

1. a

Eventos a tener en cuenta

- Acceso a la aplicación mediante Usuario/Clave
- Lista de asientos disponibles limitados
- Liberar por abonado el asiento que tenga asignado
- Poner tiempo de cancelaciones de 48h mínimo, liberando el asiento
- Si no asiste, bloquear solicitudes de asiento durante dos jornadas,
- Tiempo de respuesta lo menor posible
- Para navegadores Filezilla y Chrome —> mirar conexión con Filezilla (FTP)
- Limitar peticiones a la vez, para no superar el máximo de solicitudes permitidas (conflictos entre dos usuarios pidiendo el mismo asiento a la vez)

Perfiles de usuario

- **Usuarios:**
- Acceso con usuario contraseña a la app
- Gestionar solicitudes de asiento
- Abonos familiares estar próximos a la ubicación de reserva
- **Responsables de gestión de asientos (usuario administrador):**
- Acceso general a la app con usuario y contraseña
- Podrá definir qué asientos están libres
- Asignar asientos según tipo de abono, y puede ser automático
- Consulta de reservas por partido (pintar como quedó la reserva en cada partido y mostrar)
- Envío de confirmación por correo electrónico al usuario (automáticamente al realizar la reserva)
- Envío de correo electrónico a los responsables de la organización del recinto en pdf con la asignación de los asientos.
- Envío de correo electrónico de satisfacción al finalizar el partido a los usuarios.
- **Todo en software libre**
- Primera fase:
- Prototipo funcional con funciones básicas
- Tiempo de respuesta lo menor posible
- Funcionar en Filezilla y Chrome
- No se puede procesar dos peticiones simultáneas

DATOS DE ABONADOS

- N° de abonado
- Familiar si/no
- ID de todos
- Nombre
- Apellidos
- Email
- Fecha de nacimiento
- Teléfono

DATOS DE ASIENTOS

- N° asiento
- Sector
- Fila

DATOS DE PARTIDOS

- Fecha
- Equipo local
- Equipo visitante
- Asientos disponibles

Tablas/Entidades:

A cada campo se le asignará su formato correspondiente, entero, clave primaria, Texto...etc

- Usuarios:
- ID_Usuario, N_Abonado, Es_Familiar_, Nombre, Apellidos, Email, Fecha, Fecha_Nacimiento, Teléfono, Usuario, Password
- Asientos:
- ID_Asiento, Sector, Fila, Número_Asiento
- Partidos:
- ID_Partido, Fecha, Equipo_Local, Equipo_Visitante, Asientos_Disponibles
- Reservas:
- ID_Reserva, ID_Usuario, ID_Asiento, ID_Partido, Fecha_Reserva, Estado
- Bloqueos:
- ID_Bloqueo, ID_Usuario, Fecha_Inicio, Fecha_Fin
- Correos enviados:
- ID_Correo, ID_Usuario, Asunto, Fecha_Envío, Contenido

Relaciones entre tablas:

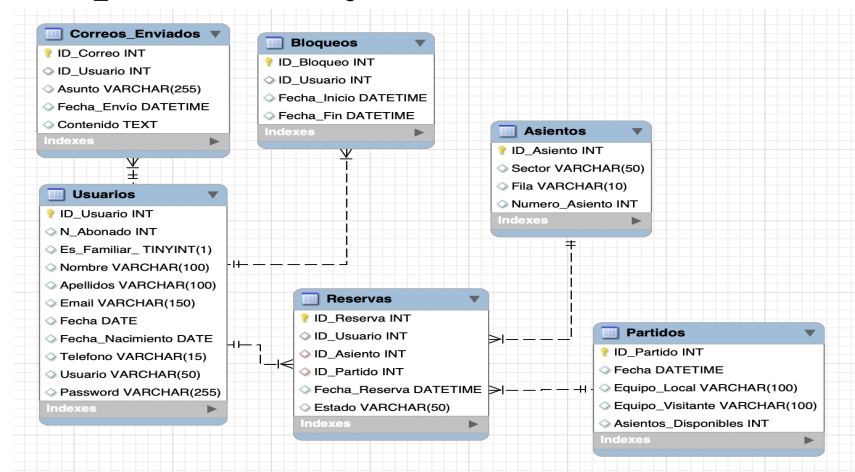
Usuarios-Reservas, un usuario puede hacer varias reservas.

Asientos-Reservas, un asiento puede estar asociado a varias reservas, pero no a la vez

Partidos-Reservas, un partido puede tener múltiples reservas

Usuarios-Bloqueos, un usuario puede tener historial de bloqueos

Correos_Enviados-Usuarios, registro de correos enviados a usuarios



Eventos clave del sistema:

Asignar y liberar asientos para los partidos

Verificar y gestionar bloqueos de usuarios

Notificaciones automáticas para correos electrónicos.

Controlar el acceso para no tener conflictos de reservas de asientos

b.

Considero MySql

- Por ser un proyecto que no va a manejar grandes volúmenes de datos, los datos que va a tener son relacionados con estructuras claras y ser compatible con FileZilla y Chrome, así con compatibilidad con software libre y ser de fácil manejo para el usuario administrador.
- Permite una escalabilidad en caso de ampliación de requisitos, tanto de usuarios como de mayores volúmenes de datos.
- Soporta conexiones múltiples, solucionando el caso de que dos usuarios quieran reservar el mismo asiento a la vez

- Tiene soporte para acceso remoto, pudiendo los administradores gestionar el sistema desde cualquier lugar.
- Es compatible con herramientas como MySQL WorkBench, que facilita el diseño y modelar la base de datos, permisos, relaciones y demás.
- Permite automatizar el envío de correos, solicitado en el proyecto.

C.

Para el Frontend:

Por ser esenciales en cualquier página web, HTML, CSS Y JavaScript con el framework React,

Para Backend:

Usaría PHP por su rendimiento, compatibilidad, manejo de bases de datos y desarrollo rápido, con el framework Laravel,

Para el envío de emails:

Al usar Laravel, tiene la posibilidad de usar Laravel Mail, proporcionando soporte para varios canales de entrega.

d.

Valoraría la implementación de una aplicación web, teniendo en cuenta que esta conlleva a un menor costo y tiempo de implementación y desarrollo, aparte, se podría usar desde cualquier dispositivo por ser por navegador, evitando el desarrollar para cada plataforma(IOS y Android), con los gastos que conlleva a publicar en dichas plataformas y el mantenimiento de estas.

2.

1.- Análisis:

- Identificar las necesidades del cliente.
- Estudio de mercado y competencia.
- Definir selección de asientos, procesos de pagos, gestión de eventos, etc.

2.- Diseño:

- Arquitectura de la aplicación.
- Interfaz de usuario.
- Base de datos.
- Seguridad y protección de datos

3.- Codificación:

- Preparación del entorno de desarrollo.
- Implementación de frontend.
- Desarrollo de backend.
- Configuración de base de datos.
- Integración de sistemas de pago y servicios de terceros.

4.- Pruebas:

- Unitarias y de integración.
- Usabilidad.
- Rendimiento y carga.
- Seguridad.
- Compatibilidad con diferentes dispositivos y navegadores.

5.- Verificación con el cliente:

- Desplegar la aplicación en los equipos del cliente.
- Usuario final utiliza el sistema en entorno de pruebas o de la propia organización
- Si no hay problemas, pasa a producción.

6.- Puesta en producción:

- Capacitar al personal.
- Lanzamiento oficial.
- Monitorización inicial para comprobar el rendimiento de la aplicación.

7.- Documentación:

- Documentar manual de uso.
- Documentación para posibles errores..
- Actualizar la documentación con cada actualización.

8.- Mantenimientos:

- Monitorear el rendimiento.
- Realizar actualizaciones y correcciones de errores.
- Implementar nuevas funcionalidades.

3.a.**Como pruebas unitarias:**

- Comprobar que no se puede reservar un asiento que ya está reservado.
- Que se apliquen descuentos en el caso de que los haya en el momento del pago.

Como pruebas de integración:

- Verificar que la aplicación se comunica correctamente con la base de datos,
- Comprobar que la pasarela de pago se completa correctamente.

b.

Como todo proyecto comercial, entre los más importantes podemos encontrar:

- **Plan de negocio**
- **Plan financiero**
- **Plan de costes**
- **Calendario de ejecución,**
- **Documentación técnica,**
- **Manual de usuario,**
- **Plan de pruebas,**

c.

Para la puesta en marcha de la aplicación, seguiremos los siguientes pasos.

- Verificar funcionalidades en un entorno de preproducción.
- Preparar el servidor de producción y la base de datos.
- Migrar el código y recursos al servidor.
- Implementar medidas de seguridad.
- Realizar pruebas finales en el entorno real.
- Formar al personal en el uso y gestión de la aplicación.
- Iniciar con un grupo limitado de usuarios.
- Supervisar el rendimiento y la estabilidad post-lanzamiento.

d.

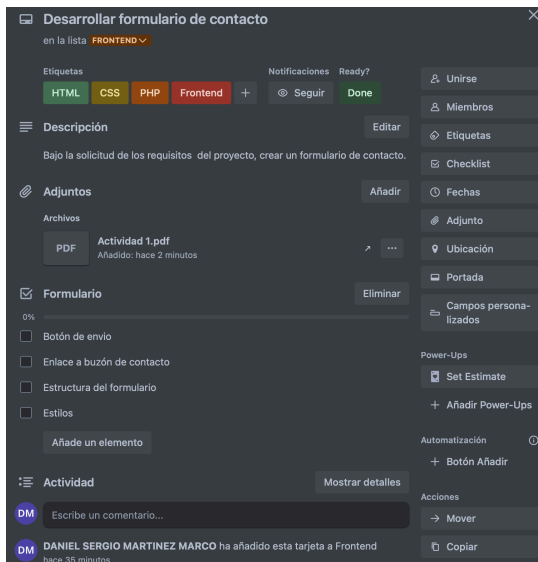
Como medidas de mantenimiento correctivo, priorizaremos según impacto, analizando las causas e intentando dar una respuesta rápida, documentando y posteriormente haciendo un seguimiento. **Como medida en mantenimiento evolutivo**, analizaremos el impacto de las nuevas funcionalidades, planificamos actualizaciones con sus pruebas, y una vez implementadas, formar de nuevo al personal y monitorear la implementación.

4.

Una posible solución sería usar los servicios de Ticketor, cumple con los requerimientos del proyecto, la forma de pago es sobre entrada vendida, facilitando la inversión inicial, teniendo el plan más económico 2.90% de las ventas totales + €0.69 por entrada, incluyendo más funcionalidades de las indicadas anteriormente, como generación de códigos QR, herramientas de marketing, tiendas online, comida, valores que aportan al proyecto inicial.

<https://es.ticketor.com>

5.



6.

Priorizar la asignación de asientos según antigüedad del abonado o asiduidad, premiando la fidelidad

Soporte multiplataforma con diseño responsivo para otros dispositivos.

Aplicación de recordatorios de próximos eventos

7.

Mediante lectores de qr, al sacar las entradas se enviaría en el email el código qr que se podría leer a la entrada del pabellón, pudiendo registrar si se sale, siendo válido solo durante el partido, o si se compra la entrada en ventanilla, se imprime el boleto con el código qr.

8.

Se podría implementar una recomendación personalizadas de ubicación con herramientas como Microsoft Azure Personalizer, basadas en un historial de usuario, o detección de fraudes en reservas, para usuarios que realicen compras de reservas con diferentes cuentas para tener la posibilidad de revender las entradas, una herramienta que podríamos usar es Anomaly Detection de AWS.

Anexo;

Por falta de espacio, no se añaden más capturas de pantalla y desarrolla más de lo deseado.

Para la localización de información concreta, como tecnologías más idóneas a usar, herramientas y documentación necesaria, he utilizado ayuda de chatgpt y perplexity, al desconocer de estas tecnologías y no encontrar mediante buscadores tradicionales información tan concreta fácilmente. Se resumen muchas preguntas por pasarme del límite impuesto de 4 páginas, se me quedaba en 11 páginas en un comienzo sin imágenes, desarrollando varias preguntas que considero que podrían aclarar lo que quería expresar, espero que el resumen de cada pregunta cumpla con la respuesta solicitada.

Para imagen de relación de tablas, creé las tablas en MySQL Workbench y generé el EER Diagram con ingeniería inversa.