

Tabla de contenido

**Descripción del proyecto**

**Objetivos de proyecto**

**Alcance del proyecto**

**Análisis de viabilidad**

*-Viabilidad técnica*

*-Viabilidad económica*

**Análisis de requerimientos**

**Requerimientos de interfaz**

Requerimientos funciona**les**

**Requerimientos no funcionales**

**Maquetación**

**Diseño**

*-Diagrama de casos de uso*

*-Diagrama de secuencia*

*-Diagrama de clases*

**Descripción del proyecto**

Este proyecto tiene como objetivo desarrollar una plataforma integral y fácil de usar para la búsqueda, compra y gestión de entradas para conciertos y eventos en vivo. Con funcionalidades avanzadas para mejorar la experiencia de los usuarios, esta plataforma ofrecerá un servicio completo tanto para los asistentes como para los organizadores de eventos. A continuación, se detallan las principales características y funcionalidades que formarán parte de la plataforma

**Exploración y Búsqueda de Eventos**

Los usuarios podrán buscar conciertos y eventos según diferentes criterios, como el artista, género musical y ubicación, facilitando una búsqueda personalizada y eficiente.

**Información Detallada de Eventos**

* Detalles del Concierto: Cada evento contará con información precisa, incluyendo el nombre del concierto, artista, fecha, lugar y los tipos de boletos disponibles.
* Visualización de Precios y Asientos: Los usuarios podrán ver los precios y la disponibilidad de los asientos en tiempo real. Además, se proporcionará un mapa interactivo del lugar para elegir y visualizar los asientos disponibles de manera intuitiva.
* Contenido Multimedia: Cada artista contará con videos, fotos y una biografía que enriquecerán la experiencia del usuario, proporcionando un contexto más completo sobre el evento.

**Carrito de Compras y Gestión de Entradas**

* Los usuarios podrán agregar múltiples entradas al carrito y revisar o modificar su pedido antes de realizar la compra.
* Se ofrecerán entradas digitales con código QR, lo que permitirá un acceso rápido y seguro a los eventos sin necesidad de llevar boletos físicos.

**Gestión de Usuario y Personalización**

* Perfiles de Usuario: Los usuarios podrán registrarse para crear su perfil personal, donde se almacenarán sus preferencias, datos personales y métodos de pago, facilitando futuras compras y mejorando la experiencia general.

**Seguridad y Privacidad**

Para garantizar la seguridad, la plataforma implementará un sistema de autenticación de dos factores, asegurando que solo los usuarios autorizados puedan acceder a su cuenta y realizar transacciones.

Los organizadores de eventos tendrán acceso a funcionalidades específicas según sus roles, permitiendo gestionar eventos, entradas y ventas de manera controlada y segura.

**Soporte al Cliente**

Los usuarios podrán dejar **comentarios y valoraciones** sobre su experiencia de compra, lo que ayudará a mejorar continuamente la plataforma y proporcionará información útil para futuros compradores.

**Objetivos de proyecto**

**Objetivo Específico:**

Implementar un proceso de compra de entradas sencillo y eficaz, permitiendo a los usuarios seleccionar, revisar y comprar múltiples boletos para eventos de su interés, todo ello con un carrito de compras intuitivo y opciones de pago seguras.

**Objetivos Generales:**

* Proveer una plataforma intuitiva y eficiente que permita a los usuarios buscar conciertos y otros eventos en vivo según sus preferencias (artista, género, ubicación) y realizar la compra de entradas de manera sencilla y segura.
* Brindar una experiencia adaptada a las preferencias de cada usuario, permitiéndoles almacenar su información personal, métodos de pago y gestionar sus entradas, todo ello con altos estándares de seguridad y protección de datos.
* Proporcionar a los organizadores de eventos herramientas que les permitan crear, gestionar y vender entradas para conciertos y otros eventos, con un sistema de control de acceso basado en roles para garantizar un manejo seguro y eficiente.

**Alcance del proyecto**

* Nombre del producto: TicketVerse
* Tipo de producto: Sistema de venta de tickets
* Versión: 1.0

Consistencia con definiciones similares: Este documento es coherente con la descripción del Sistema previamente elaborada, donde se detallan las funcionalidades clave como la búsqueda de eventos, la selección de asientos, el procesamiento de pagos y la emisión de tickets digitales. Además, las especificaciones aquí contenidas están alineadas con los requisitos de negocio y las pautas establecidas en documentos de mayor nivel, garantizando que el sistema cumpla con las expectativas de los organizadores de eventos y de los usuarios finales.

**1. Requerimientos Funcionales:**

**Gestión de Eventos:**

Los usuarios podrán buscar conciertos por artista, fecha y lugar.

Los conciertos mostrarán detalles como el lugar, la fecha y los boletos disponibles.

**Proceso de Compra:**

Confirmación de compra

Registro y Autenticación de Usuarios:

Los usuarios podrán registrarse e iniciar sesión para gestionar sus compras.

**Interfaz de Usuario (UI):**

Diseño responsivo que funcione en móviles, tabletas y computadoras.

**2. Requerimientos No Funcionales:**

**Rendimiento**:  
La página web debe soportar una carga considerable, especialmente durante la venta de boletos para eventos populares. Se debe permitir el acceso simultáneo de al menos 1000 usuarios, con 500 usuarios conectados durante los picos de venta. El sistema debe manejar 10 transacciones por segundo en momentos de mayor tráfico, asegurando que el 95% de las transacciones se completen en menos de 1 segundo y que el tiempo de carga de la página no exceda los 2 segundos en condiciones normales.

**Fiabilidad**:  
La fiabilidad del sistema es clave para la satisfacción del usuario. No se deben permitir más de 5 incidentes críticos al mes.

**Disponibilidad**:  
Se requiere un tiempo de disponibilidad del 99.5% mensual, garantizando que los usuarios puedan acceder y realizar compras sin interrupciones.

**Mantenibilidad**:  
El sistema debe ser fácil de mantener, con actualizaciones de software y gestión de servidores realizadas semanalmente para estadísticas y mensualmente para mejoras de seguridad y rendimiento. Un equipo especializado será responsable de estas tareas.

**Portabilidad**:  
Para facilitar la migración a otros entornos, menos del 10% del sistema y un máximo del 20% del código deben depender del servidor. Se utilizarán tecnologías portátiles como JavaScript, HTML5 y Node.js, asegurando compatibilidad con sistemas operativos como Windows.Principio del formularioFinal del formulario

**Entregables:**

Prototipo de la interfaz de usuario

Aplicación web funcional para la compra de boletos.

**Exclusiones del Proyecto:**

No se desarrollará una aplicación móvil nativa para iOS o Android en esta fase.

No habrá integración con plataformas de terceros (como redes sociales).

Soporte de idiomas adicionales fuera del inglés no estará disponible

**Análisis de viabilidad**

***Viabilidad técnica***

* Frontend: Tecnologías como HTML, CSS, JavaScript.
* Backend: Lenguajes y frameworks para la lógica del servidor (Node.js)
* Integración de Pagos: Plataformas de pago para tarjetas.
* Escalabilidad:Capacidad de la web para manejar grandes volúmenes de tráfico durante lanzamientos o promociones.

***Capacidades del Equipo***

* Programadores web con experiencia en frontend y backend.
* Diseñadores UI/UX para una experiencia de usuario atractiva.
* Especialistas en seguridad informática.
* Administradores de sistemas para la gestión del servidor.

***Identificación de Riesgos Técnicos***

Sobrecarga del servidor en eventos populares.

Brechas de seguridad y protección de datos de usuarios.

Problemas de integración con sistemas de pago y terceros.

***Viabilidad económica***

* Desarrollo Web (Frontend y Backend)\*: $80,000
* Diseño de Interfaz de Usuario (UI/UX)\*: $20,000
* Integración de Pasarelas de Pago\*: $15,000
* Capacitación y Soporte Técnico Inicial\*: $10,000
* Total de Costos de Desarrollo: \*$125,000\*\*

***Costos de Infraestructura***

* Servidores en la Nube (Almacenamiento y Procesamiento): $20,000 anuales
* Servicios de Seguridad y Encriptación de Datos: $10,000 anuales
* Ancho de Banda y Escalabilidad para Altas Demandas: $5,000 anuales
* Total de Costos de Infraestructura: $35,000 anuales

***Costos de Mantenimiento***

* Actualización y Mantenimiento de Software: $15,000 anuales
* Soporte Técnico y Atención al Cliente: $8,000 anuales
* Optimización de SEO y Marketing Digital: $7,000 anuales
* Total de Costos de Mantenimiento: $30,000 anuales
* 500,000\* anuales
* Costos Iniciales\*: $125,000 (incluye desarrollo, diseño y capacitación).
* Costos Anuales Recurrentes\*: $65,000 (incluye infraestructura y mantenimiento).
* Proyección de Ingresos Anuales\*: $500,000.

Por cada dólar invertido, se espera recuperar aproximadamente $2.28

Análisis de requerimientos

-Requerimientos de interfaz

**Diseño General**

**Estilo Visual:**

* Colores: Se utilizarán tonos neutros como blanco y gris, con acentos en azul para los botones y títulos.
* Tipografía: Se empleará una fuente legible como Arial o Verdana.

**Disposición:**

* La página estará dividida en secciones fijas: búsqueda de eventos, selección de asientos, resumen de compra y confirmación.

**Elementos de la Página**

**1. Barra de Navegación:**

* Enlaces a las secciones: Inicio, Eventos, Mi Cuenta, Ayuda.
* Un cuadro de búsqueda para encontrar conciertos por artista o fecha.

**2. Sección de Búsqueda de Eventos:**

* Listado de conciertos en formato de tabla, con columnas para el artista, fecha y precio.
* Cada fila representará un evento, sin interactividad en la selección.

**3. Selección de Asientos:**

* Presentación de una lista de asientos disponibles con su respectivo precio, sin mapa interactivo.
* Los usuarios seleccionarán los asientos marcándolo.

**4. Resumen de Compra:**

* Una sección simple que muestra los boletos seleccionados, el precio total y un botón para proceder al pago.
* Opción de volver a la lista de eventos para modificar la selección.

**5. Proceso de Pago:**

* Campos fijos para ingresar información del usuario: nombre, correo electrónico y método de pago.
* Un botón de "Comprar" para finalizar la compra.

**6. Confirmación de Compra:**

* Página de confirmación con un mensaje de agradecimiento y un resumen de la compra.
* Opción para imprimir los boletos directamente desde la página.

-Requerimientos funcionales

### Gestión de Conciertos

El sistema debe permitir a los usuarios buscar conciertos por:

* Artista o banda.
* Género Musical.
* Fecha del concierto.

### Página del concierto.

Cada concierto debe tener una página detallada con:

* Nombre del evento.
* Artista o banda.
* Fecha y hora.
* Lugar.
* Precios de las entradas.
* Disponibilidad de asientos.
* Opción de ver próximos eventos o conciertos destacados.

### Gestión de Usuarios.

El sistema debe permitir a los usuarios registrarse y crear una cuenta:

* Nombre.
* Correo Electrónico.
* Contraseña.

Los usuarios deben poder:

* Descargar o imprimir las entradas.
* Recibir notificaciones sobre eventos futuros o sugerencias.

### Proceso de Compra.

Los usuarios deben poder seleccionar:

* El número de entradas.
* La ubicación o los asientos.
* El sistema debe permitir varios métodos de pago como: Tarjeta de crédito/Débito/Transferencias.
* Al comprar la entrada saltara un código QR.
* Enviar confirmación de la compra vía correo electrónico con los detalles del evento y un enlace para descargar las entradas.

-Requerimientos no funcionales

La página web debe ser capaz de soportar una carga considerable, especialmente durante la venta de boletos para eventos populares. Los requisitos específicos incluyen:

**- Número de terminales:** La web debe permitir el acceso simultáneo de al menos 1000 usuarios.

**- Número de usuarios simultáneamente conectados**: Se espera que al menos 500 usuarios estén conectados al mismo tiempo durante los picos de venta.

**- Número de transacciones por segundo:** El sistema debe manejar al menos 10 transacciones por segundo, especialmente en las horas de mayor tráfico, como el lanzamiento de boletos.

**- Tiempo de respuesta**: El 95% de las transacciones deben realizarse en menos de 1 segundo, y el tiempo de carga de la página no debe exceder los 2 segundos en condiciones normales.

**3.3.2 Fiabilidad**

La fiabilidad del sistema es crucial para garantizar la satisfacción del usuario. Se especifican los siguientes factores:

**- Total de incidentes permisibles:** No se deben permitir más de 5 incidentes de fallos críticos al mes.

**3.3.3 Disponibilidad**

El sistema debe garantizar una alta disponibilidad para permitir a los usuarios comprar boletos en cualquier momento. Los requisitos son:

**- Disponibilidad:** El sistema debe tener un tiempo de disponibilidad del 99.5% en un ciclo mensual, asegurando que los usuarios puedan acceder y realizar compras sin interrupciones.

**3.3.4 Mantenibilidad**

La mantenibilidad del sistema es fundamental para asegurar su funcionamiento continuo. Se definen los siguientes aspectos:

**- Tipo de mantenimiento:** El mantenimiento del sistema debe incluir actualizaciones periódicas del software y la gestión de servidores.

**- Responsables del mantenimiento:** Las tareas de mantenimiento deberán ser realizadas por un equipo de desarrolladores especializado y administradores de sistemas.

**- Frecuencia de mantenimiento:** Las tareas de mantenimiento deben realizarse semanalmente para generar estadísticas de acceso y mensualmente para actualizaciones de seguridad y rendimiento.

**3.3.5 Portabilidad**

Para facilitar la migración a otras plataformas o entornos, se deben considerar los siguientes atributos:

**- Porcentaje de componentes dependientes del servidor**: Se espera que menos del 10% del sistema sea dependiente del servidor.

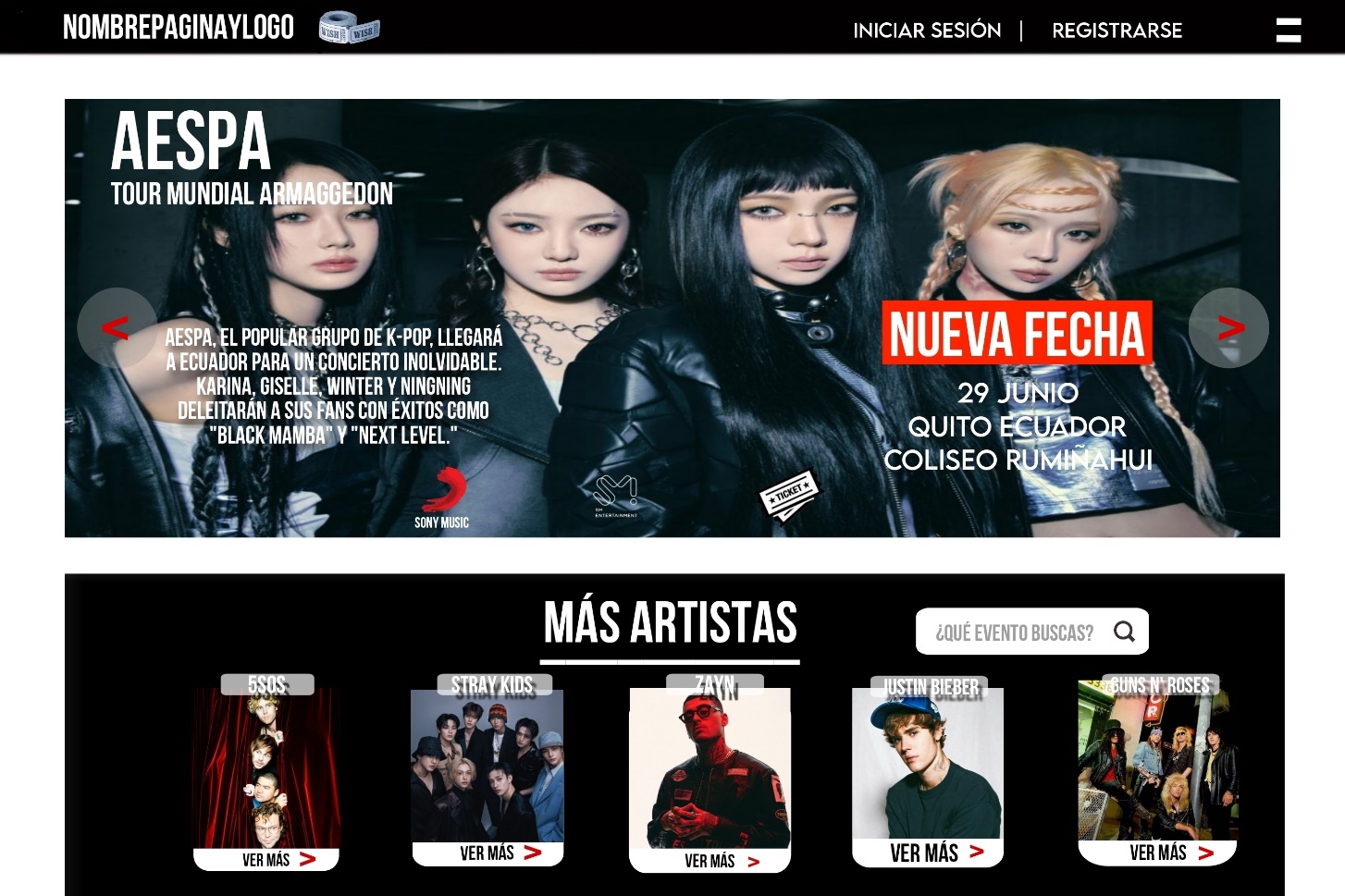
**- Porcentaje de código dependiente del servidor**: El código que dependa de características específicas del servidor no debe exceder el 20%.

**- Uso de un determinado lenguaje:** Se utilizará JavaScript y HTML5 debido a su alta portabilidad entre diferentes navegadores y dispositivos.

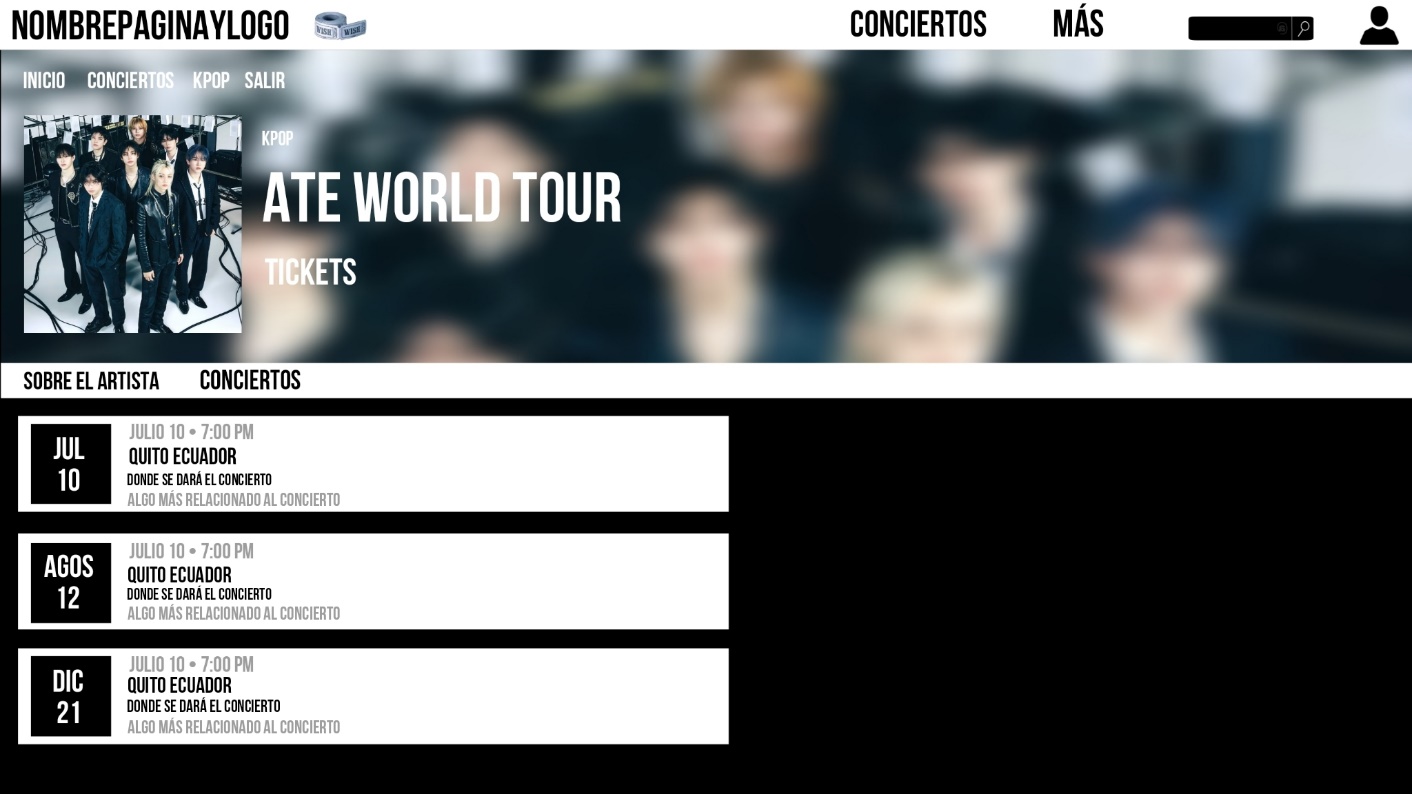
**- Uso de un determinado compilador o plataforma de desarrollo:** Se recomienda el uso de Node.js para el backend debido a su portabilidad y capacidad de manejo de múltiples conexiones simultáneas.

**- Uso de un determinado sistema operativo:** El sistema debe ser compatible con algunos sistemas operativos (Windows) para su desarrollo y ejecución.

**Maquetación**

****

****

****

****

****

Diseño Diagrama de casos de uso

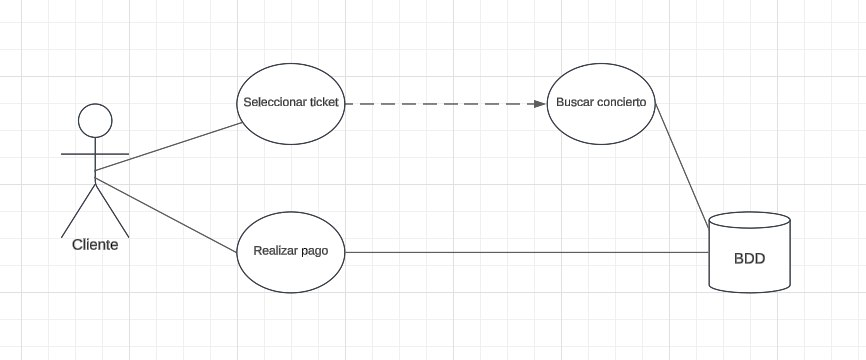


Diagrama de secuencia

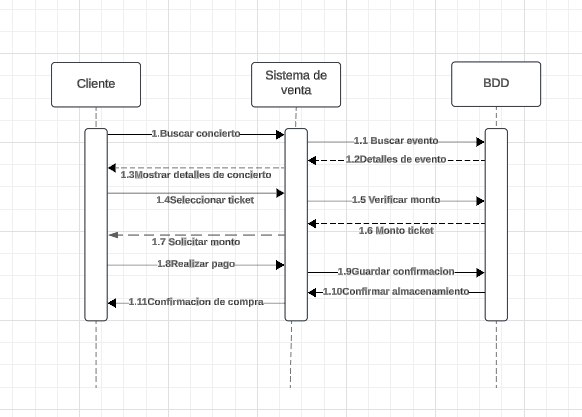
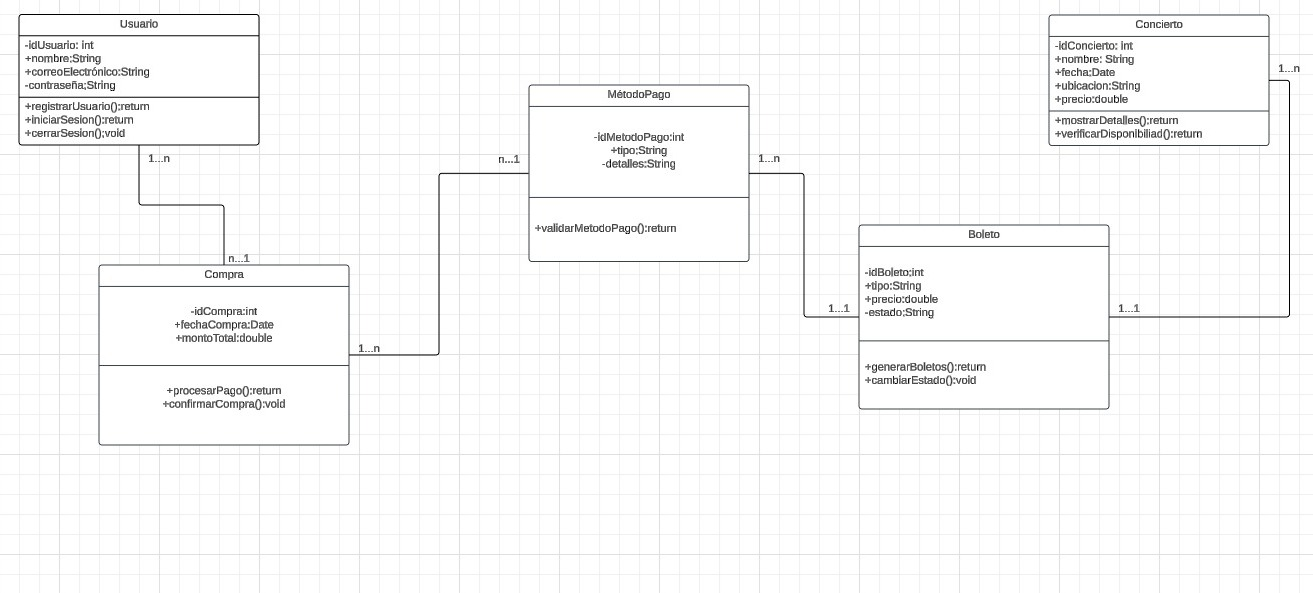


Diagrama de clases



**Implementación**

**Fase inicio**

**Visión**

Crear una plataforma de compra de boletos en línea que ofrezca una experiencia de usuario optimizada y personalizada para la adquisición de entradas a eventos y conciertos. La plataforma debe ser rápida, intuitiva y accesible desde cualquier dispositivo, permitiendo a los usuarios encontrar, comprar y gestionar sus boletos de manera segura y eficiente.

El propósito es que los usuarios puedan acceder fácilmente a eventos y conciertos de su interés, sin la necesidad de navegar entre múltiples sitios y opciones. La plataforma ofrecerá una interfaz amigable que facilite el proceso de compra y ofrezca opciones de personalización, con el objetivo de mejorar la experiencia de compra y optimizar el tiempo del usuario.

**Equipo scrum**

**Desarrollador Front-End (Joel Romero):** Especialista en interfaces de usuario, responsable de implementar el registro de usuario y la visualización de eventos, garantizando una experiencia intuitiva y fluida.

**Desarrollador Back-End (Cristian Sangoluisa):** Encargado de desarrollar las APIs necesarias para el registro y la exploración de eventos, gestionar la infraestructura del servidor y mantener la conexión entre la base de datos y el front-end.

**Especialista en Soporte Técnico (Diego Falcones):** Responsable de la configuración del entorno de pruebas y despliegue, asegurando que el sistema funcione correctamente y asistiendo en la resolución de problemas técnicos durante el proceso de pruebas.

**Tester/QA (Dany Fernández):** Su responsabilidad es garantizar la calidad del producto mediante pruebas continuas de las funcionalidades desarrolladas, asegurando que se minimicen errores y se mejore la usabilidad en el flujo de registro y exploración de eventos.

**Scrum Master (Sabine Vela):** Facilita las reuniones diarias (Daily Standups) y la Sprint Review, apoyando al equipo en el seguimiento del progreso, la eliminación de impedimentos, y manteniendo una comunicación efectiva y fluida en todas las fases del sprint.

**Product backlog**

El Product Owner (María Fernanda López) crea un backlog inicial basado en la visión del producto y en reuniones con los stakeholders, que incluyen organizadores de eventos, promotores de conciertos y usuarios finales. Algunos de los elementos iniciales del Product Backlog son:

**Registro y Autenticación de Usuarios**

* Implementar la funcionalidad de registro de usuario.
* Desarrollar la funcionalidad de inicio de sesión.
* Validar y autenticar usuarios para gestionar sus compras.

**Interfaz de Usuario (UI)**

* Diseñar una interfaz de usuario responsiva compatible con móviles, tabletas y computadoras.
* Asegurar que la UI sea intuitiva y amigable para facilitar la navegación de los usuarios.

**Gestión de Eventos**

* Implementar el sistema de búsqueda de conciertos (filtro por artista, fecha y lugar).
* Mostrar detalles de cada concierto, incluyendo lugar, fecha y boletos disponibles.

**Proceso de Compra**

* Crear el flujo de compra de boletos con carrito de compras.
* Implementar la confirmación de compra y los detalles de los boletos adquiridos.

**Rendimiento**

* Optimizar la página para soportar usuarios simultáneos y 10 transacciones por segundo en momentos de mayor tráfico.
* Asegurar que el tiempo de carga de la página no supere los 2 segundos en condiciones normales.

**Fiabilidad**

* Establecer sistemas de monitoreo y alertas para mantener la cantidad de incidentes críticos bajo el límite de 5 incidentes por mes.
* Configurar mecanismos de recuperación y redundancia para mitigar fallos críticos.

**Disponibilidad**

* Configurar la infraestructura para alcanzar un tiempo de disponibilidad del 99.5% mensual.
* Implementar soluciones de backup y failover para asegurar la continuidad del servicio.

**Mantenibilidad**

* Configurar un flujo de trabajo para que el equipo de mantenimiento ejecute estas actualizaciones de manera eficiente.

**Portabilidad**

* Utilizar tecnologías portátiles como JavaScript, HTML5 y Node.js para facilitar la migración a otros entornos.

**Criterios de éxito**

**Desarrollo del MVP en el plazo de tres meses:** La plataforma debe lanzarse en 12 semanas con una versión mínima que permita a los usuarios registrarse, explorar eventos y comprar boletos de manera segura. Esta versión debe incluir funcionalidades básicas de búsqueda de eventos, selección de asientos, y confirmación de compra.

**Engagement de los usuarios beta:** Durante la fase de prueba, al menos 300 usuarios deben registrarse en la plataforma y completar la compra de boletos. El objetivo es identificar posibles áreas de mejora en la experiencia de usuario, seguridad de pago y estabilidad general de la aplicación.

**Retención del usuario durante las primeras semanas:** El 70% de los usuarios que interactúan con la plataforma deben realizar más de una búsqueda o compra en los primeros 30 días. Esto permitirá evaluar la efectividad del diseño en la retención y satisfacción del cliente.

**Satisfacción del cliente:** Un 80% de los usuarios beta deben reportar una experiencia de usuario positiva en cuanto a la facilidad de navegación, la rapidez del proceso de compra y la relevancia de las recomendaciones de eventos. Los comentarios de los usuarios también deben reflejar altos niveles de seguridad y confianza en el sistema de pago.

**Escalabilidad y estabilidad del sistema**: La plataforma debe soportar picos de tráfico sin afectar el rendimiento, con un máximo de errores críticos del 1% en áreas clave como la finalización de compra y la generación de boletos. Esto garantizará una experiencia sólida y sin interrupciones durante momentos de alta demanda, como lanzamientos de boletos para eventos populares.

**Fase de planificación del sprint**

**Primer sprint**

La empresa **ITTICKET** ha decidido desarrollar una plataforma en línea para la compra de boletos, orientada a ofrecer una experiencia rápida, intuitiva y personalizada para que los usuarios encuentren y compren boletos para eventos de su interés. Esta plataforma está destinada tanto a usuarios casuales como a asistentes frecuentes de eventos.

El Product Owner (Maria Fernanda López) ha priorizado los elementos del Product Backlog, enfocándose en las funcionalidades esenciales para lograr una versión mínima viable (MVP) de la plataforma.

Después de discutir con el equipo de desarrollo, se seleccionaron los siguientes elementos para el próximo sprint:

**Registro de usuario**

* Como usuario, quiero poder registrarme utilizando mi correo electrónico y una contraseña para asegurar una cuenta personalizada.

**Creación de perfil de usuario**

* Como usuario, quiero completar mi perfil con información personal y recibir notificaciones sobre los tipos de eventos y géneros musicales

**Diseño básico de la interfaz**

* Como usuario, quiero que la plataforma tenga una interfaz intuitiva y fácil de usar que me permita navegar entre las opciones de registro, creación de perfil, y

búsqueda de eventos.

**Sprint Goal**

1. Permitir a los usuarios registrarse en la plataforma y explorar los eventos disponibles.

**Estimación del esfuerzo**

|  |
| --- |
| * **Registro de usuario** **Estimación**: 2 días Incluye la implementación del registro básico con correo electrónico |
| * **Navegación de eventos** **Estimación**: 3 días Permite a los usuarios explorar una lista de eventos disponibles con detalles básicos como nombre, fecha y ubicación. |
| * **Diseño básico de la interfaz** **Estimación**: 2 días Se diseña la interfaz principal del registro y la página de eventos, asegurando que sea intuitiva y fácil de usar. |

**Total estimado**: 7 días  
El equipo ha acordado que su capacidad máxima por sprint es de alrededor de 10 – 15 días, por lo que consideran que esta selección de elementos es manejable dentro del sprint de dos semanas.

**Sprint backlog**

**Registro de usuario**

* Implementar registro con correo y contraseña.
* Realizar pruebas unitarias para el módulo de registro.
* Realizar pruebas de integración para las opciones de redes sociales.

**Creación de perfil de usuario**

* Crear formularios para que el usuario ingrese sus habilidades e intereses.
* Conectar el formulario con la base de datos para almacenar el perfil del usuario.
* Validar los datos ingresados por el usuario (campos obligatorios).
* Realizar pruebas de usabilidad en el flujo de creación de perfil.

**Fase de ejecución del Sprint**

**Asignación de tareas al equipo Scrum**

**Product Owner: María Fernanda López**

Asegurar que se definan claramente los requisitos y objetivos del registro de usuario.

* Validar que las funcionalidades implementadas sean consistentes con la visión del producto.
* Participar en la revisión del progreso y retroalimentación del equipo.

**Scrum Master: Sabine Vela**

* Facilitar las reuniones diarias (daily standups) y el seguimiento del progreso del equipo.
* Ayudar a eliminar impedimentos que enfrente el equipo durante el sprint.
* Asegurar que el equipo siga los procesos ágiles y mantener la comunicación fluida.
* **Equipo de Desarrollo**

|  |
| --- |
| **Sabine Vela** |
| * Implementar el registro de usuario con correo y contraseña. |
| * Realizar las pruebas unitarias para el módulo de registro. |
| **Cristian Sangoluisa** |
| * Desarrollar la interfaz básica para la visualización de eventos disponibles. |
| * Integrar la vista de eventos con la base de datos para mostrar los eventos actuales. |
| **Diego Falcones** |
| * Crear las pruebas unitarias y de integración para la funcionalidad de visualización de eventos. |
| * Realizar pruebas de usabilidad en el flujo de registro y eventos. |

**Desarrollo Incremental**

**Product Owner: Maria Fernanda López**

* **Dia 1**: Asegurarse de que los requisitos de registro sean claros y se alineen con las necesidades del usuario.
* **Dia 2**: Validar que la funcionalidad de registro con correo, Google y Facebook esté implementada correctamente y que cumpla con la visión del producto.
* **Dia 3**: Participar en las revisiones de progreso con el equipo para asegurar que el producto esté avanzando como se espera.

**Scrum Master: Sabine Vela**

* **Dia 1**: Facilitar las reuniones diarias (daily standups) para evaluar el progreso.
* **Dia 2**: Ayudar a eliminar impedimentos que el equipo pueda encontrar al implementar las opciones de registro.
* **Dia 3**: Asegurarse de que el equipo siga los procesos ágiles, manteniendo la comunicación fluida entre todos.

**Equipo de Desarrollo**

**Sabine Vela**

**Dia 1**: Implementar el registro de usuario básico con correo y contraseña.

**Dia 2**: Realizar las pruebas unitarias para el módulo de registro, asegurando que todo funcione según lo esperado.

**Cristian Sangoluisa**

* **Dia 1**: Desarrollar la interfaz básica para la visualización de eventos disponibles, asegurando que sea fácil de usar y comprensible.
* **Dia 2**: Integrar la vista de eventos con la base de datos, permitiendo que los usuarios puedan ver los eventos en tiempo real.

**Diego Falcones**

* **Dia 1**: Crear las pruebas unitarias y de integración para la funcionalidad de visualización de eventos, verificando que la lista de eventos se cargue correctamente.
* **Dia 2**: Realizar pruebas de usabilidad en el flujo de registro y visualización de eventos para asegurarse de que los usuarios puedan navegar sin problemas.

**Resolución de impedimentos**

**Falla 1: Error en el registro de usuario con correo electrónico**

* **Descripción**: Los usuarios no pueden completar el registro utilizando correo electrónico porque el sistema muestra un mensaje de error, incluso cuando los datos ingresados son correctos.
* **Resolución**: Dany revisa el código de validación del formulario de registro y descubre que hay un error de sintaxis en el manejo de los datos del formulario. Después de corregir el código, se realizan pruebas y se confirma que ahora los usuarios pueden registrarse correctamente con correo electrónico.

**Falla 2: Problemas de visibilidad en la interfaz del perfil de usuario**

* **Descripción**: La interfaz de creación de perfil no se muestra correctamente en dispositivos móviles; algunos campos de texto se superponen con otros elementos de la interfaz.
* **Resolución**: Cristian ajusta los estilos CSS y realiza modificaciones en el diseño responsivo para que la interfaz se vea correctamente en pantallas de diferentes tamaños. Tras las modificaciones, se realizan pruebas en varios dispositivos móviles y se confirma que la interfaz ahora es funcional y visualmente coherente.

**Fase de revisión y retrospectiva**

**Sprint retrospectiva**

**Lo que funcionó bien:**

* **Comunicación**: El equipo estuvo bien coordinado durante el sprint. Todos sabían lo que estaba haciendo cada uno y se ayudaron mutuamente cuando hubo dudas.
* **Progreso continuo**: Pudimos completar las tareas planificadas, como el registro de usuario y la interfaz básica.

**Lo que se puede mejorar:**

* **Estimación de tiempo**: Algunas tareas tomaron más tiempo de lo esperado, como la integración de las opciones de registro con redes sociales.
* **Pruebas de integración**: Hubo algunos problemas al integrar el registro con las redes sociales que podrían haberse detectado antes si se hubieran hecho más pruebas.

**Acciones para mejorar:**

* **Mejorar la estimación de tiempo**: Deberemos ser más realistas con los tiempos de estimación en tareas complejas.
* **Realizar pruebas de integración antes**: En el siguiente sprint, haremos pruebas de integración más temprano para evitar problemas al final.

REFERENCIAS:

* pmoinformatica.com. (s. f.). *7 Técnicas de levantamiento de requerimientos software*. <https://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>
* *Tutorial de diagrama de clases UML*. (s. f.). Lucidchart. <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>