**1. Estrategia de desarrollo y estimación del esfuerzo:**

Para el desarrollo del proyecto, se puede seguir un enfoque de metodología ágil, como Scrum, que permite una gestión iterativa e incremental del trabajo. Se pueden establecer sprints de desarrollo con duración de 2-4 semanas, en los que se aborden funcionalidades específicas del sistema. El equipo se organizará en roles como desarrolladores, diseñadores, tester, y un Scrum Master para facilitar el proceso.

La estimación del esfuerzo se realizará mediante técnicas de story points o puntos de función, donde se evaluará la complejidad de las tareas y se asignará un valor en puntos. Basado en la velocidad del equipo (cantidad de puntos completados por sprint), se puede predecir la duración aproximada del proyecto.

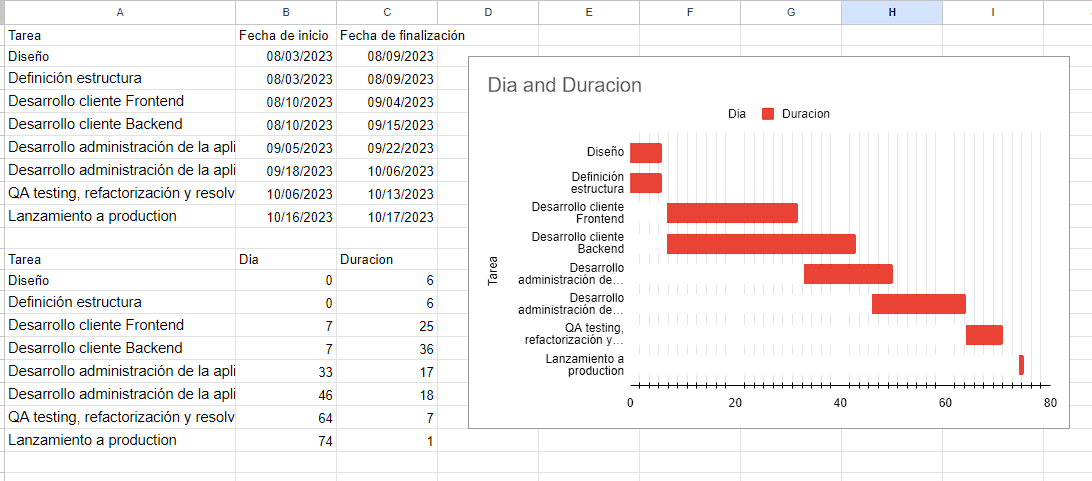
Este ejemplos con puntos de historia o esfuerzo para la creación de una aplicación web que está divida en Frontend y Backend

| **Tareas** | **Puntos historia** |
| --- | --- |
| **Diseño** |  |
| Diseño ux/ui marketplace | 15 |
| **Definición estructura** |  |
| Diseño de la estructura de la base de datos. | 8 |
| Definir estructura en la nube a utilizar, creación repositorios git. | 6 |
| **Desarollo cliente** |  |
| Creación de roles backend cliente, administrador y super administrador. | 4 |
| Desarrollo sesión de registro y login frontend | 4 |
| Desarrollo sesión registro y login backend | 4 |
| Desarrollo del frontend marketplace listado de eventos artísticos. | 6 |
| Desarrollo backend listado de eventos artísticos. | 2 |
| Creación single page del evento con comentarios. Frontend | 10 |
| Creación single page del evento con comentarios. backend | 8 |
| Creación de formulario para la compra de ticket sin registro y colocar un checkbox para elegir si desea ser registrado de una vez con los datos suministrados. frontend | 8 |
| Creación de formulario para la compra de ticket sin registro y colocar un checkbox para elegir si desea ser registrado de una vez con los datos suministrados. backend | 8 |
| Generación de recibo y ticket con una clave para consulta. Los recibo o ticket deben de llegar al email del cliente. backend | 10 |
| Persistencia del carrito basado en el backend se crea una id temporal y se crea una sesión del cliente con el ítem, y se crea un log de este cliente y el proceso. También si ya está registrado este mismo log podría guardar en que proceso quedó la compra. Todo proceso de compra tiene un id. Backend | 15 |
| Método de pagos basado en alguna plataforma de pago ya se stripe, place to pay, paypal, etc frontend | 10 |
| Método de pagos basado en alguna plataforma de pago ya se stripe, place to pay, paypal, etc Frontend | 10 |
| Método de pagos basado en alguna plataforma de pago ya se stripe, place to pay, paypal, etc backend | 12 |
| **Desarrollo administración de la aplicación** |  |
| Login administrador frontend | 6 |
| Creación de administradores solo el súper administrador puede registrar un nuevo administrador frontend | 6 |
| Creación de administradores solo el súper administrador puede registrar un nuevo administrador backend | 6 |
| CRUD de eventos con los siguientes campos, categoría, descripción, espacios disponibles, número de cupos, valor del tiquete, fecha de apertura, fecha de cierre de ventas. - Frontend | 10 |
| CRUD de eventos con los siguientes campos, categoría, descripción, espacios disponibles, número de cupos, valor del tiquete, fecha de apertura, fecha de cierre de ventas. Backend | 10 |
| CRUD de promociones relacionadas a la campaña. Frontend | 6 |
| CRUD de promociones relacionadas a la campaña. Backend | 6 |
| CRUD de descuentos relacionados a la campaña Frontend | 6 |
| CRUD de descuentos relacionados a la campaña Backend | 6 |
| CRUD productos relacionados Frontend | 6 |
| CRUD productos relacionados Backend | 6 |
| Creación dashboard de estadística de uso, visualización, comentarios de cada producto Frontend | 10 |
| Creación dashboard de estadística de uso, visualización, comentarios de cada producto Backend | 10 |
| Listado de eventos con una toggle para activar o desactivar si es necesario. frontend | 8 |
| Listado de eventos con una toggle para activar o desactivar si es necesario. Backend | 8 |
| **Testing y deploy a production** |  |
| QA testing, refactorización y resolver errores | 15 |
| Lanzamiento a production | 4 |

En este caso definire la velocidad de 30 de un sprint de 15 días:

1. El diseño ux/ui tomaría 15 puntos dificultad: 15/30 = 0.5 y 0,5 \* 2 = 1 semana
2. Definición estructura14 puntos dificultad 14/30 = 0.46 = y 0,46\*2 = 1 semana
3. Desarrollo cliente frontend 48 puntos dificultad 48/30 = 1.6 y 1.6\*2 = 3.2 semana
4. Desarrollo cliente backend 63 puntos dificultad 63/30 = 2.1 y 2.1\*2 = 4.2 semana
5. Desarrollo administración de la aplicación frontend48 puntos dificultad 48/30 = 1.6 y 1.6\*2 = 3.2 semana
6. Desarrollo administración de la aplicación Backend52 puntos dificultad 52/30 = 1.73 y 1.73\*2 = 3.46 semana
7. QA testing, refactorización y resolver errores 15 puntos dificultad 15/30 = 0.5 y 0.5\*2 = 1 semana
8. Lanzamiento a production 4 puntos dificultad 4/30 = 0.13 y 0.13\*2 = 2 dias

Con el diagrama de gantt se puede determinar que son 76 días basado en el esfuerzo de cada tarea.



**2. Propuesta técnica y diseño arquitectónico:**

* El sistema se puede diseñar con una arquitectura de tres capas: presentación, lógica de negocio y persistencia de datos.
* Capa de Presentación: Interfaz web y/o aplicación móvil para que los usuarios accedan al marketplace, compren boletos y configuren sus preferencias.
* Capa de Lógica de Negocio: Gestiona la lógica de las transacciones, estadísticas, gestión de eventos y preferencias. Aquí se incluirá el motor de recomendaciones basado en las preferencias de los usuarios.
* Capa de Persistencia de Datos: Base de datos para almacenar la información de eventos, usuarios, transacciones y preferencias.
* Se utilizarán tecnologías web modernas, como React o Angular para la interfaz de usuario, Node.js o Django para el backend, y bases de datos relacionales o NoSQL según las necesidades del sistema.

**Casos de uso:**

* Comprar boletos para eventos.
* Configurar preferencias y recibir recomendaciones.
* Publicar eventos y gestionar propuestas de publicación.
* Visualizar estadísticas y comentarios de eventos.

**Características sistémicas:**

* Seguridad y encriptación de datos.
* Alta disponibilidad para evitar caídas del sistema en momentos de alta demanda.
* Escalabilidad para manejar un gran número de usuarios y eventos.
* Eficiencia en el procesamiento de transacciones y generación de estadísticas.

**Escenarios de calidad:**

* Pruebas de carga para asegurar la capacidad del sistema bajo diferentes cargas de usuarios.
* Pruebas de seguridad para identificar vulnerabilidades y asegurar la protección de datos del usuario.
* Monitoreo constante para detectar y resolver cuellos de botella o problemas de rendimiento.

**3. Riesgos técnicos del proyecto:**

* Integración con sistemas de pago: Asegurar que la integración con los proveedores de pago sea segura y robusta para evitar problemas en las transacciones financieras.
* Seguridad y protección de datos: Garantizar que la información del usuario esté protegida y evitar vulnerabilidades que puedan poner en riesgo la privacidad.
* Escalabilidad: El sistema debe ser capaz de manejar el aumento de usuarios y eventos sin degradar su rendimiento.
* Experiencia del usuario: Asegurar que la interfaz sea intuitiva y atractiva para que los usuarios encuentren fácilmente los eventos y realicen compras sin problemas.

**4. Gestión del equipo técnico y control del proceso**:

Mediante reuniones cortas se asegura que todos los miembros trabajen eficientemente y se eliminen obstáculos para el desarrollo. Las reuniones diarias de Scrum permitirán el seguimiento del progreso y la identificación temprana de posibles problemas.

El control del proceso se realizará mediante revisión de entregables en cada sprint, así como la comparación de las estimaciones iniciales con el trabajo completado. Se realizarán retrospectivas para mejorar continuamente los procesos y abordar posibles deudas técnicas.

**5. Mecanismos y estrategias para la seguridad:**

* Se implementará un sistema de autenticación seguro para proteger las cuentas de usuario.
* Todas las comunicaciones entre el cliente y el servidor se realizarán a través de conexiones seguras mediante HTTPS.
* Se utilizarán prácticas de codificación segura y se realizarán pruebas de seguridad regulares.
* Se llevará a cabo una auditoría de seguridad para identificar posibles vulnerabilidades.

**6. Gestión de procesos y deuda técnica:**

La deuda técnica se manejará de manera proactiva durante el desarrollo. Se priorizarán las tareas para abordar la deuda técnica en cada sprint y se asignará tiempo para refactorizar y mejorar el código existente.

Se establecerán prácticas de desarrollo y revisión de código para asegurar que se sigan los estándares y se eviten problemas de deuda técnica en el futuro. También se realizarán pruebas automatizadas para garantizar la calidad del código y detectar problemas temprano.

Por último se dejará una semana solo para pruebas y corrección de algunos bugs antes de su lanzamiento a producción y así garantizar que el sistema esta lo mas estable posible.