fundamental es la creación de un prototipo de aplicación web destinado a solucionar la problemática concreta de la descentralización de la información de la agenda cultural de Medellín.

4.2 Metodología de desarrollo de sistema de información.

De las metodologías de desarrollo analizadas en el presente trabajo, se eligió el modelo Double Diamond (Doble Diamante) como enfoque principal para la ejecución del proyecto. Se basa en un proceso de diseño centrado en el ser humano, estructurado en cuatro fases: Descubrir, Definir, Desarrollar y Entregar. Su aplicación resulta especialmente adecuada para proyectos culturales y tecnológicos, ya que promueve la comprensión profunda de las necesidades y comportamientos de los usuarios antes de pasar a la fase de implementación técnica. A través de la alternancia entre momentos de exploración y enfoque, el modelo permite desarrollar soluciones más pertinentes, inclusivas y sostenibles, alineadas con los objetivos del sistema de información.

La metodología se divide en cuatro etapas principales que en el contexto del desarrollo de un sistema de información culturales:

Descubrir: En esta fase, se identificaron las necesidades de la ciudadanía frente al acceso a la información de eventos culturales de la ciudad, mediante el análisis de plataformas culturales similares de otras ciudades, el levantamiento de requerimientos funcionales y no funcionales, y la definición del alcance del prototipo.

Definir: En esta fase, se organizó y sintetizó la información recopilada en la etapa anterior con el fin de diseñar la arquitectura funcional del sistema, los módulos, la base de datos y las interfaces del prototipo, estableciendo la estructura necesaria para la gestión y consulta de eventos culturales de la ciudad.

Desarrollar: Se construirá el prototipo de la Agenda Cultural de Medellín utilizando Laravel y PostgreSQL, implementando funcionalidades como la gestión de eventos, categorías, ubicaciones, inicio de sesión y visualización de la agenda cultural. Además, se integrarán y validarán los diferentes componentes del prototipo para garantizar su correcto funcionamiento.

Entregar: Finalmente, se realizarán pruebas de funcionamiento del prototipo en entorno local, con el fin de validar cada uno de los módulos desarrollados. Asimismo, se demostrará el funcionamiento del sistema mediante un video demostrativo publicado en GitHub y YouTube, acompañado de la documentación técnica y del manual de usuario del proyecto.

Para la construcción técnica, se empleará el framework Laravel, lo que garantizará un prototipo robusto, escalable y eficiente.

4.3 Relación de las fases con los objetivos específicos

La relación entre las fases de la metodología Double Diamond y los objetivos de la investigación es directa y estratégica. El objetivo general de desarrollar un prototipo de aplicación web se cumple mediante la ejecución secuencial y exitosa de las cuatro fases del Doble Diamante. Cada objetivo específico constituye un hito fundamental dentro de una fase específica, asegurando que el proceso de desarrollo esté alineado con las metas de la investigación.

Tabla 1 *Alinear Fases con Objetivos específicos*

#	Fase MDD	Objetivo Específico
1	**Descubrir**	1. Levantar los requerimientos de información necesarios para definir el alcance del prototipo.
2	**Definir**	2. Diseñar la arquitectura funcional y de interfaz del prototipo que garantice un flujo de información eficiente y experiencia amigable.
3	**Desarrollar**	3. Desarrollar el prototipo según los requerimientos del diseño.
4	**Entregar**	4. Evaluar el funcionamiento del prototipo. 5. Desplegar el prototipo y la documentación en un repositorio de acceso público

Nota. Elaboración propia.

5. Resultados

A partir de la aplicación de la metodología Double Diamond en el desarrollo del prototipo, se espera obtener resultados que reflejen un proceso de desarrollo participativo, centrado en las necesidades reales de los usuarios.

5.1. Fase 1: Descubrir

Se llevó a cabo una investigación exhaustiva para entender el contexto real y las necesidades de los usuarios, sin limitaciones previas. En este sentido, se llevará a cabo un proceso exhaustivo de levantamiento de información con el propósito de definir con precisión el alcance y los requisitos del prototipo.

Tabla 2 *Plataformas de agendas culturales en ciudades de Colombia*

Plataforma	Descripción	Observaciones	Web
Agenda cultural de Bogotá	Portal oficial que centraliza la oferta cultural, turística y de ocio en Bogotá, con eventos diarios gratuitos y de pago.	Además de la agenda, incluye trámites, rutas de peregrinaje, caminatas ancestrales (con inscripción previa) y datos sobre participación ciudadana en festivales.	https://bogota.gov.co/que-hacer/agenda-cultural
Calendario de Eventos de Cali	Organiza eventos por fecha y categoría, con opciones de búsqueda y herramientas para compartir actividades en redes sociales y calendarios digitales.	Permite compartir eventos a Facebook, LinkedIn, Google Calendar, Twitter, Instagram y YouTube y Combina actividades culturales con trámites ciudadanos (pago de impuestos, rendición de cuentas), a diferencia de Bogotá que enfatiza más lo artístico y turístico.	https://www.cali.gov.co/calendario/

Nota. Elaboración propia.

De estas agendas culturales de Colombia, la que más presenta lo desarrollado al prototipo de la Agenda Cultural de Medellín es la Agenda Cultural de Bogotá

Agenda cultural de Bogotá

En Bogotá la oferta cultural se canaliza a través de herramientas digitales, como su página web oficial de la ciudad. Este sitio promueve el acceso cultural de los eventos al centralizar la información sobre diversas actividades culturales, ya sea de música, cine, obras de teatro u otras expresiones artísticas.

Allí encontrarás no solo eventos organizados por la Alcaldía Mayor de Bogotá sino también eventos de entidades privadas. Igualmente, la plataforma te brinda información útil como fechas, lugares, puntos de ventas y una breve nota informativa relevante sobre cada actividad. Esta plataforma es de consulta, como se espera realizar el presente prototipo.

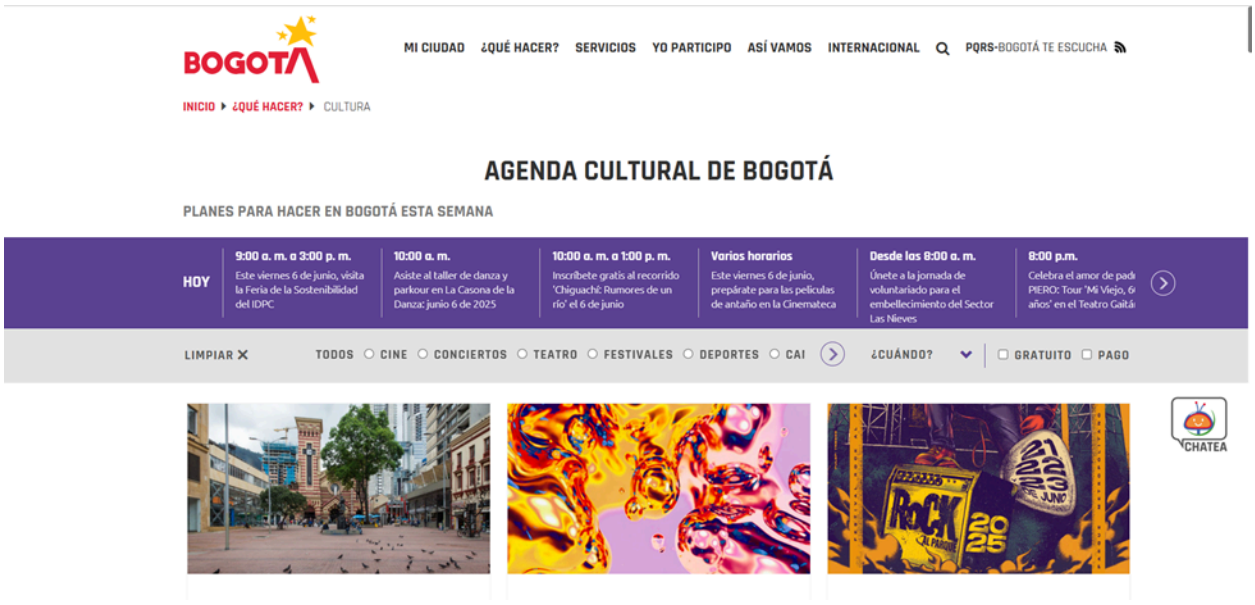


Figura 4. *Agenda cultural de Bogotá*

Nota. Captura de pantalla de Agenda cultural de Bogotá, Pagina web bogota.gov.co, 2025, https://bogota.gov.co/que-hacer/agenda-cultural

5.1.1 Requerimientos

Los requerimientos de un proyecto de software son las necesidades, condiciones o funcionalidades que el sistema debe cumplir para satisfacer las expectativas del usuario y los objetivos del proyecto. Incluyen tanto los requerimientos funcionales, que describen lo que el sistema debe hacer, como los no funcionales, que especifican las características de calidad como rendimiento, seguridad o usabilidad. Estos requerimientos sirven como guía fundamental para el diseño, desarrollo y validación del software, asegurando su coherencia con las metas planteadas (Sommerville, 2011).

Requerimientos Funcionales

Los requerimientos funcionales describen las acciones o procesos específicos que el sistema debe realizar para cumplir con su propósito. Definen las funciones, entradas, salidas y comportamientos esperados del software ante determinadas situaciones, como el registro de usuarios, la gestión de eventos o la generación de reportes. En otras palabras, especifican qué hace el sistema para satisfacer las necesidades del usuario (Sommerville, 2011).

Requerimientos NO Funcionales

Por su parte, los requerimientos no funcionales establecen las características de calidad y las restricciones bajo las cuales el sistema debe operar. Incluyen aspectos como la seguridad, el rendimiento, la usabilidad, la disponibilidad y la escalabilidad, asegurando que el sistema no solo funcione correctamente, sino que lo haga de manera eficiente, confiable y acorde con las

expectativas de los usuarios (Sommerville, 2011).

5.1.2 Requerimientos del Prototipo

Tabla 3 *Requerimientos funcionales del prototipo*

Req	Requerimiento	Descripción	Observaciones	Actor / Rol	P
RF01	Gestión de eventos	El prototipo debe permitir crear, editar, eliminar y consultar eventos culturales.	Incluye información como nombre, fecha, hora, categoría y ubicación	Administrador	A
RF02	Gestión de categorías	Permite administrar las categorías de los eventos.	Incluye crear, editar y eliminar categorías.	Administrador	A
RF03	Gestión de ubicaciones	Permite registrar y gestionar ubicaciones de eventos.	Incluye dirección, comuna.	Administrador	A
RF04	Agenda cultural	Permite a los usuarios consultar los eventos disponibles.	Incluye filtros por fecha, categoría y ubicación.	Usuario final	A
RF05	Login	Permite el inicio de sesión para administradores y/o usuarios con los permisos correspondientes.	Acceso seguro al panel de administración del prototipo.	Administrador	A
RF06	Reportes	Permite generar reportes sobre eventos, categorías y uso del sistema.	Reportes, estadisticas.	Admistrador	M

Nota. Elaboración propia.

5.1.3 Requerimientos NO Funcionales del Prototipo

Tabla 4 *Requerimientos no funcionales.*

Req	Requerimiento	Descripción	Observaciones	Actor / Rol P	P
RNF01	Experiencia del usuario - UIX	El prototipo cuenta con una interfaz intuitiva, amigable y fácil de usar para facilitar la navegación de los	Colores pastel, navegación intuitiva. Para esto se utilizará las directrices de la Casita de la Abuela	Usuario final / Administrador	A

Req	Requerimiento	Descripción	Observaciones	Actor / Rol P	P
		usuarios.			
RNF02	Legales	El protitipo debe cumplir con la Ley 1581 de 2012 sobre protección de datos personales en Colombia.	Se debe garantizar la autorización del tratamiento de datos, almacenamiento seguro, confidencialidad y uso adecuado de la información.	Administrador	A
RNF03	Inclusión y accesibilidad	El prototipo no debe tener elementos alusivos a la discriminación por sexo, pensamiento políticio, religión.	Debe promover lenguaje inclusivo, adecuado contraste de colores, claridad en los textos y acceso equitativo a la información para todo tipo de usuarios.	Usuario final / Administrador	M
RFN04	Rendimiento	Debe garantizar tiempos de respuesta rápidos en las consultas.	Las consultas de eventos deben ejecutarse en un tiempo menor a 3 segundos.	Usuario final	A
RFN05	Disponibilidad	El prototipo debe estar disponible y operativo en entorno local durante las sesiones de prueba y demostración.	El sistema deberá ejecutarse de manera estable y sin fallos durante las pruebas funcionales realizadas en ambiente local.	Administrador	M

Nota: P=Prioridad: A=Alta M=Media B=Baja

Nota. Elaboración propia.

5.1.4 Inventario de Módulos

Tabla 5 *Inventario de módulos del prototipo*

Código	Módulo	Descripción	Actor / Rol	P
M01	Gestión de Eventos	Permite crear, modificar y eliminar eventos. Incluye datos como: nombre, descripción, categoría, fecha, hora, lugar.	Administrador/ Gestor cultural	A
M02	Gestión de Categorías	Permite ver la clasificación de los eventos dependiendo del enfoque artístico.	Administrador/ Gestor cultural	A
M03	Gestión de Ubicaciones	La ubicación geográfica de los eventos en la ciudad, incluyendo información como dirección, comuna y puntos de referencia para facilitar la localización de cada actividad cultural por parte de los ciudadanos.	Administrador/ Gestor cultural	M
M04	Login	Gestiona el acceso seguro al prototipo mediante la autenticación de credenciales de usuario. Permite a los administradores o gestores culturales ingresar para gestionar la información de eventos.	Administrador/ Gestor cultural	A
M05	Reportes	Genera informes detallados sobre los eventos registrados, participación y estadísticas de uso del sistema.	Administrador	M
M06	Agenda Cultural	Permite a los usuarios visualizar los eventos culturales disponibles en la ciudad, filtrándose por fecha, categoría o ubicación. Presenta la información de manera organizada, mostrando detalles como nombre del evento, lugar, hora y tipo de actividad.	Usuario final	M
M07	Configuración General	Configuración del perfil del administrador, permitiendo la edición de su información personal y cambio de contraseña.	Administrador	A
M08	Seguridad	Usuarios Roles Permisos de quien se registra en la plataforma,	Administrador	A

Nota. Elaboración propia.

5.1.5 Requerimientos funcionales de los módulos del prototipo

Tabla 6 *Módulo "Gestión de Eventos"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_001	Crear evento	El administrador podrá registrar nuevos eventos con nombre, descripción, fecha, hora, categoría y ubicación e imagen.	Administrador/Gestor cultural	A
RF_002	Editar evento	El administrador podrá modificar la información de	Administrador/Gestor	A

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
		los eventos registrados.	cultural	
RF_003	Eliminar evento	El administrador podrá eliminar eventos según sea el caso.	Administrador/Gestor cultural	A
RF_004	Consultar eventos	El sistema mostrará una lista de eventos registrados con sus detalles principales.	Administrador/Gestor cultural	A

Nota. Elaboración propia.

Tabla 7 *Módulo "Gestión de Categorías"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_005	Crear categoría	El administrador podrá crear nuevas categorías para clasificar eventos.	Administrador	A
RF_006	Editar categoría	El administrador podrá modificar las categorías existentes.	Administrador	A
RF_007	Eliminar categoría	El administrador podrá eliminar categorías en desuso.	Administrador	A

Nota. Elaboración propia.

Tabla 8 *Módulo "Gestión de Ubicaciones"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_008	Registrar ubicación	El administrador podrá registrar ubicaciones con dirección y comuna asociada.	Administrador	M
RF_009	Editar ubicación	El administrador podrá modificar la información de una ubicación registrada.	Administrador	M
RF_010	Eliminar ubicación	El administrador podrá eliminar ubicaciones que no estén asociadas a eventos activos.	Administrador	M
RF_011	Visualizar eventos	El sistema mostrará la dirección y comuna de cada evento, permitiendo al usuario ubicar geográficamente el evento	Usuario final	A

Nota. Elaboración propia.

Tabla 9 *Módulo "Login"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_012	Inicio de sesión administrador	El administrador y/o usuario con los permisos correspondientes ingresará su correo y contraseña para acceder al panel de administración	Administrador	A

Nota. Elaboración propia.

Tabla 10 *Módulo "Reportes"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_013	Generar reportes de eventos	El sistema generará reportes de estadísticas sobre eventos (cantidad, categorías, ubicaciones más utilizadas).	Administrador/Gestor cultural	M

Nota. Elaboración propia.

Tabla 11 *Módulo "Agenda cultural"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_014	Consultar agenda cultural	Los ciudadanos podrán consultar la agenda cultural filtrando por categoría, fecha o ubicación.	Usuario final	M
RF_015	Detalle de evento	Los ciudadanos podrán visualizar la información completa de un evento seleccionado.	Usuario final	M

Nota. Elaboración propia.

Tabla 12 *Módulo "Configuración general"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_016	Consultar perfil de usuario	El administrador podrá visualizar su información personal (nombre, email, rol)	Administrador	A
RF_017	Editar perfil	El administrador podrá modificar su nombre, teléfono y foto de perfil	Administrador	A
RF_018	Cambiar contraseña	El administrador podrá cambiar su contraseña ingresando la actual y una nueva	Administrador	A

Nota. Elaboración propia.

Tabla 13 *Módulo "Seguridad"*

ID_RF	Nombre requisito	Descripción	Actor / Rol	P
RF_019	Consultar usuarios	El prototipo mostrará una tabla con todos los usuarios registrados.	Administrador	A
RF_020	Crear usuario	El administrador creará cuentas para usuarios con permisos para crear eventos.	Administrador	A
RF_021	Editar perfil	El administrador podrá modificar datos de usuarios existentes, incluyendo los del administrador.	Administrador	A
RF_022	Eliminar usuario	El administrador podrá eliminar usuarios del sistema	Administrador	A
RF_023	Ver roles	El administrador podrá ver los roles disponibles	Administrador	A
RF_024	Asignar permisos	El administrador podrá definir que permisos tiene cada rol	Administrador	A
RF_025	Verificación de permisos	El sistema valida el acceso según rol del usuario	Sistema	A

Nota. Elaboración propia.

5.1.6 Requerimientos no funcionales

Tabla 14 *Requerimientos no funcionales*

ID_RNF	Requisito	Descripción	Actor / Rol	Prioridad
RNF_001	Usabilidad de la interfaz	La aplicación debe contar con una interfaz responsive y fácil de usar para ciudadanos y administradores.	Usuario / Administrador	Alta
RNF_002	Disponibilidad del sistema	El sistema debe estar disponible en línea la mayor parte del tiempo, con mínimos periodos de mantenimiento.	Todos	Alta
RNF_003	Protección de datos	El sistema debe proteger los datos de los usuarios y administradores mediante cifrado y autenticación segura.	Administrador / Usuario	Alta
RNF_004	Soporte de carga	El sistema debe soportar un creciente número de usuarios y eventos sin afectar el rendimiento.	Todos	Alta

ID_RNF	Requisito	Descripción	Actor / Rol	Prioridad
RNF_005	Tiempo de respuesta	Las consultas y visualizaciones de eventos deben ejecutarse en menos de 3 segundos.	Usuario	Alta
RNF_006	Compatibilidad multiplataforma	La aplicación debe ser accesible desde navegadores modernos y dispositivos móviles.	Usuario	Alta

Nota. Elaboración propia.

5.2 Fase 2: Definir

En esta fase se organizó la información recopilada durante la etapa "Descubrir" con el propósito de diseñar la arquitectura funcional del prototipo, la estructura de la base de datos, los módulos y las interfaces necesarias para el desarrollo de la Agenda Cultural de Medellín.

5.2.1 Arquitectura del sistema

Para el desarrollo del presente prototipo se diseñó una arquitectura en capas, con el propósito de organizar de manera eficiente la gestión, almacenamiento y visualización de la información relacionada con los eventos culturales de la ciudad.

Esta arquitectura permite separar las funcionalidades de consulta ciudadana, administración de eventos y gestión de datos, facilitando la centralización de la información cultural en un único sistema. De esta manera, se mejoró el acceso de los ciudadanos a la oferta cultural de Medellín y se fortaleció la gestión realizada por los administradores del sistema.

Frontend.

Fue definido utilizando HTML5, Tailwind CSS, JavaScript y las plantillas Blade integradas en Laravel con el propósito de brindar a los ciudadanos una interfaz amigable y accesible para la consulta de eventos culturales de la ciudad de Medellín.

A través de esta capa, los usuarios pueden visualizar información relacionada con fechas, categorías, ubicaciones e imágenes de los eventos culturales registrados en el sistema, facilitando el acceso centralizado a la oferta cultural de la ciudad. Los módulos implementados para el usuario final, administrador del sistema y gestor cultural fueron los siguientes:

Módulo de Gestión de eventos: Visualización de la información completa de cada evento, incluyendo nombre, descripción, fecha, hora, categoría, ubicación e imagen representativa.

Módulo de Categorías: Organización y filtrado de los eventos culturales según su categoría artística, facilitando la búsqueda de actividades culturales.

Módulo de Ubicaciones: Visualización de la dirección y comuna de cada evento cultural para facilitar su localización por parte de los usuarios.

Módulo de Login: Autenticación de usuarios mediante el ingreso de credenciales para acceder de manera segura al sistema.

Módulo de administrador: Interfaz administrativa para el inicio de sesión y acceso seguro al panel de gestión del sistema.

Módulo de Reportes: Representación de información estadística mediante gráficos y tablas relacionadas con los eventos registrados y el uso del sistema.

Módulo de Configuración General: Gestión de la información del perfil del administrador, actualización de datos personales y cambio de contraseña.

Módulo de Seguridad: Administración de usuarios, roles y permisos para controlar el acceso a las funcionalidades del sistema y garantizar la seguridad de la información.

Backend

El backend del prototipo fue definido utilizando el framework Laravel y el lenguaje de

programación PHP, bajo el patrón arquitectónico Modelo-Vista-Controlador (MVC), con el propósito de estructurar la lógica del sistema, la gestión de la información cultural y la comunicación con la base de datos PostgreSQL.

En esta capa se definieron los modelos, controladores, rutas y servicios necesarios para la administración de los eventos culturales y el funcionamiento general del prototipo. Asimismo, se establecieron los módulos encargados de gestionar la autenticación de usuarios, la administración de eventos, categorías, ubicaciones, reportes y seguridad del sistema.

Módulo de Gestión de Agenda Cultural: Consulta y filtrado de eventos culturales registrados en la base de datos.

Módulo de Gestión de Eventos: Administración de eventos mediante CRUD (Crear, Leer, Actualizar, Eliminar).

Módulo de Categorías: Para administrar las categorías utilizadas para clasificar los eventos culturales.

Módulo de Login: Para la autenticación e inicio de sesión de los usuarios con los permisos adecuados.

Módulo de Reportes: Generación de estadísticas y reportes relacionados con los eventos registrados en el sistema.

Módulo de Ubicaciones: Gestión de la información de dirección y comuna asociada a los eventos culturales.

Módulo de Configuración General: Administración de la información del perfil del usuario y configuración básica del sistema.

Módulo de Seguridad: Gestión de usuarios, roles y permisos para controlar el acceso a las funcionalidades del sistema.

Base de datos

La base de datos del prototipo de información fue definida utilizando PostgreSQL como

sistema gestor de bases de datos relacional, con el propósito de almacenar y organizar la información relacionada con los eventos culturales de la ciudad de Medellín.

Asimismo, mediante Laravel se definieron migraciones para la creación y administración de las tablas del sistema, facilitando la estructuración de la base de datos y el control de cambios durante el desarrollo del prototipo. Estas migraciones permitieron establecer claves primarias, claves foráneas y restricciones de integridad para garantizar la consistencia y disponibilidad de la información cultural almacenada en el sistema.

5.2.2 Modelo conceptual

Inventario de entidades

Tabla 15 *Inventario de entidades*

#	Entidad	Descripción Entidad
1	Eventos	Registra cada actividad cultural del sistema. Incluye nombre, descripción, categoría, fecha, hora, ubicación y organizador.
2	Categoría	Define la clasificación de los eventos según su tipo (música, teatro, danza, literatura, arte visual, etc.), facilitando su organización y filtrado.
3	Ubicaciones	Almacena la información geográfica de los eventos, incluyendo dirección, comuna y puntos de referencia para su localización.
4	Usuario	Representa a los usuarios finales del sistema (ciudadanos o visitantes) que consultan la agenda cultural o interactúan con los eventos.
5	Rol	Define los diferentes perfiles del sistema (administrador, gestor, usuario), controlando permisos de acceso y acciones disponibles.
6	Permisos	Define los permisos del rol

Nota. *Elaboración propia.*

Inventario de Relaciones

Tabla 16 *Inventario de relaciones*

#	Nombre Relación	Descripción	Entidades		Cardinalidades	
1	**evento_ubicacion**	Un evento tiene una ubicación; Una ubicación tiene muchos eventos	**Izquierda**	evento	**Izq/Der**	**N:1**
			Derecha	ubicacion	**Der/Izq**	**1:N**
2	**evento_categoria**	Un evento tiene una categoría; Una categoría tiene muchos eventos	**Izquierda**	evento	**Izq/Der**	**1:1**
			Derecha	categoria	**Der/Izq**	**1:N**
3	**usuario_rol**	Un usuario tiene un rol; Un rol tiene muchos usuarios	**Izquierda**	usuario	**Izq/Der**	**1:N**
			Derecha	rol	**Der/Izq**	**1:1**
4	**rol_permiso**	Un rol tiene muchos permisos; Un permiso tiene muchos roles	**Izquierda**	rol	**Izq/Der**	**1:N**
			Derecha	permiso	**Der/Izq**	**1:N**
4	**usuario_evento**	Un usuario tiene evento; Un evento evento tiene muchos usuarios	**Izquierda**	usuario	**Izq/Der**	**1:1**
			Derecha	evento	**Der/Izq**	**1:N**

Nota. Elaboración propia.

5.2.3 Diagrama de Chen entidad - Relación

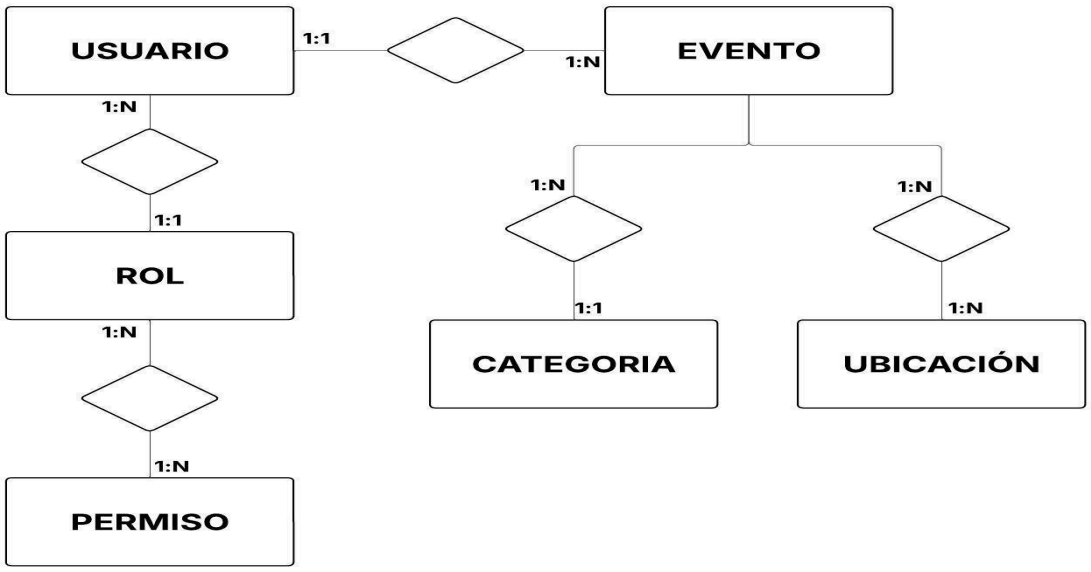


Figura 5. *Diagrama de Entidad-Relación*

Nota. Elaboración propia

5.2.4 Modelo Físico

Tabla 17 *Tabla eventos*

1	Tabla	eventos	Descripción Tabla	Registra cada actividad cultural del sistema. Incluye nombre, descripción, categoría, fecha, hora, ubicación y organizador.				Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios
1	id_evento	Id del evento	bigint		No	S			
2	nombre_evento	Nombre del evento cultural	Varchar	100	No				
3	descripcion	Descripción del evento cultural	Text						
4	imagen	Ecard representativa del evento cultural	Varchar	255					
5	fecha	Fecha del evento cultural	Date		No				
6	hora	Hora del evento cultural	Time		No				
7	id_categoria	Id de la tabla categoria	bigint				S		
8	id_ubicacion	Id de la tabla ubicación	bigint				S		
9	id	Id de la tabla de users	bigint				S		
10	created_at	Fecha creación	timestamp						
11	updated_at	Fecha actualización	timestamp						

Nota. Elaboración propia

Tabla 18 *Tabla categorias*

2	Tabla	categorias	Descripción Tabla	Define la clasificación de los eventos según su tipo (música, teatro, danza, literatura, arte visual, etc.), facilitando su organización y filtrado.				Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios
1	id_categoria	ID de la categoría	bigint		No	S			
2	nombre_categoria	Nombre de la categoría del evento	Varchar	100	No				
3	descripcion	Descripción	Text						
4	created_at	Fecha creación	timestamp						
5	updated_at	Fecha actualización	timestamp						

Nota. Elaboración propia

Tabla 19 *Tabla ubicaciones*

3	Tabla	ubicaciones	Descripción Tabla	Almacena la información geográfica de los eventos, incluyendo dirección, comuna y puntos de referencia para su visualización en el mapa.				Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios
1	id_ubicacion	ID de la ubicación	Bigint		No	S			
2	nombre_lugar	Nombre del lugar	Varchar	255	No				
3	direccion	Dirección	Varchar	255	No				
4	comuna	Comuna	Varchar	50					
5	ciudad	Ciudad donde se realiza el evento el cual es Medellín por defecto	Varchar	100					Por defecto Medellín
6	departamento	Departamento donde se realiza el evento el cual es Antioquia por defecto	Varchar	100					Por defecto Antioquia
7	pais	Pais donde se realiza el evento el cual es Colombia por defecto	Varchar	1000					Por defecto Colombia
8	observaciones	Observaciones	Text						
9	created_at	Fecha creación	timestamp						
10	updated_at	Fecha actualización	timestamp						

Nota. Elaboración propia

Tabla 20 *Tabla users*

4	Tabla	users	Descripción Tabla	Representa a los usuarios finales del sistema (ciudadanos o visitantes) que consultan la agenda cultural o interactúan con los eventos.				Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios
1	id	Identificación del usuario	bigint		No	S			Basado en el sistema de login de Laravel
2	name	Nombre completo del usuario	Varchar	255	No				
3	email	Correo electronico del usuario	Varchar	255	No			S	
4	email_verified_at	Fecha verificación email	timestamp						
4	password	Contraseña del usuario encriptada	Varchar	255	No				
5	id_rol	Rol asignado al usuario	bigint				S		
6	remember_token	Token "recordarme"	Varchar	100					
7	created_at	Fecha creación	timestamp						
8	updated_at	Fecha actualización	timestamp						

Nota. Elaboración propia

Tabla 21 *Tabla roles*

5	Tabla	roles	Descripción Tabla	Define los diferentes perfiles del sistema (administrador, gestor, usuario), controlando permisos de acceso y acciones disponibles.					Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios	
1	id_rol	Id del rol	bigint		No	S				
2	nombre_rol	Nombre del rol	Varchar	50	No					
3	descripcion	Descripción	Text							
4	created_at	Fecha creación	timestamp							
5	updated_at	Fecha actualización	timestamp							

Nota. Elaboración propia

Tabla 22 *Tabla permisos*

6	Tabla	permisos	Descripción Tabla	Define los permisos					Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios	
1	id_permiso	Id del permiso	bigint		No	S				
2	nombre_permiso	Nombre del permiso	Varchar	120	No					
3	descripcion	Descripción	Text							
4	created_at	Fecha creación	timestamp							
5	updated_at	Fecha actualización	timestamp							

Nota. Elaboración propia

Tabla 23 *Tabla rol_permiso*

7	Tabla	rol_permiso	Descripción Tabla	Define los permisos del rol					Versión	1
#	Campo	Descripción Campo	Tipo Dato	Tamaño / Longitud	Nulo	PK	FK	UK	Nota/Comentarios	
1	id_rol	Id del rol	bigint		No	S	S			
2	id_permiso	ID del permiso	bigint		No	S	S			
3	created_at	Fecha creación	timestamp							
4	updated_at	Fecha actualización	timestamp							

Nota. Elaboración propia

Inventario de tablas

Tabla 24 *Inventario de las tablas.*

Nombre	Nombre físico	Descripción	Clave primaria (PK)	Claves foráneas (FK)
Evento	evento	Registra los eventos culturales de la ciudad, incluyendo nombre, descripción, fecha, hora, categoría, ubicación y usuario creador.	id_evento	id_categoria, id_ubicacion, id_usuario
Categoría	categoria	Define los tipos de eventos (música, teatro, danza, literatura, arte, etc.) para clasificar la oferta cultural.	id_categoria	
Ubicaciones	ubicaciones	Almacena la información geográfica de los eventos, como dirección y comuna.	id_ubicacion	
Usuario	usuario	Representa a los usuarios del sistema que gestionan o consultan los eventos.	id_usuario	id_rol
Rol	rol	Define los diferentes niveles de permisos dentro del sistema	id_rol	
Permisos	permiso	Define los permisos del rol	id_permiso	
Rol - Permiso	rol_permiso	Relación entre el rol y permiso	id_rol id_permiso	

Nota. Elaboración propia

3.3 Fase 3: Desarrollar

Esta fase consistió en la construcción e implementación del prototipo funcional de la Agenda Cultural de Medellín, mediante la codificación de los componentes definidos en las fases anteriores y el uso de las tecnologías seleccionadas para el desarrollo del sistema.

Durante esta etapa se desarrollaron las funcionalidades relacionadas con la gestión de eventos culturales, autenticación de usuarios, administración del sistema, generación de reportes y visualización de información cultural, integrando el frontend, backend y base de datos en un entorno de desarrollo local.

3.3.1 Base de datos PostgreSQL

La base de datos del sistema fue implementada en PostgreSQL mediante migraciones y scripts de Lenguaje de Definición de Datos (DDL), permitiendo la creación de la estructura física de las tablas, atributos, restricciones y relaciones necesarias para el funcionamiento del prototipo.

Asimismo, se definieron claves primarias y claves foráneas para garantizar la integridad y consistencia de la información relacionada con eventos culturales, categorías, ubicaciones, usuarios, roles y permisos.

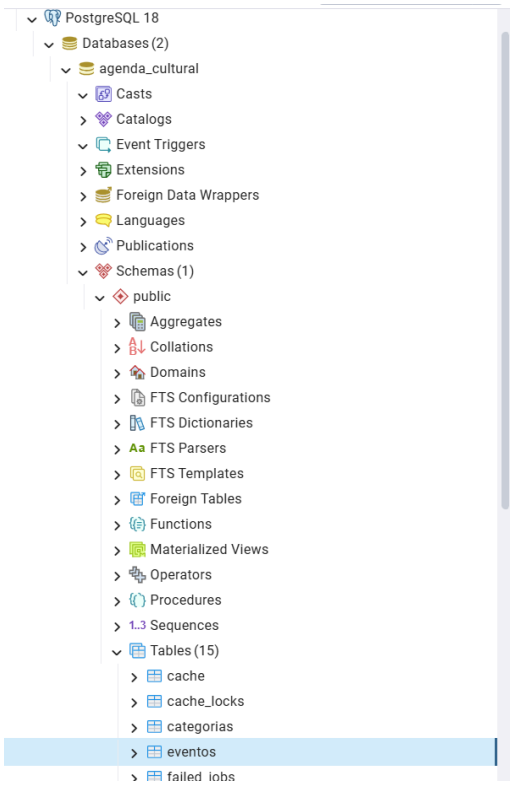


Figura 6. *Captura de pantalla de base de datos PostgreSQL "agenda_cultural"*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

3.3.2 Desarrollo con el Framework Laravel

El desarrollo del prototipo se realizó aprovechando el framework Laravel y el lenguaje de

programación PHP, implementando el patrón arquitectónico Modelo-Vista-Controlador (MVC) para organizar la lógica de negocio, la interacción con la base de datos y las interfaces del sistema de manera estructurada y segura.

Modelos: Se desarrollaron modelos como Evento, Categoria, Ubicacion, Usuario y Rol, representando las tablas de la base de datos y facilitando la ejecución de operaciones CRUD (Crear, Leer, Actualizar y Eliminar) dentro del sistema.

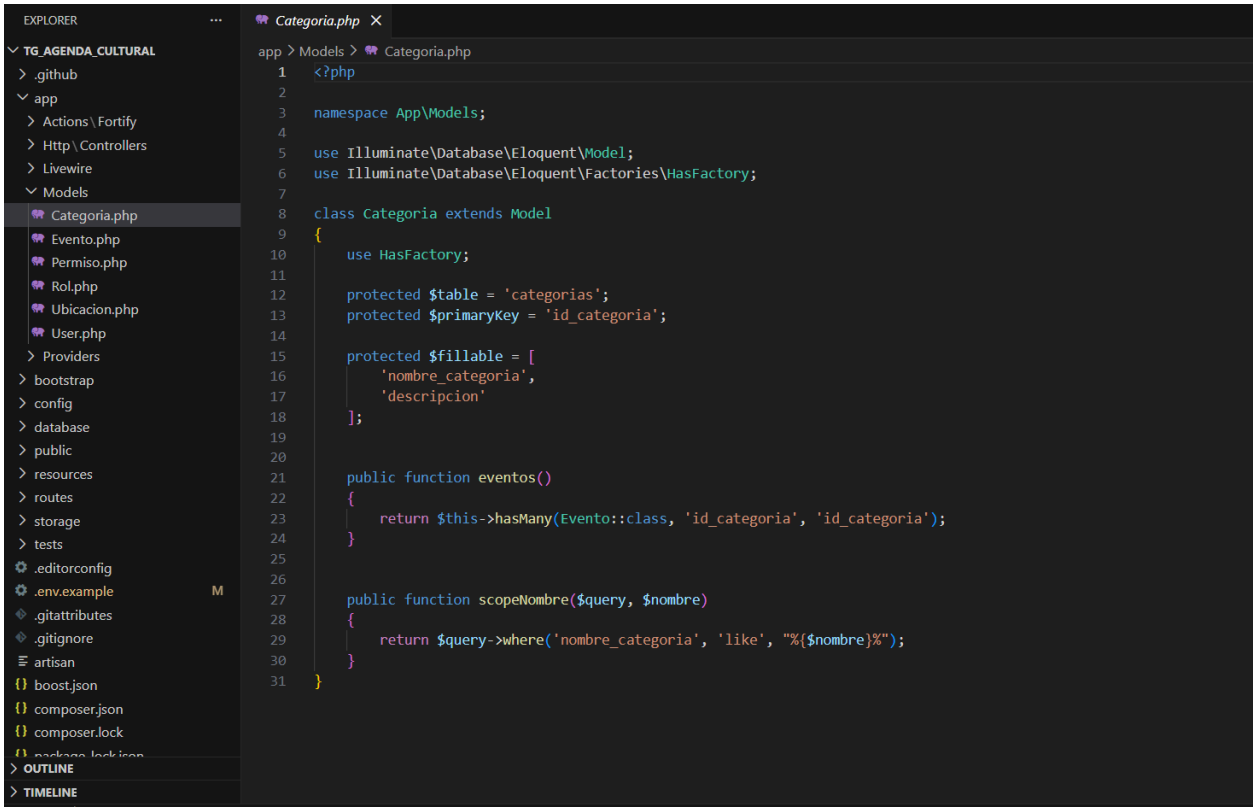


Figura 7. *Captura de pantalla de Models del prototipo Agenda cultural.*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Vistas: Se desarrollaron interfaces de usuario mediante las plantillas Blade de Laravel, utilizando tecnologías como HTML5, Tailwind CSS y JavaScript para la construcción de

módulos frontend y administrativos con diseño responsive y accesible.

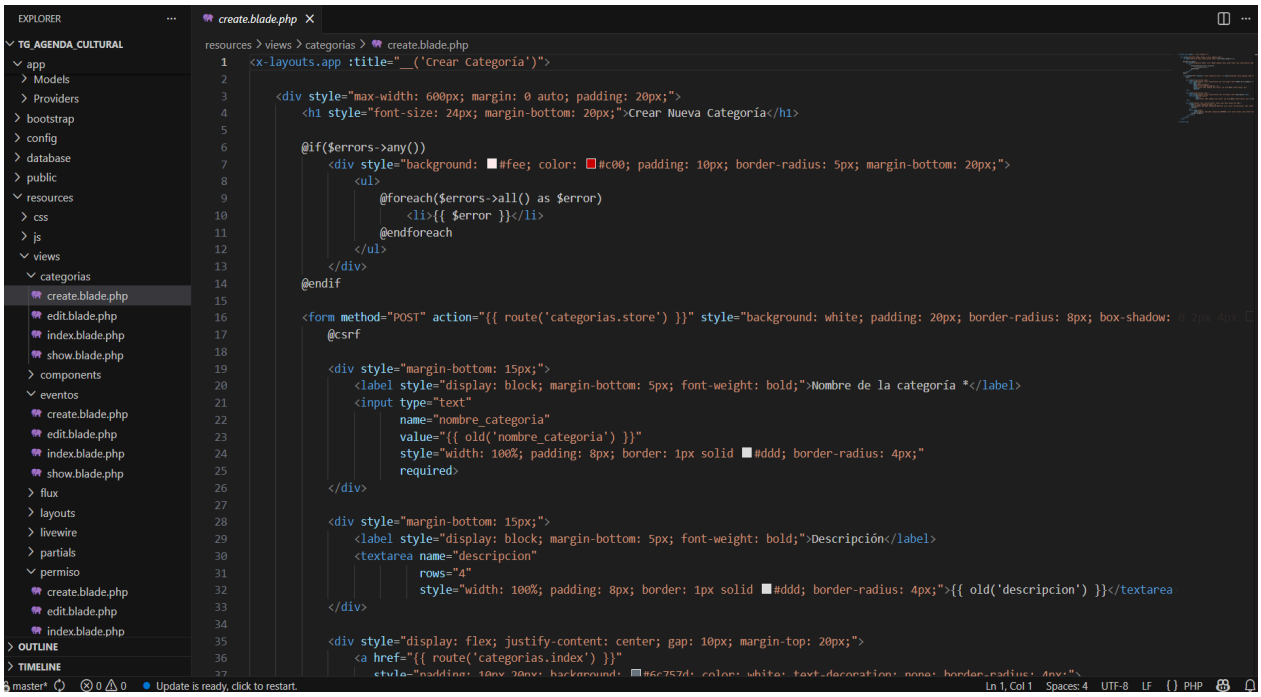


Figura 8. *Captura de pantalla de Vistas del prototipo Agenda cultural.*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Controladores: Se implementaron controladores encargados de gestionar las peticiones HTTP, aplicar la lógica de negocio e interactuar con los modelos y vistas del sistema, permitiendo administrar la información cultural de manera organizada.

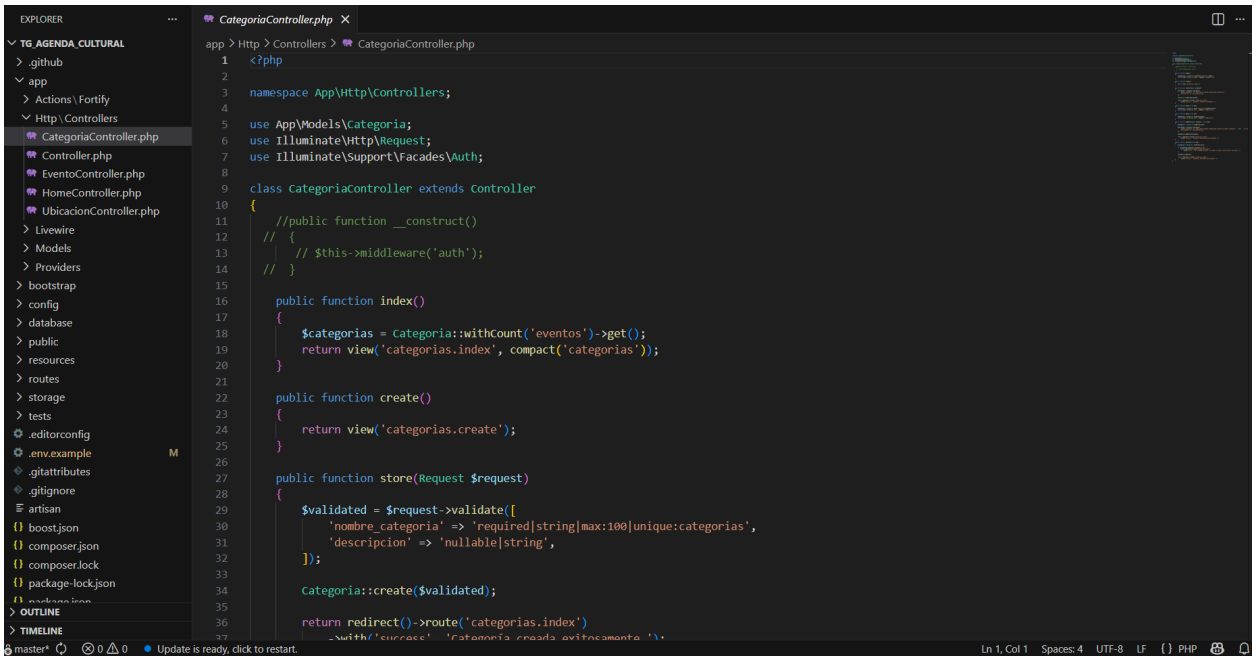


Figura 9. *Captura de pantalla de los Controladores del prototipo Agenda cultural.*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Autenticación: Se implementó el sistema de autenticación de Laravel para el inicio de sesión de administradores y usuarios autorizados, incorporando validación de credenciales, control de acceso y gestión de permisos dentro del sistema.

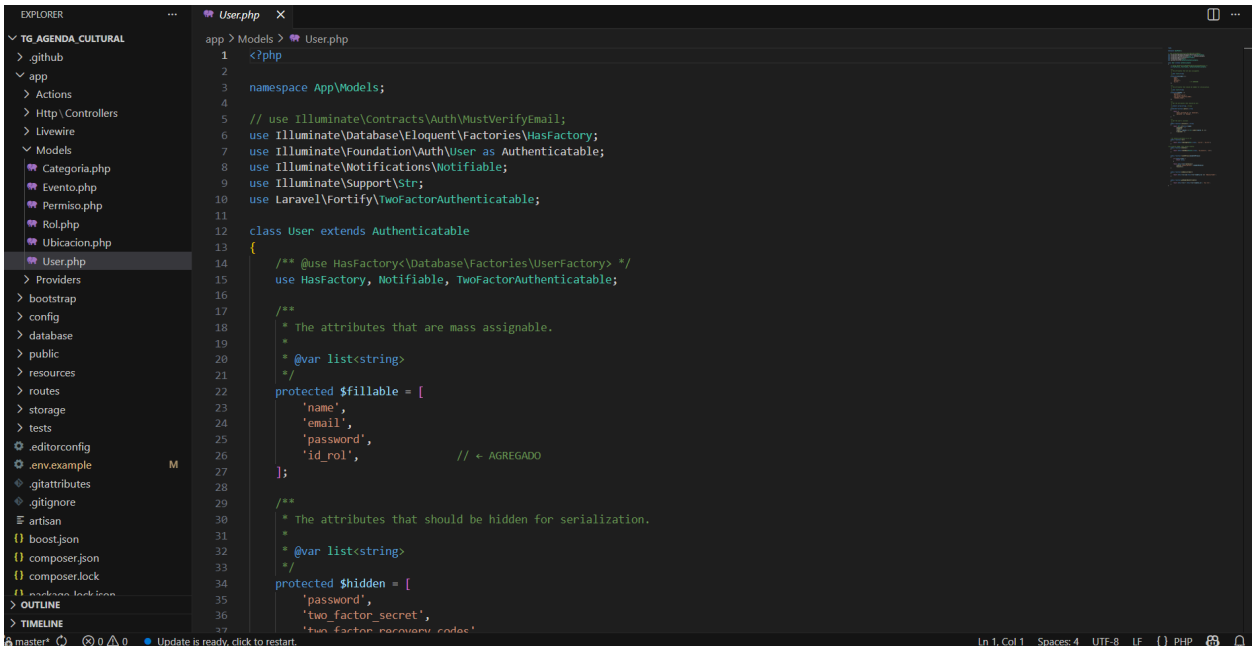


Figura 10. *Captura de pantalla de la autenticación del prototipo Agenda cultural.*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

APIs y Servicios: Se desarrollaron endpoints y servicios internos para facilitar la comunicación entre el frontend y el backend, permitiendo la transferencia y consulta de información relacionada con eventos culturales, categorías y ubicaciones dentro del sistema.

Repositorio: Se estableció como repositorio GitHub, ya que es la mejor opción para este prototipo por su integración con control de versiones, facilidad para trabajo colaborativo y despliegue continuo. El enlace al repositorio es el siguiente: https://github.com/jrs0118/TG_agenda_cultural

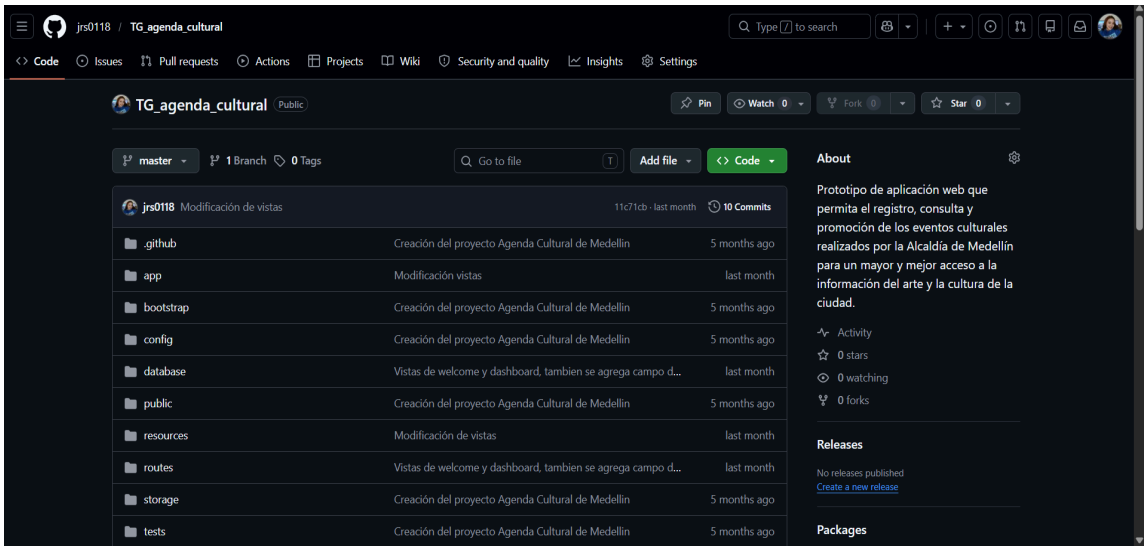


Figura 11. *Captura de pantalla del repositorio GitHub del prototipo Agenda cultural.*
Nota. Captura de pantalla del repositorio GitHub del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia. https://github.com/jrs0118/TG_agenda_cultural

3.4 Fase 4: Entregar

La fase final se enfocó en la validación, despliegue y entrega formal del prototipo funcional de la Agenda Cultural de Medellín, con el propósito de evaluar el correcto funcionamiento del sistema y garantizar su disponibilidad para consulta y administración de eventos culturales.

3.4.1 Testing

Se realizó un proceso de validación funcional y técnica del prototipo mediante pruebas unitarias, pruebas de integración y consultas sobre la base de datos PostgreSQL, permitiendo verificar el correcto funcionamiento de cada uno de los módulos desarrollados.

Asimismo, se ejecutaron pruebas de gestión y consulta de información cultural mediante sentencias SQL con operaciones de filtrado, agrupamiento y relaciones entre tablas, evaluando la

capacidad del sistema para administrar múltiples eventos culturales y generar información estadística relacionada con categorías, ubicaciones y agenda cultural.

Entre las pruebas realizadas se incluyeron: Inserción masiva de registros de eventos culturales.

Verificación del sistema de autenticación y control de acceso:

Tabla 25 *Pruebas del sistema de autenticación.*

#	Prueba	Resultado esperado	Resultado obtenido	Estado
1	Inicio de sesión correcto	Una vez iniciada la sesión, el sistema se redirige al dashboard.	El usuario accede correctamente al al dashboard	Aprobado
2	Inicio de sesión incorrecto	Mostrar un mensaje de error	Se mostró "These credentials do not match our records"	Aprobado
3	Cierre de sesión	Al cerrar sesión, el sistema se redirige a la página principal de la Agenda cultural.	El usuario fue redirigido correctamente	Aprobado

Nota. Elaboración propia

Prueba 1: Inicio de sesión correcto

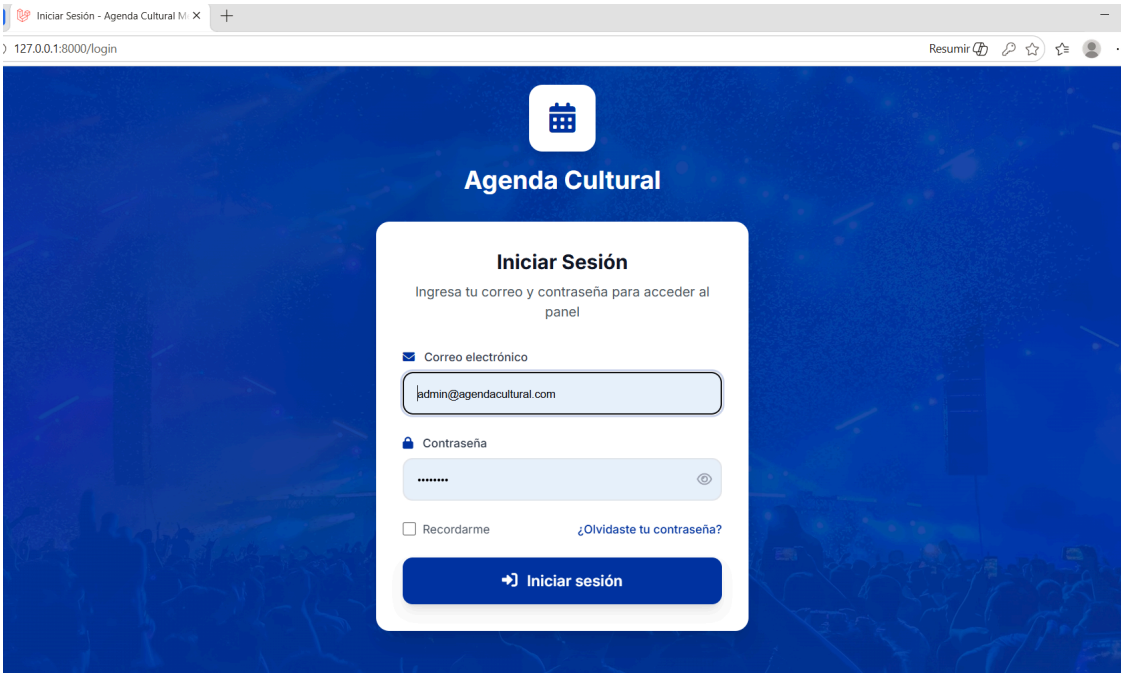


Figura 12. *Captura de inicio de sesión "login" con los datos ingresados*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio).

Elaboración propia.

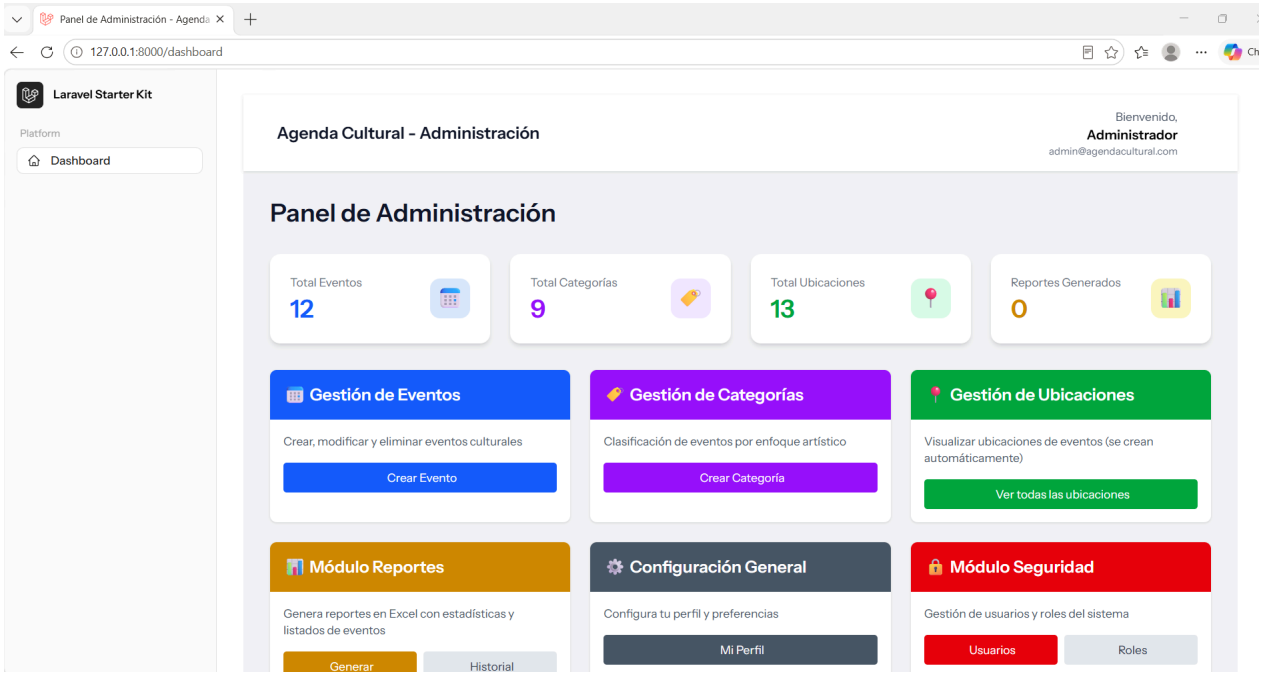


Figura 13. *Captura de Dashboard después del "login"exitoso.*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio).
Elaboración propia.

Prueba 2: Inicio de sesión incorrecto

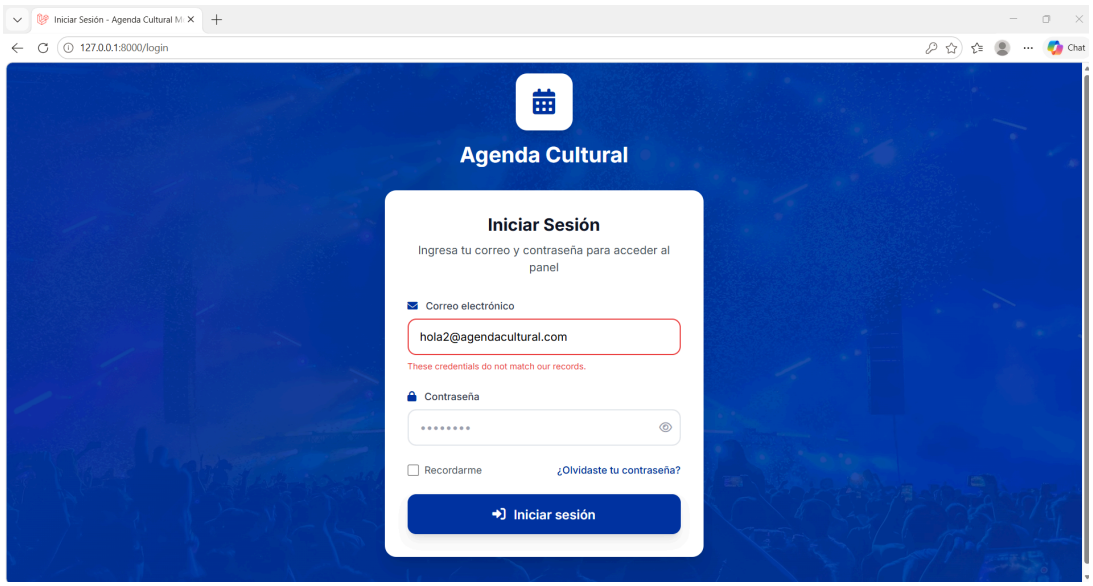


Figura 14. *Captura de inicio de sesión "login" con mensaje de error "These credentials do not match our records"*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio).
Elaboración propia.

Prueba 3: Cierre de sesión

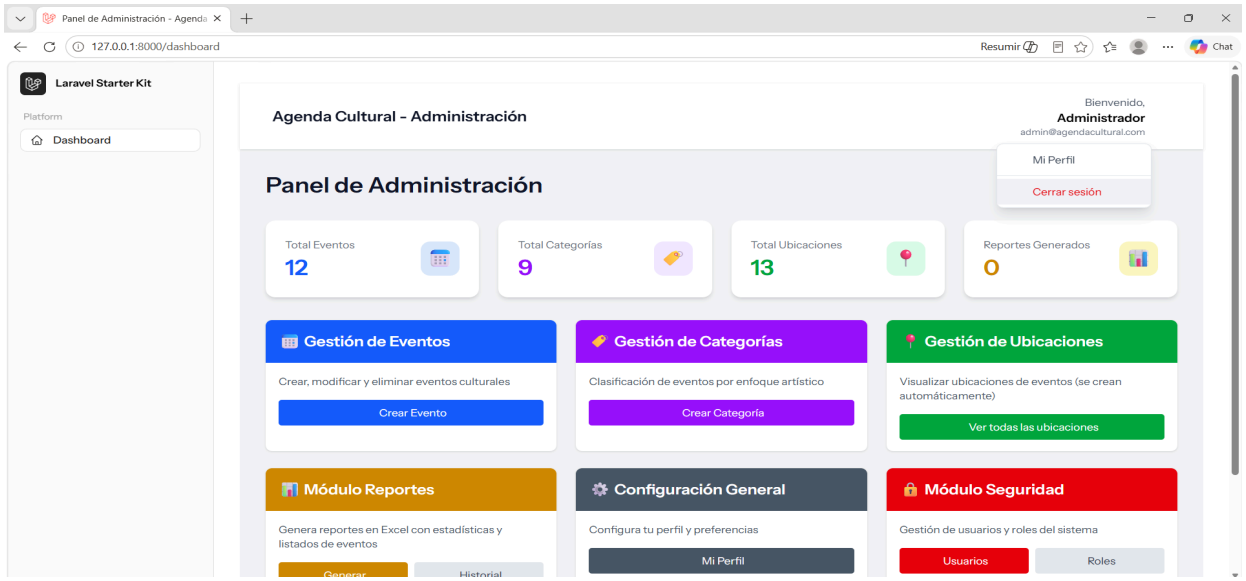


Figura 15. *Captura del Dashboard, menú de usuario y cerrar sesión*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

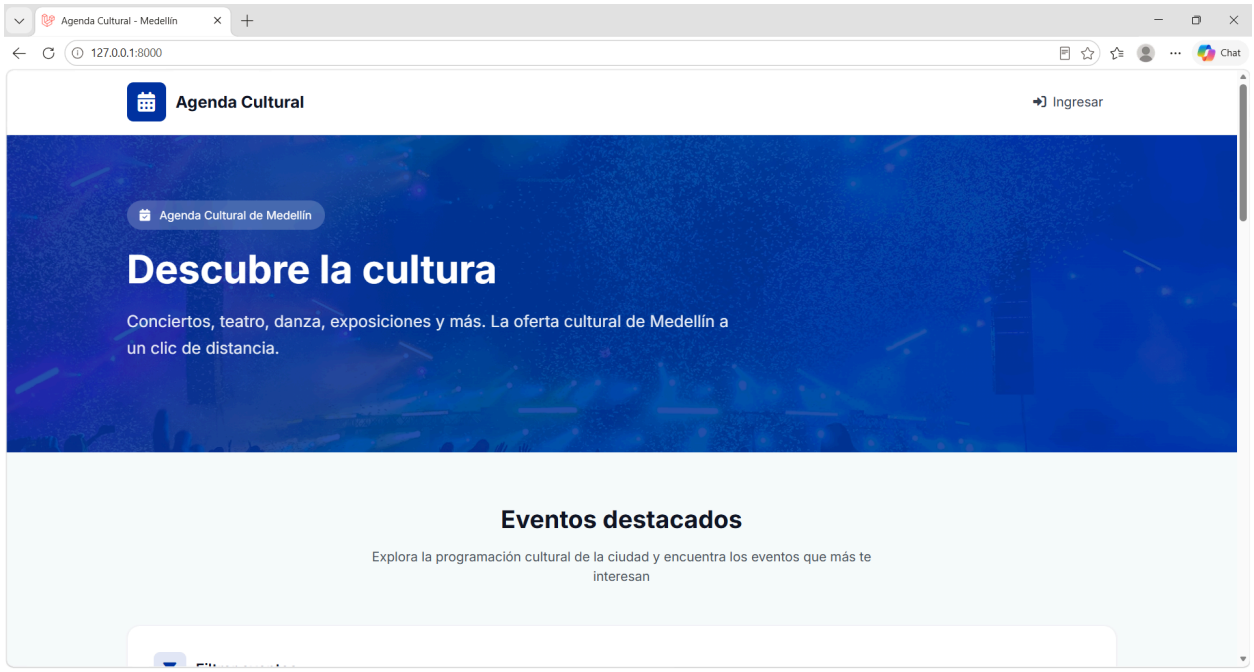


Figura 16. *Captura de la página principal de la Agenda Cultural después de cerrar sesión.*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio).

Elaboración propia.

Inserción masiva de eventos en la base de datos en PostgreSQL

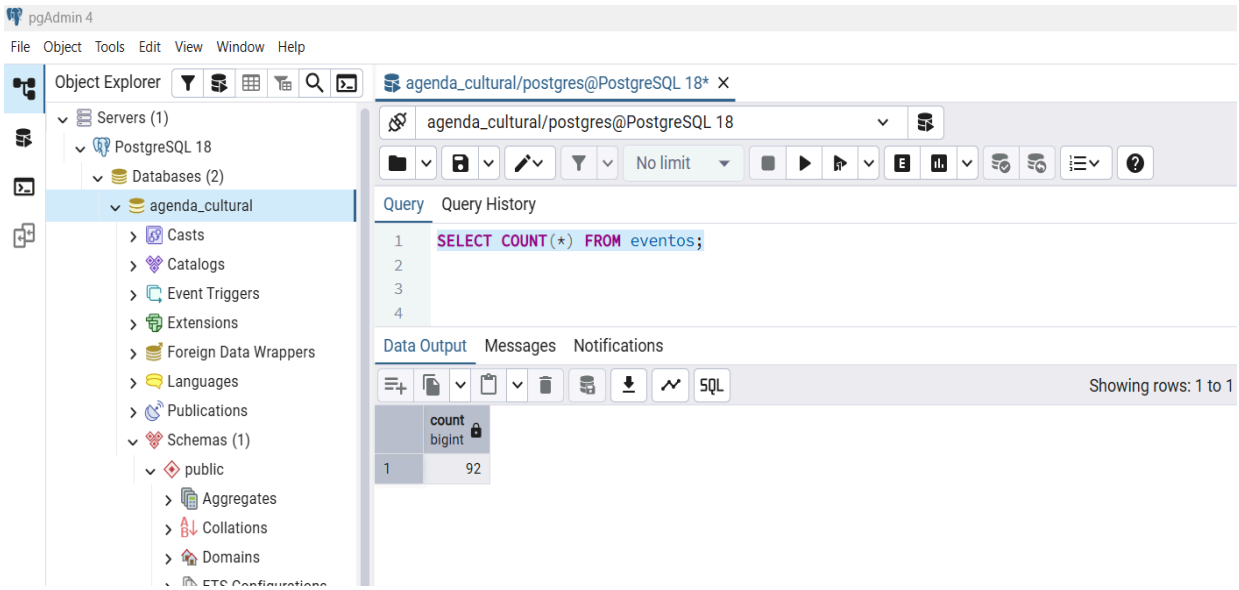


Figura 17. *Captura de la base de datos de Agenda Cultural de Medellín en PostgreSQL antes de insertar 100 registros.*
Nota. Captura de pantalla de la base de datos del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

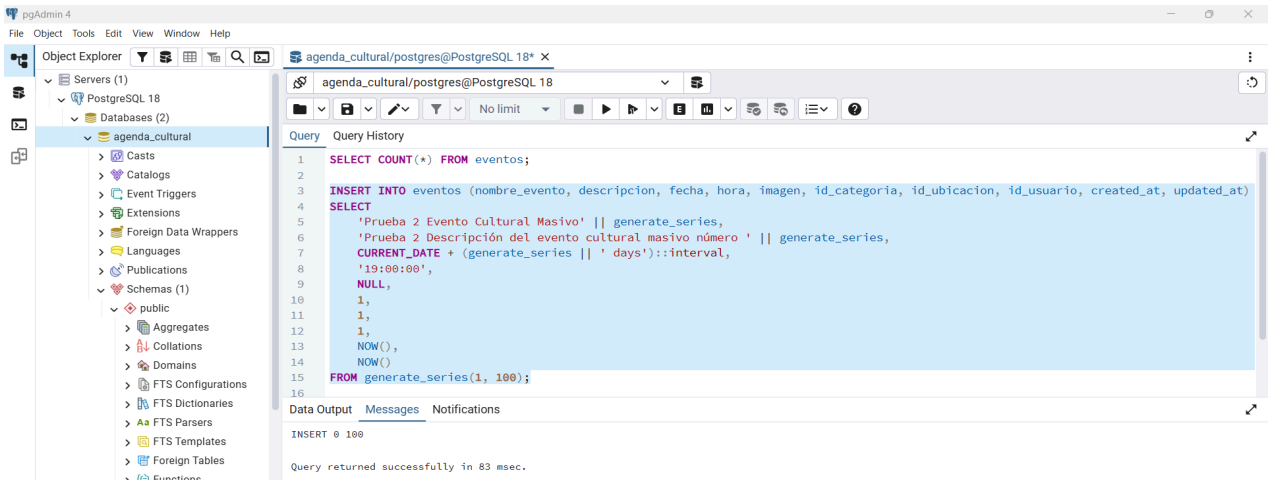


Figura 18. *Captura de la base de datos de Agenda Cultural de Medellín en PostgreSQL durante la ejecución de la inserción masiva de registros. El proceso fue ejecutado en dos ocasiones, permitiendo la inserción de un total de 200 eventos culturales.*
Nota. Captura de pantalla de la base de datos del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

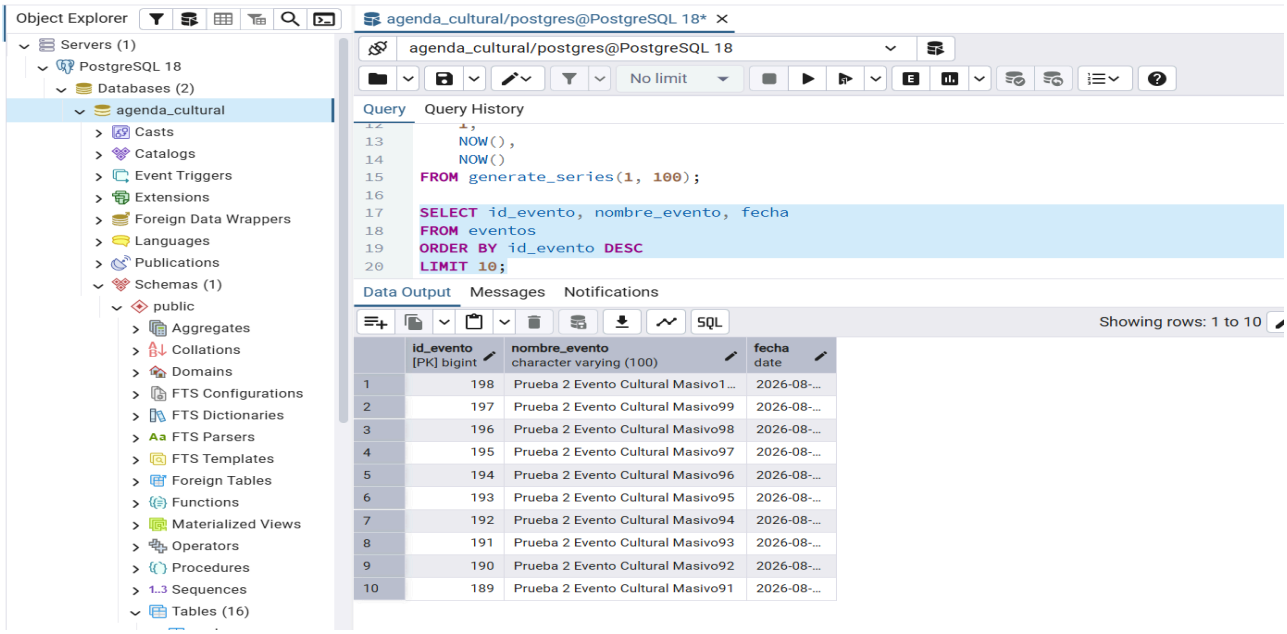


Figura 19. *Captura de la base de datos de Agenda Cultural de Medellín en PostgreSQL*

Nota. Captura de pantalla de la base de datos del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

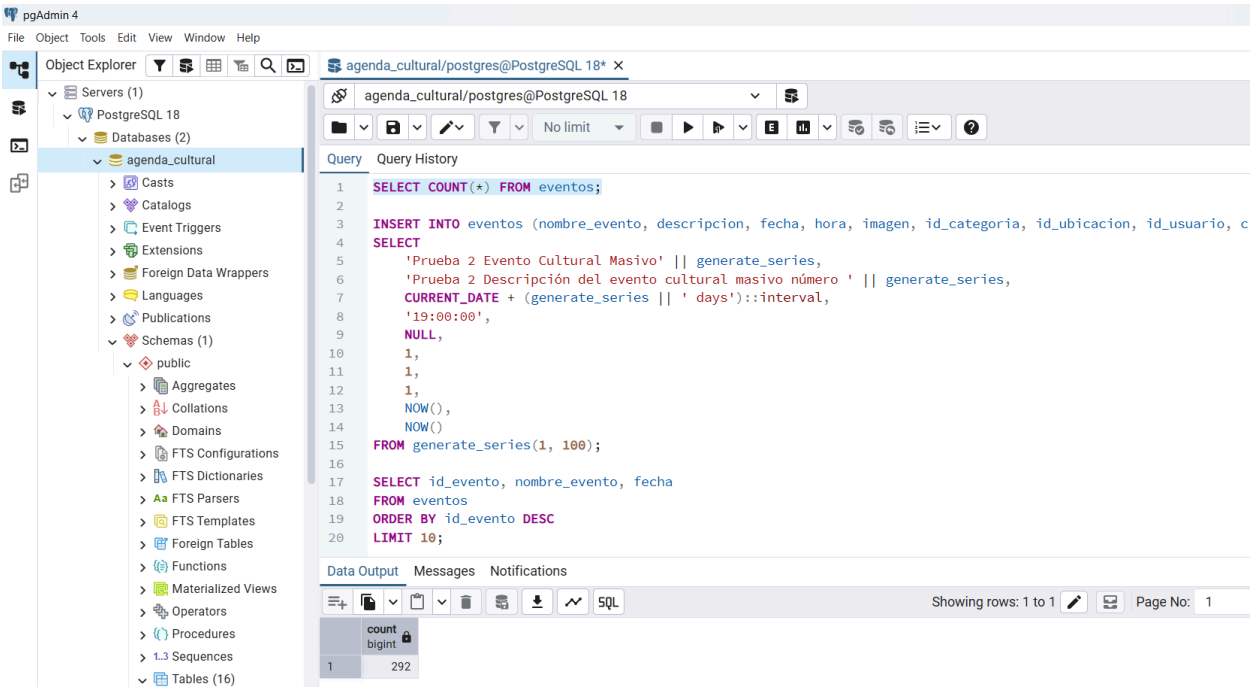


Figura 20. Captura de la base de datos de Agenda Cultural de Medellín en PostgreSQL con el número de datos insertados

Nota. Captura de pantalla de la base de datos del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

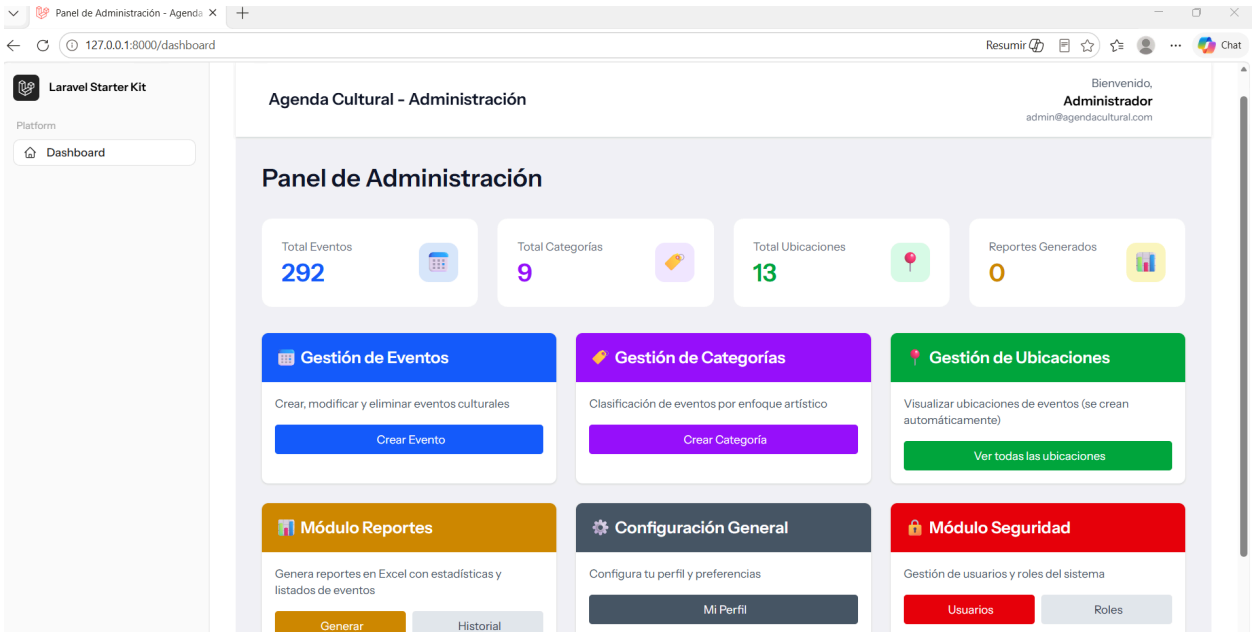


Figura 21. *Captura de Dashboard con el número de eventos insertados a través de la base de datos masivamente.*

Nota. Captura de pantalla de la base de datos del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Validación de operaciones CRUD sobre eventos, categorías y ubicaciones.

Tabla 26 *Pruebas del sistema de autenticación.*

#	Prueba	Resultado esperado	Resultado obtenido	Estado
1	Crear evento	Evento registrado en la base de datos	El evento aparece creado	Aprobado
2	Consultar eventos	Tabla con todos los eventos	Se visualizan correctamente los eventos	Aprobado
3	Editar evento	Datos modificados correctamente	Los cambios se guardaron exitosamente	Aprobado
4	Eliminar evento	Evento eliminado del sistema	El evento ya no se encuentra en la tabla	Aprobado

Elaboración propia.

Prueba 1: Crear evento

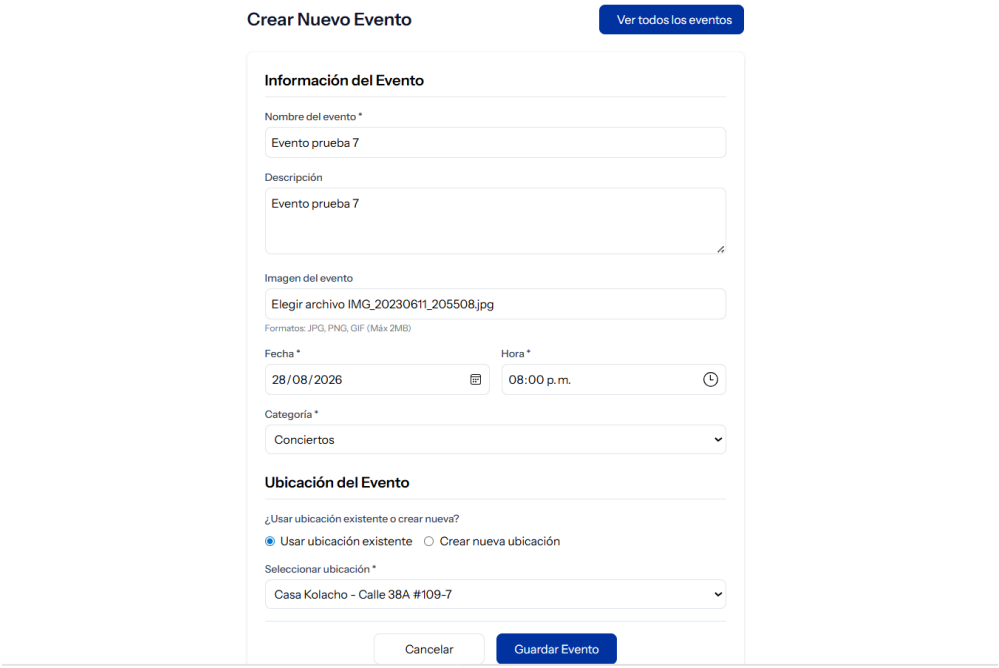


Figura 22. *Captura del módulo Eventos.*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

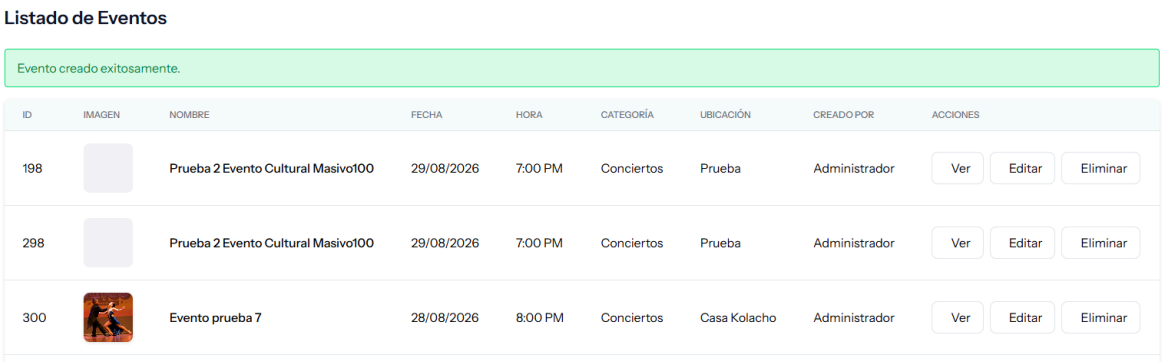


Figura 23. *Captura de evento creado correctamente*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Prueba 2: Listar eventos

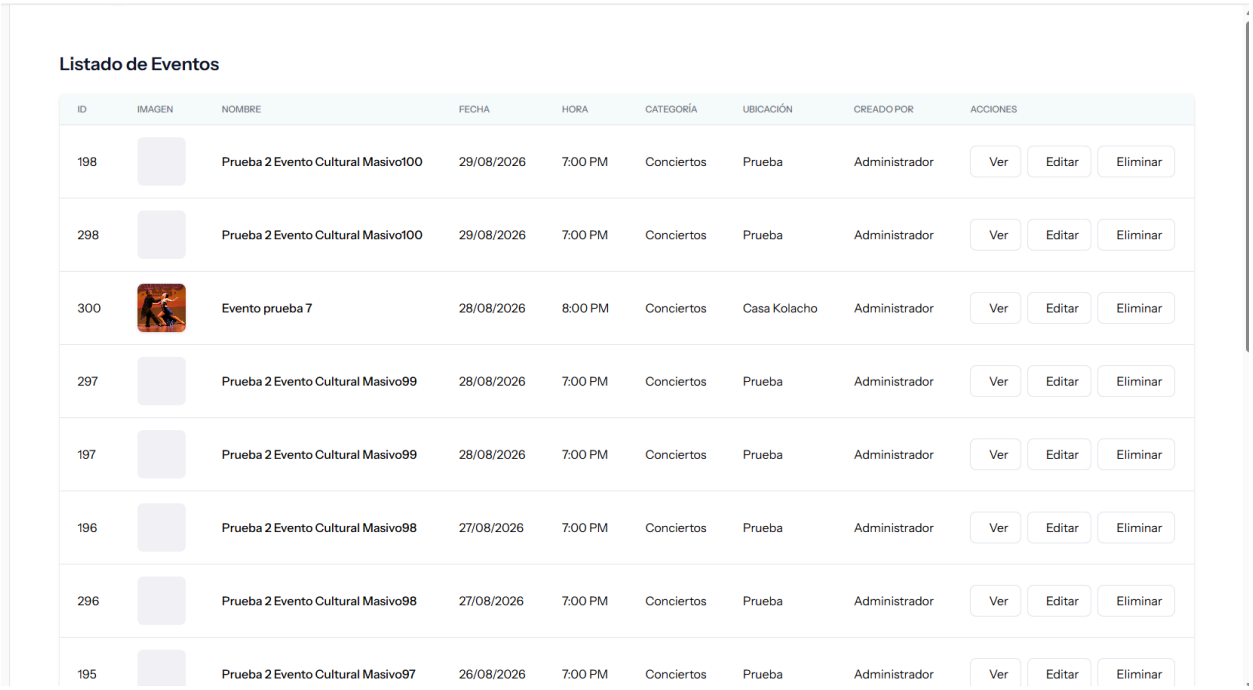


Figura 24. *Captura de listar los eventos*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Prueba 3: Editar eventos

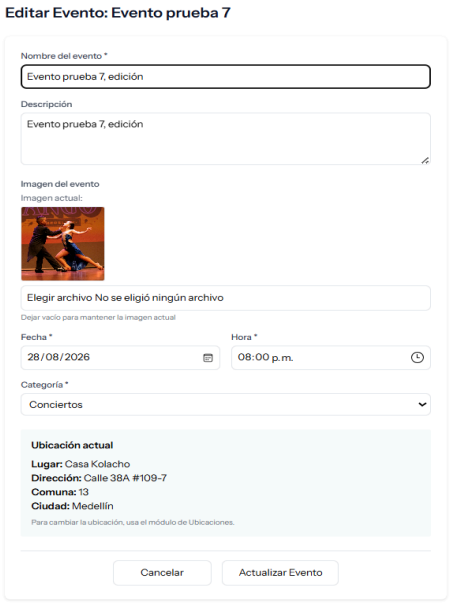


Figura 25. *Captura de editar eventos*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

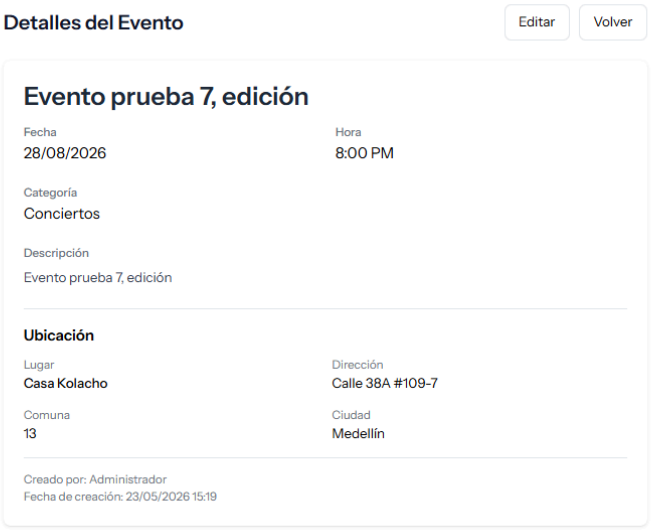


Figura 26. *Captura de evento modificado*

Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio).

Elaboración propia.

Prueba 4: Eliminar evento

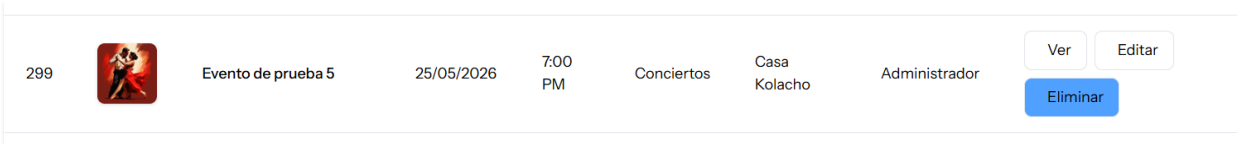


Figura 27. *Captura de eliminar evento*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

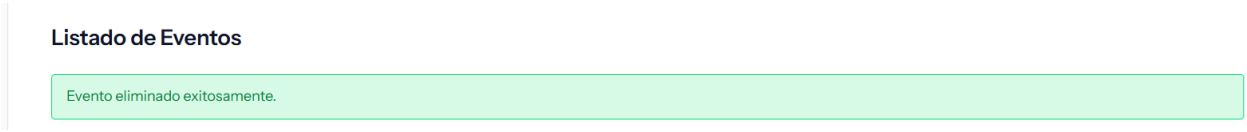


Figura 28. *Captura evento eliminado exitosamente*
Nota. Captura de pantalla del prototipo Agenda cultural de Medellín (proyecto propio). Elaboración propia.

Estas pruebas permitieron validar el cumplimiento de los objetivos relacionados con el registro, consulta y gestión eficiente de la información cultural dentro del prototipo.

3.4.2 Despliegue

El despliegue del prototipo se realizó en un entorno local. La instalación inicial requiere conexión a internet para clonar el repositorio desde GitHub y descargar las dependencias necesarias (Composer y npm).

Una vez instalado, el prototipo puede ejecutarse en entorno local, permitiendo el acceso a funcionalidades como gestión de eventos, generación de reportes y visualización de la agenda cultural.

El proceso completo de despliegue y demostración del prototipo puede consultarse en el

siguiente video demostrativo

Video en Youtube “prototipo de sistema para la agenda cultural”: https://youtu.be/-rVOcT-yxtI

3.4.3 Documentación del proyecto

Como parte de la entrega final del prototipo, se generó la documentación técnica y funcional asociada al sistema:

Manual Técnico: Incluye la descripción de la arquitectura del sistema, configuración del entorno de desarrollo, instalación del proyecto, estructura de la base de datos y mantenimiento general del sistema.

Manual de Usuario: Guía dirigida a administradores, gestores culturales y usuarios finales, explicando el funcionamiento de las principales funcionalidades del sistema, incluyendo registro, consulta y administración de eventos culturales.

Documentación en GitHub Wiki: Como complemento de la documentación técnica del prototipo, se implementó una Wiki en GitHub asociada al repositorio del proyecto, con el propósito de centralizar información relacionada con la instalación, configuración del entorno, arquitectura del sistema, estructura de la base de datos y funcionamiento de los módulos desarrollados.

Dentro de la Wiki se documentaron aspectos relacionados con migraciones de Laravel, arquitectura MVC, manual de instalación, estructura de módulos y configuración de la base de datos PostgreSQL.

6. Conclusiones

El desarrollo del prototipo de la Agenda Cultural de Medellín permitió cumplir con el objetivo general planteado en el proyecto, orientado al registro, consulta y promoción de eventos culturales realizados por la Alcaldía de Medellín. A partir del levantamiento de requerimientos fue posible identificar las necesidades relacionadas con el acceso centralizado a la información cultural de la ciudad, lo que facilitó la definición del alcance, módulos y funcionalidades del sistema, dando cumplimiento al primer objetivo específico.

Asimismo, el diseño de la arquitectura funcional y de interfaz permitió estructurar un sistema organizado en capas, integrando frontend, backend y base de datos para garantizar una adecuada gestión de la información cultural y una experiencia de usuario amigable, cumpliendo con el objetivo relacionado con el diseño del prototipo.

De igual manera, la implementación del sistema mediante Laravel, PHP y PostgreSQL permitió desarrollar funcionalidades relacionadas con gestión de eventos, categorías, ubicaciones, autenticación, reportes y administración de usuarios, dando cumplimiento al objetivo específico asociado al desarrollo del prototipo según los requerimientos definidos.

El uso de tecnologías como Blade, Tailwind CSS y JavaScript contribuyó al desarrollo de interfaces accesibles, responsive e interactivas para la consulta de eventos culturales. Además, la base de datos relacional desarrollada en PostgreSQL permitió gestionar de manera organizada la información correspondiente a eventos culturales, categorías, ubicaciones, usuarios y permisos, garantizando la integridad y consistencia de los datos.

Por otra parte, las pruebas funcionales y consultas SQL realizadas permitieron evaluar y

validar el correcto funcionamiento del sistema, evidenciando su capacidad para gestionar múltiples registros y generar consultas estadísticas relacionadas con la información cultural, cumpliendo así con el objetivo específico de evaluación del prototipo.

Adicionalmente, la publicación del código fuente, la documentación técnica y la Wiki del proyecto en GitHub facilitó el control de versiones, la disponibilidad del sistema y la posibilidad de futuras mejoras o ampliaciones, permitiendo contar con un repositorio de acceso público para la consulta y continuidad del desarrollo del prototipo.

7. Recomendaciones

Se recomienda continuar con el fortalecimiento y ampliación del prototipo Agenda Cultural de Medellín mediante la implementación de nuevas funcionalidades que permitan mejorar la experiencia de los usuarios y optimizar la gestión de la información cultural de la ciudad. Una de las principales recomendaciones consiste en realizar el despliegue del sistema en un entorno web público o en la nube, permitiendo el acceso desde diferentes dispositivos y ampliando el alcance de la plataforma hacia la ciudadanía.

De igual manera, se considera importante que futuras implementaciones del prototipo permitan su disponibilidad a través de internet, facilitando el acceso público y centralizado a la información de eventos culturales por parte de ciudadanos, gestores culturales y entidades interesadas. También podría integrarse un sistema de notificaciones o recordatorios de eventos mediante correo electrónico o dispositivos móviles para fortalecer la interacción con los usuarios.

Asimismo, se recomienda que en futuras versiones el sistema permita que un mismo evento pueda desarrollarse en múltiples ubicaciones y horarios diferentes, facilitando la programación de eventos culturales itinerantes o con varias funciones dentro de la ciudad, sin necesidad de registrar el evento repetidamente. Igualmente, incorporar herramientas de búsqueda y consulta más eficientes para los administradores y gestores culturales, permitiendo localizar eventos de manera más ágil sin depender exclusivamente de filtros por fecha, optimizando así la gestión y administración de la información cultural registrada en la plataforma.

Se sugiere fortalecer los mecanismos de seguridad mediante la implementación de controles adicionales de autenticación, auditoría y respaldo de la información.

8. Referencias bibliográficas

8.1 Bibliografía

Alcaldía de Medellín - Secretaría de Cultura Ciudadana. (2018). Página Agenda Cultural de Medellín. *Página Agenda Cultural de Medellín*. Medellín.

Alcaldía de Medellín - Secretaría de Cultura Ciudadana. (2025). Plan Decenal de Cultura 2025-2035. En *Capítulo 2. Las apuestas del Plan: Componente estratégico* (págs. 24-34). Medellín.

Alcaldía Mayor de Bogotá. (2025). *Agenda Cultural de Bogotá*. Obtenido de Bogotá: https://bogota.gov.co/que-hacer/agenda-cultural

Cancino Velásquez, J. A. (2012). *Cultura digital. Web 2.0 y herramientas tecnológicas*. Obtenido de www.gestiopolis.com: http://www.gestiopolis.com/cultura-digital-web20-herramientas-tecnologicas/

Gonzáles Pérez, C. A. (Diciembre de 1999). Sistemas de Información para la Gestión de Recursos Culturales: Teoría, Metodología y Tecnologías. Santiago de Compostela.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación (6a ed.). McGraw-Hill Education.

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Otwell, T. (2023). Laravel Documentation. Laravel LLC. Recuperado de https://laravel.com/docs

Johnson, R., Walls, C., & Syer, M. (2022). Spring Boot Reference Documentation. VMware, Inc. Recuperado de https://docs.spring.io/spring-boot

Royce, W. W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, 26, 1–9.

Design Council. (2005). *The Double Diamond: A universally accepted depiction of the design process*. Design Council UK. Recuperado de https://www.designcouncil.org.uk

Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.

Ramírez Sánchez, J. (2026). Agenda Cultural de Medellín [Repositorio GitHub]. GitHub. https://github.com/jrs0118/TG_agenda_cultural

PostgreSQL Global Development Group. (2025). PostgreSQL Documentation. https://www.postgresql.org/docs/

PROTOTIPO DE SISTEMA PARA LA AGENDA CULTURAL DE LA CIUDAD DE MEDELLÍN

MANUAL DE USUARIO

JUANITA RAMÍREZ SÁNCHEZ

ASESOR
JAIME ERNESTO SOTO URDANETA

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2026

1. Introducción

La Agenda Cultural de Medellín es una plataforma web diseñada para centralizar y facilitar la consulta y gestión de eventos culturales de la ciudad. El sistema permite a los ciudadanos consultar eventos culturales, mientras que los administradores y gestores culturales pueden administrar la información registrada en la plataforma.

2. Acceso al sistema

- Abra su navegador de preferencia
- Ingrese la URL : http://127.0.0.1:8000
- Los ciudadanos pueden consultar eventos sin iniciar sesión.
- Los administradores y gestores culturales deben ingresar sus credenciales para acceder al panel de administración.

3. Funcionalidades principales

Ciudadanos

- Consultar evento
- Filtrar eventos por categoría, fecha y ubicación.
- Visualizar información detallada de cada evento cultural disponible.

Administradores y gestores culturales

- Crear, editar y eliminar eventos culturales.
- Administrar categorías y ubicaciones.
- Generar reportes de eventos.
- Gestionar usuarios y roles del sistema.
- Actualizar información de perfil y contraseña.

4. Gestión de eventos

El sistema permite registrar eventos culturales ingresando información como nombre, descripción, categoría, fecha, hora, ubicación e imagen del evento. Además, los usuarios autorizados pueden editar o eliminar eventos previamente registrados.

5. Reportes

El módulo de reportes permite generar listados y resúmenes de eventos culturales utilizando filtros como categoría, ubicación y rango de fechas. Los reportes pueden descargarse en formato Excel.

6. Cierre de sesión

Para finalizar la sesión, el usuario debe seleccionar la opción “Cerrar sesión” desde el menú superior del sistema.

PROTOTIPO DE SISTEMA PARA LA AGENDA CULTURAL DE LA CIUDAD DE MEDELLÍN

MANUAL TÉCNICO

JUANITA RAMÍREZ SÁNCHEZ

ASESOR

JAIME ERNESTO SOTO URDANETA

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

MEDELLÍN

2026

1. Tecnologías utilizadas

El prototipo Agenda Cultural de Medellín fue desarrollado utilizando Laravel 10 como framework backend, Blade y Tailwind CSS para la interfaz de usuario y PostgreSQL como sistema gestor de base de datos.

2. Requisitos mínimos

Software	Versión
PHP	8.1 o superior
Composer	Última versión
PostgreSQL	14 o superior
Node.js	npm

3. Instalación paso a paso del prototipo Agenda Cultural de Medellín

- **Clonar el repositorio**
  git clone https://github.com/jrs0118/TG_agenda_cultural.git
  cd TG_agenda_cultural

- **Instalar dependencias**
  composer install
  npm install

- **Instalar dependencias**
  copy .env.example .env
  php artisan key:generate

- **Configurar el entorno editanto el archivo .env**

- **Ejecutar migraciones y seeders:**
  php artisan migrate
  php artisan db:seed --class=RolesYPermisosSeeder

- **Crear enlace de las imágenes:**
  php artisan storage:link

- **Iniciar el servidor**
  php artisan serve
  URL local: http://127.0.0.1:8000

4. Estructura principal del prototipo

El sistema está organizado en controladores, modelos, migraciones, vistas Blade, rutas y configuraciones generales. Además, cuenta con módulos para eventos, categorías, ubicaciones, reportes y seguridad.

5. Rutas principales

URL	Función
/	Página principal
/login	Inicio de sesión
/dashboard	Panel de control
/eventos	Gestión de eventos
/reportes	Generación de reportes
/seguridad	Gestión de usuarios
/configuracion	Perfil y configuración

6. Comandos útiles

Comando	Descripción
php artisan serve	Iniciar servidor
php artisan migrate	Crear tablas en BD
php artisan db:seed	Cargar datos iniciales
php artisan storage:link	Habilitar imágenes
php artisan view:clear	Limpiar caché de vistas

7. Credenciales de acceso

Correo	admin@agendacultural.com
Contraseña	password