

Fase 3: Elaboración del producto BackLog

3.1 Generación de epicas

- **1. Gestión de géneros musicales personalizados**
 - **Descripción:** Permitir que los usuarios seleccionen y gestionen sus géneros musicales favoritos desde su perfil. Esta información se usará para personalizar eventos sugeridos y recomendaciones de canciones.
 - **Objetivo de negocio:** Aumentar la personalización de la experiencia y el engagement del usuario desde el primer uso.
- **2. Sistema de recomendaciones de canciones por usuarios**
 - **Descripción:** Habilitar a los usuarios para sugerir canciones dentro del género permitido del evento. Las sugerencias quedarán visibles para otros asistentes y formarán parte de la playlist del evento.
 - **Objetivo de negocio:** Incentivar la participación activa y construir una experiencia colaborativa, generando mayor retención y sentido de comunidad.
- **3. Votación de canciones sugeridas por otros asistentes**
 - **Descripción:** Implementar un sistema de votación para priorizar canciones sugeridas. Las más votadas se destacarán en el evento.
 - **Objetivo de negocio:** Mejorar la interacción entre usuarios, y permitir que los eventos respondan más al gusto colectivo, elevando la satisfacción general.
- **4. Notificaciones y recordatorios de eventos**
 - **Descripción:** Enviar notificaciones automáticas antes del evento, y alertas en caso de cambios. Configurables por el usuario.
 - **Objetivo de negocio:** Reducir el ausentismo, aumentar la asistencia efectiva y mantener el contacto constante con el usuario.
- **5. Visualización del perfil del evento**
 - **Descripción:** Mostrar a los usuarios toda la información relevante de cada evento, incluyendo género, lugar y canciones propuestas.
 - **Objetivo de negocio:** Mejorar la tasa de conversión en inscripciones al brindar claridad y transparencia sobre lo que ofrece cada evento.
- **6. Creación de eventos por parte de organizadores**
 - **Descripción:** Brindar una interfaz para que organizadores creen eventos, definan género musical y gestionen propuestas y votaciones.
 - **Objetivo de negocio:** Escalar la app permitiendo a terceros generar contenido y promover sus propios eventos bajo el ecosistema de la app.

- **7. Integración con Spotify o YouTube para escuchar las recomendaciones**
 - **Descripción:** Permitir escuchar un preview o la canción completa desde la app usando APIs externas (Spotify, YouTube).
 - **Objetivo de negocio:** Aumentar la atraktividad y usabilidad de la app, facilitando la elección y aprobación de canciones sugeridas.
- **8. Sistema de badges o logros según participación**
 - **Descripción:** Crear un sistema de gamificación que premie a los usuarios que más participan (sugerencias, votos, asistencia).
 - **Objetivo de negocio:** Incentivar la fidelización de usuarios, el crecimiento de la comunidad y la generación de contenido.
- **9. Recomendación inteligente de eventos según historial**
 - **Descripción:** Mostrar sugerencias automáticas de nuevos eventos basadas en comportamiento anterior (géneros favoritos, asistencia, votos).
 - **Objetivo de negocio:** Mejorar el retorno recurrente de usuarios y ofrecer una experiencia cada vez más relevante y personalizada.
- **10. Definición de Arquitectura Modular y Escalable**
 - **Descripción:** Diseñar arquitectura basada en microservicios o separación de capas (backend, frontend, base de datos).
 - **Objetivo de negocio:** Permitir crecimiento sostenido y optimizar la performance futura.
- **11. Implementación de Logging y Monitoreo Centralizado**
 - **Descripción:** Integrar servicios de logs, alertas y métricas
 - **Objetivo de negocio:** Detectar fallas más rápido y optimizar el mantenimiento del sistema.
- **12. Gestión de géneros musicales para administradores**
 - **Descripción:** Crear un módulo de administración donde se pueda gestionar la tabla de referencia de géneros musicales. Los cambios impactan directamente en lo que se muestra a los usuarios durante el registro y en los eventos.
 - **Objetivo de negocio:** Mantener una base de datos ordenada, actualizada y flexible, permitiendo adaptar el sistema a nuevas tendencias musicales.
- **13. Gestión de tipos de evento como tabla de referencia**
 - **Descripción:** Crear una tabla de referencia editable desde un panel de administración que contenga los distintos tipos de evento. Esto permitirá que los organizadores los seleccionen al crear un evento, asegurando clasificación uniforme y filtros útiles.
 - **Objetivo de negocio:** Mejorar la organización y segmentación de eventos, facilitando búsquedas y recomendaciones personalizadas para los usuarios.
- **14. Gestión del estado del evento**
 - **Descripción:** Implementar una lógica basada en estados del evento, que determine si el evento es visible, editable, o cerrado. El estado debe almacenarse en una tabla de referencia que el sistema pueda interpretar.

- **Objetivo de negocio:** Mantener claridad en la interfaz para los usuarios, evitando confusión con eventos antiguos o cancelados, y ordenando la información.

3.2 Refinamiento de épicas en features

● 1. Gestión de géneros musicales personalizados:

- Descomponer cada épica en features más manejables:
 - **Selección de géneros al momento de registrarse:** Permitir que al crear la cuenta, el usuario pueda seleccionar uno o varios géneros musicales favoritos.
 - **Edición de géneros desde el perfil:** Permitir que el usuario pueda cambiar sus géneros favoritos en cualquier momento desde su perfil.
 - **Sistema de guardado de géneros asociados al usuario:** Base de datos para almacenar y actualizar los géneros seleccionados por cada usuario.
 - **Personalización de eventos y canciones basados en géneros favoritos:** Algoritmo que priorice mostrar eventos y canciones recomendadas basados en esos géneros.
- Documentar criterios de aceptación de alto nivel
 - Al registrarse, debe ser obligatorio seleccionar al menos un género.
 - Desde el perfil, el usuario debe poder agregar o quitar géneros cuando quiera.
 - El sistema debe guardar automáticamente los géneros seleccionados por el usuario.
 - Las recomendaciones de eventos y canciones deben basarse en los géneros seleccionados.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Selección de géneros al momento de registrarse** -> Must have
 - **Edición de géneros desde el perfil** -> must have
 - **Sistema de guardado de géneros asociados al usuario** -> must have
 - **Personalización de eventos y canciones basados en géneros favoritos** -> should have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Selección de géneros al momento de registrarse** -> Reach: 10 // Impact: 9 // confident: 8 // effort: 3 // total: $(10*9*8)/3 = 240$
 - **Edición de géneros desde el perfil** -> Reach: 8 // Impact: 7 // confident: 8 // effort: 5 // total: $(9*7*8)/5 = 100,8$
 - **Sistema de guardado de géneros asociados al usuario** -> Reach: 8 // Impact: 9 // confident: 8 // effort: 4 // total: $(8*9*8)/4 = 144$
 - **Personalización de eventos y canciones basados en géneros favoritos** -> Reach: 9 // Impact: 7 // confident: 9 // effort: 5 // total: $(9*7*9)/5 = 113,4$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature

- **Selección de géneros al momento de registrarse** -> S
 - **Edición de géneros desde el perfil** -> M
 - **Sistema de guardado de géneros asociados al usuario** -> M
 - **Personalización de eventos y canciones basados en géneros favoritos** -> L
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué sistema o librería de UI usar para la selección de géneros (checkbox, multiselect, etc.)?
 - **Spike 2:** ¿Qué modelo de base de datos es más eficiente para almacenar géneros múltiples por usuario?
 - Documentar supuestos y riesgos potenciales
 - En el word 1
- **2. Sistema de recomendaciones de canciones por usuarios**
- Descomponer cada épica en features más manejables:
 - **Habilitar sugerencia de canciones:** Permitir que usuarios inscriptos en un evento puedan proponer canciones.
 - **Visualización pública de sugerencias:** Mostrar todas las canciones sugeridas por los asistentes en la página del evento.
 - Documentar criterios de aceptación de alto nivel
 - Solo usuarios inscriptos pueden sugerir. Máximo de sugerencias por usuario configurable. Validar que las canciones sugeridas estén dentro del género definido para el evento.
 - Las sugerencias deben estar visibles para todos los asistentes. Incluir nombre de canción, artista y usuario que la propuso
 - Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Habilitar sugerencia de canciones** -> Must have
 - **Visualización pública de sugerencias** -> must have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Habilitar sugerencia de canciones** -> Reach: 7 // Impact: 9 // confident: 8 // effort: 6 // total: $(7*9*8)/6=84$
 - **Visualización pública de sugerencias** -> Reach: 9 // Impact: 9 // confident: 8 // effort: 4 // total: $(9*7*8)/4=162$
 - Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Habilitar sugerencia de canciones** -> L
 - **Visualización pública de sugerencias** -> M
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué sistema o librería de UI usar para la selección de géneros (checkbox, multiselect, etc.)?
 - **Spike 2:** ¿Qué modelo de base de datos es más eficiente para almacenar géneros múltiples por usuario?
 - Documentar supuestos y riesgos potenciales

- En el word 1

• 3. Votación de canciones sugeridas por otros asistentes

- Descomponer cada épica en features más manejables:
 - **Sistema de sugerencia de canciones:** Permitir que los usuarios puedan proponer canciones para el evento.
 - **Sistema de votación:** Permitir que otros usuarios voten positiva o negativamente sobre las canciones sugeridas.
 - **Visualización del ranking de canciones:** Mostrar una lista de canciones ordenada según la cantidad de votos recibidos.
 - **Notificaciones de nuevos eventos basados en preferencias:** Alertar automáticamente cuando se crea un nuevo evento que coincide con los géneros favoritos y ubicaciones frecuentes del usuario.
 - **Cierre de votaciones antes del evento:** Establecer un límite de tiempo para votar, previo al evento.
- Documentar criterios de aceptación de alto nivel
 - El usuario debe poder sugerir una canción dentro de un período habilitado para sugerencias.
 - Cada usuario puede votar (positiva o negativamente) una vez por canción.
 - Las canciones con más votos positivos se destacarán en la lista.
 - No se podrán votar nuevas canciones una vez cerrado el período de votación.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Sistema de sugerencia de canciones** -> Must have
 - **Sistema de votación** -> Must have
 - **Visualización del ranking de canciones** -> Should have
 - **Cierre de votaciones antes del evento** -> Must have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Sistema de sugerencia de canciones** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 4 // Total: $(9 \times 8 \times 8) / 4 = 144$
 - **Sistema de votación** -> Reach: 10 // Impact: 9 // Confidence: 9 // Effort: 5 // Total: $(10 \times 9 \times 9) / 5 = 162$
 - **Visualización del ranking de canciones** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 4 // Total: $(9 \times 8 \times 8) / 4 = 144$
 - **Cierre de votaciones antes del evento** -> Reach: 7 // Impact: 6 // Confidence: 7 // Effort: 5 // Total: $(7 \times 6 \times 7) / 5 = 58,8$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Sistema de sugerencia de canciones** -> M
 - **Sistema de votación** -> M
 - **Visualización del ranking de canciones** -> S
 - **Cierre de votaciones antes del evento** -> S

- Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** Integración con servicio de notificaciones.
 - **Spike 2:** Control de configuración de notificaciones.
- Documentar supuestos y riesgos potenciales
 - En el word 1

● 4. Notificaciones y recordatorios de eventos

- Descomponer cada épica en features más manejables:
 - **Configuración de notificaciones en perfil:** El usuario puede activar o desactivar notificaciones de eventos desde su perfil. Permite elegir qué tipo de notificaciones recibir: recordatorios, cambios, nuevos eventos recomendados.
 - **Notificaciones de recordatorio de evento:** 24-48 horas antes del evento, enviar una notificación recordando fecha, hora y ubicación.
 - **Notificaciones de cambios en eventos:** Notificar a los usuarios si hay cambios en eventos a los que están inscriptos (hora, sede, cancelación).
 - **Notificaciones de nuevos eventos basados en preferencias:** Alertar automáticamente cuando se crea un nuevo evento que coincide con los géneros favoritos y ubicaciones frecuentes del usuario.
- Documentar criterios de aceptación de alto nivel
 - El usuario puede configurar qué notificaciones quiere recibir (obligatorio).
 - El sistema envía recordatorios automáticos antes de cada evento.
 - Si un evento cambia o se cancela, los usuarios inscriptos reciben una alerta inmediata.
 - Nuevos eventos compatibles con los gustos del usuario son notificados automáticamente.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Configuración de notificaciones en perfil** -> Must have
 - **Notificaciones de recordatorio de evento** -> should have
 - **Notificaciones de cambios en eventos** -> Must have
 - **Notificaciones de nuevos eventos basados en preferencias** -> Could
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Configuración de notificaciones en perfil** -> Reach: 9 // Impact: 8 // confident: 6 // effort: 3 // total: $(9*8*6)/3=144$
 - **Notificaciones de recordatorio de evento** -> Reach: 7 // Impact: 9 // confident: 7 // effort: 3 // total: $(7*9*7)/3=147$

- **Notificaciones de cambios en eventos** -> Reach: 9 // Impact: 9 // confident: 8 // effort: 4 // total: $(9 \times 9 \times 8) / 4 = 162$
 - **Notificaciones de nuevos eventos basados en preferencias** -> Reach: 7 // Impact: 6 // confident: 8 // effort: 4 // total: $(7 \times 6 \times 8) / 4 = 84$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Configuración de notificaciones en perfil** -> L
 - **Notificaciones de recordatorio de evento** -> S
 - **Notificaciones de cambios en eventos** -> M
 - **Notificaciones de nuevos eventos basados en preferencias** -> M
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** Integración con servicio de notificaciones.
 - **Spike 2:** Control de configuración de notificaciones.
 - Documentar supuestos y riesgos potenciales
 - En el word 1

● 5. Visualización del perfil del evento

- Descomponer cada épica en features más manejables:
 - **Mostrar lugar del evento:** Visualizar dirección, nombre del lugar y, opcionalmente, un mapa.
 - **Mostrar fecha y hora del evento:** Indicar claramente el día y la hora de inicio.
 - **Mostrar géneros musicales del evento:** Listar los géneros musicales asociados.
 - **Mostrar canciones sugeridas y votadas:** Visualizar un listado de canciones sugeridas con sus votos.
 - **Mostrar lista de asistentes confirmados:** Mostrar cuántos usuarios asistirán (nombre, avatar u otra info breve).
- Documentar criterios de aceptación de alto nivel
 - El usuario debe ver claramente el lugar (texto + mapa opcional).
 - El usuario debe ver la fecha y hora del evento.
 - El usuario debe ver el género o géneros del evento.
 - El usuario debe ver las canciones sugeridas, ordenadas por votos.
 - El usuario debe ver la cantidad de asistentes confirmados (y opcionalmente sus nombres/avatars).
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Mostrar lugar del evento** -> Must have
 - **Mostrar fecha y hora del evento** -> Must have
 - **Mostrar géneros musicales del evento** -> Must have
 - **Mostrar canciones sugeridas y votadas** -> Should have
 - **Mostrar lista de asistentes confirmados** -> Could have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Mostrar lugar del evento** -> Reach: 10 // Impact: 9 // Confidence: 9 // Effort: 3 // Total: $(10 \times 9 \times 9) / 3 = 270$

- **Mostrar fecha y hora del evento** -> Reach: 10 // Impact: 8 // Confidence: 9 // Effort: 2 // Total: $(10 \times 8 \times 9) / 2 = 360$
 - **Mostrar géneros musicales del evento** -> Reach: 9 // Impact: 7 // Confidence: 8 // Effort: 3 // Total: $(9 \times 7 \times 8) / 3 = 168$
 - **Mostrar canciones sugeridas y votadas** -> Reach: 8 // Impact: 7 // Confidence: 8 // Effort: 4 // Total: $(8 \times 7 \times 8) / 4 = 112$
 - **Mostrar lista de asistentes confirmados** -> Reach: 7 // Impact: 6 // Confidence: 7 // Effort: 4 // Total: $(7 \times 6 \times 7) / 4 = 73,5$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Mostrar lugar del evento** -> S
 - **Mostrar fecha y hora del evento** -> S
 - **Mostrar géneros musicales del evento** -> S
 - **Mostrar canciones sugeridas y votadas** -> M
 - **Mostrar lista de asistentes confirmados** -> M
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué librería o servicio de mapas (por ejemplo, Mapbox, Google Maps) utilizar para integrar la ubicación del evento?
 - **Spike 2:** ¿Qué estrategia de carga usar para evitar tiempos lentos si hay muchos asistentes o muchas canciones sugeridas?
 - Documentar supuestos y riesgos potenciales
 - En el word 1

• 6. Creación de eventos por parte de organizadores

- Descomponer cada épica en features más manejables:
 - **Formulario de creación de eventos:** Pantalla para ingresar nombre, descripción, fecha, hora, lugar y género musical.
 - **Selector de género musical:** Permitir que el organizador elija uno o varios géneros para el evento
 - **Sistema de guardado de eventos:** Guardar en base de datos todos los datos del evento, vinculados al organizador.
 - **Gestión de propuestas de canciones:** Asociar automáticamente la funcionalidad de sugerir/votar canciones al nuevo evento creado.
 - **Validaciones básicas en el formulario:** No permitir guardar eventos incompletos (por ejemplo, fecha o género vacío).
- Documentar criterios de aceptación de alto nivel
 - El organizador debe poder acceder a un formulario para crear eventos.
 - El formulario debe exigir nombre, fecha, hora, lugar y género musical como campos obligatorios.
 - El sistema debe guardar correctamente el evento asociado al organizador.
 - Al crearse el evento, debe quedar habilitada la posibilidad de sugerir y votar canciones.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Formulario de creación de eventos** -> Must have

- **Selector de género musical** -> Must have
 - **Sistema de guardado de eventos** -> Must have
 - **Gestión de propuestas de canciones** -> Must have
 - **Validaciones básicas en el formulario** -> Must have
- Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Formulario de creación de eventos** -> Reach: 10 // Impact: 9 // Confidence: 8 // Effort: 5 // Total: $(10 \times 9 \times 8) / 5 = 144$
 - **Selector de género musical** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 3 // Total: $(9 \times 8 \times 8) / 3 = 192$
 - **Sistema de guardado de eventos** -> Reach: 10 // Impact: 9 // Confidence: 9 // Effort: 4 // Total: $(10 \times 9 \times 9) / 4 = 202,5$
 - **Gestión de propuestas de canciones** -> Reach: 8 // Impact: 7 // Confidence: 8 // Effort: 4 // Total: $(8 \times 7 \times 8) / 4 = 112$
 - **Validaciones básicas en el formulario** -> Reach: 9 // Impact: 8 // Confidence: 9 // Effort: 3 // Total: $(9 \times 8 \times 9) / 3 = 216$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Formulario de creación de eventos** -> M
 - **Selector de género musical** -> S
 - **Sistema de guardado de eventos** -> M
 - **Gestión de propuestas de canciones** -> L
 - **Validaciones básicas en el formulario** -> S
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Cómo manejar validaciones en tiempo real en el formulario? (Framework/librería de formularios)
 - **Spike 2:** ¿Qué arquitectura de base de datos es mejor para guardar eventos vinculados a múltiples organizadores?
 - Documentar supuestos y riesgos potenciales
 - En el word 1

• 7. Integración con Spotify o YouTube para escuchar las recomendaciones

- Descomponer cada épica en features más manejables:
 - **Botón de reproducción junto a cada sugerencia:** Añadir un botón para reproducir un fragmento o canción completa.
 - **Integración con API de Spotify:** Usar la API de Spotify para obtener previews (30 segundos) o canciones completas si es posible.
 - **Integración con API de YouTube:** Alternativa para reproducir canciones completas mediante videos de YouTube embebidos.
 - **Sistema de fallback:** Si no se encuentra en Spotify, buscar en YouTube automáticamente.
 - **Control de reproducción básico:** Permitir play/pause y control de volumen mínimo en la app.
- Documentar criterios de aceptación de alto nivel

- Al ver una sugerencia de canción, debe mostrarse un botón de reproducir.
- Si el usuario tiene cuenta de Spotify, debe poder escuchar un fragmento o la canción completa (según permisos).
- Si Spotify no está disponible, debe intentar reproducirse desde YouTube.
- La reproducción debe ser fluida y no trabar la experiencia de navegación en la app.
- Los usuarios deben poder pausar o reanudar la reproducción fácilmente.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Botón de reproducción junto a cada sugerencia** -> Could have
 - **Integración con API de Spotify** -> Must have
 - **Integración con API de YouTube** -> Should have
 - **Sistema de fallback** -> Should have
 - **Control de reproducción básico** -> Must have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Botón de reproducción junto a cada sugerencia** -> Reach: 10 // Impact: 6 // Confidence: 8 // Effort: 3 // Total: $(10 \times 6 \times 8) / 3 = 160$
 - **Integración con API de Spotify** -> Reach: 8 // Impact: 8 // Confidence: 8 // Effort: 5 // Total: $(8 \times 8 \times 8) / 5 = 102,4$
 - **Integración con API de YouTube** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 4 // Total: $(9 \times 8 \times 8) / 4 = 144$
 - **Sistema de fallback** -> Reach: 7 // Impact: 7 // Confidence: 7 // Effort: 5 // Total: $(7 \times 7 \times 7) / 5 = 68,6$
 - **Control de reproducción básico** -> Reach: 10 // Impact: 9 // Confidence: 9 // Effort: 3 // Total: $(10 \times 9 \times 9) / 3 = 270$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Botón de reproducción junto a cada sugerencia** -> S
 - **Integración con API de Spotify** -> M
 - **Integración con API de YouTube** -> M
 - **Sistema de fallback** -> L
 - **Control de reproducción básico** -> S
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué limitaciones tiene la API de Spotify para previews gratuitos? ¿Requiere autenticación de usuario?
 - **Spike 2:** ¿Cómo integrar un reproductor ligero de YouTube en la app sin afectar performance?
 - **Spike 3:** ¿Cómo manejar la reproducción simultánea si hay varias canciones en lista?
 - Documentar supuestos y riesgos potenciales
 - En el word 1

● 8. Sistema de badges o logros según participación

- Descomponer cada épica en features más manejables:

- **Definición de categorías de logros:** Definir tipos de badges: asistencia, sugerencias, votos, engagement general. Ej.: "Asistente Experto", "Rey de las Sugerencias", "Votante Activo".
- **Sistema automático de asignación de badges:** Al alcanzar ciertos hitos (ej: asistir a 5 eventos, sugerir 10 canciones), el sistema asigna automáticamente los badges al perfil del usuario.
- **Visualización de badges en el perfil:** Mostrar los logros del usuario en su perfil público como parte de su identidad.
- Documentar criterios de aceptación de alto nivel
 - Los badges estarán diferenciados por categorías.
 - El usuario recibe badges automáticos cuando cumple criterios de participación..
 - Los badges son visibles en el perfil público del usuario.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Definición de categorías de logros** -> should have
 - **Sistema automático de asignación de badges** -> must have
 - **Visualización de badges en el perfil** -> should have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Definición de categorías de logros** -> Reach: 8 // Impact: 7 // confident: 9 // effort: 4 // total: $(8*7*9)/4=126$
 - **Sistema automático de asignación de badges** -> Reach: 8 // Impact: 7 // confident: 8 // effort: 8 // total: $(8*7*8)/8=56$
 - **Visualización de badges en el perfil** -> Reach: 8 // Impact: 6 // confident: 8 // effort: 6 // total: $(8*6*8)/6=64$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Definición de categorías de logros**-> S
 - **Sistema automático de asignación de badges** -> XL
 - **Visualización de badges en el perfil** -> M
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** Diseño visual de badges: Investigar y definir el estilo visual de los badges (minimalistas, coloridos, realistas, gamificados).
 - **Spike 2:** Definición de hitos y métricas: Validar métricas justas y alcanzables para la entrega de badges.
 - Documentar supuestos y riesgos potenciales
 - En el word 1

● 9. Recomendación inteligente de eventos según historial

- Descomponer cada épica en features más manejables:
 - **Tracking de asistencia a eventos:** Registrar automáticamente a qué eventos asistió cada usuario.
 - **Registro de votos y géneros favoritos:** Guardar qué géneros y canciones vota o elige más.

- **Motor de recomendación básico:** Crear un algoritmo inicial que use géneros favoritos + historial de eventos para sugerir nuevos eventos.
 - **Sistema de recomendaciones en el inicio:** Mostrar eventos recomendados al ingresar a la app o en una sección destacada.
 - **Actualización dinámica de preferencias:** Ajustar sugerencias en tiempo real a medida que el usuario participa más.
- Documentar criterios de aceptación de alto nivel
 - El sistema debe registrar los eventos a los que asiste un usuario.
 - El sistema debe considerar tanto los géneros favoritos como la asistencia/votos pasados.
 - Se deben mostrar sugerencias personalizadas de eventos al ingresar a la app o en un área destacada.
 - Las recomendaciones deben actualizarse automáticamente cuando cambie la actividad del usuario.
Si el usuario es nuevo (sin historial), basarse solo en géneros favoritos.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Tracking de asistencia a eventos** -> Must have
 - **Registro de votos y géneros favoritos** -> Must have
 - **Motor de recomendación básico** -> Must have
 - **Sistema de recomendaciones en el inicio** -> Should have
 - **Actualización dinámica de preferencias** -> Could have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Tracking de asistencia a eventos** -> Reach: 9 // Impact: 8 // Confidence: 9 // Effort: 3 // Total: $(9 \times 8 \times 9) / 3 = 216$
 - **Registro de votos y géneros favoritos** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 3 // Total: $(9 \times 8 \times 8) / 3 = 192$
 - **Motor de recomendación básico** -> Reach: 10 // Impact: 10 // Confidence: 8 // Effort: 5 // Total: $(10 \times 10 \times 8) / 5 = 160$
 - **Sistema de recomendaciones en el inicio** -> Reach: 9 // Impact: 7 // Confidence: 8 // Effort: 5 // Total: $(9 \times 7 \times 8) / 5 = 100,8$
 - **Actualización dinámica de preferencias** -> Reach: 8 // Impact: 7 // Confidence: 7 // Effort: 5 // Total: $(8 \times 7 \times 7) / 5 = 78,4$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Tracking de asistencia a eventos** -> S
 - **Registro de votos y géneros favoritos** -> S
 - **Motor de recomendación básico** -> M
 - **Sistema de recomendaciones en el inicio** -> M
 - **Actualización dinámica de preferencias** -> L
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué algoritmo simple de recomendación usar? (por ejemplo, filtrado basado en contenido vs filtrado colaborativo)
 - **Spike 2:** ¿Cómo priorizar eventos nuevos y no repetir siempre los mismos?

- **Spike 3:** ¿Cómo manejar la falta de historial en usuarios nuevos?
- Documentar supuestos y riesgos potenciales
 - En el word 1

● 10. Definición de Arquitectura Modular y Escalable

- Descomponer cada épica en features más manejables:
 - **Definición de estructura modular inicial:** Establecer la separación clara entre frontend, backend, y base de datos, con comunicación bien definida.
 - **Diseño de integración entre módulos:** Definir mecanismos de comunicación entre servicios (ej: API Gateway, colas de mensajería, Webhooks).
 - **Documentación de la arquitectura:** Crear documentación técnica detallada: diagramas de arquitectura, flujos de datos, contratos de APIs.
 - **Estrategia de escalabilidad horizontal:** Planificar cómo escalar los servicios críticos en función de la carga.
- Documentar criterios de aceptación de alto nivel
 - Se define y documenta una arquitectura inicial modular y escalable.
 - Se identifica la separación de responsabilidades de cada módulo o servicio.
 - Se establecen las bases para el crecimiento: escalabilidad horizontal y modularidad.
 - Se definen lineamientos de buenas prácticas para mantener la arquitectura a largo plazo.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Definición de estructura modular inicial** -> must have
 - **Diseño de integración entre módulos** -> must have
 - **Documentación de la arquitectura** -> should have
 - **Estrategia de escalabilidad horizontal** -> should have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Definición de estructura modular inicial** -> Reach: 9 // Impact: 10 // confident: 9 // effort: 5 // total: $(9*10*9)/5=162$
 - **Diseño de integración entre módulos** -> Reach: 8 // Impact: 9 // confident: 8 // effort: 4 // total: $(8*9*8)/4=144$
 - **Documentación de la arquitectura** -> Reach: 6 // Impact: 8 // confident: 7 // effort: 3 // total: $(8*6*7)/3=112$
 - **Estrategia de escalabilidad horizontal** -> Reach: 7 // Impact: 8 // confident: 7 // effort: 5 // total: $(8*6*8)/6=78$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Definición de estructura modular inicial**-> M
 - **Diseño de integración entre módulos** -> L
 - **Documentación de la arquitectura** -> S
 - **Estrategia de escalabilidad horizontal** -> M
 - Identificar features que requieren investigación técnica (spikes)

- **Spike 1:** Análisis de tecnologías a utilizar: Evaluar si conviene usar microservicios desde el principio o diseñar una arquitectura modular monolítica inicial con opción de desacople futuro.
 - **Spike 2:** Selección de herramientas de comunicación entre servicios: Investigar herramientas más adecuadas para orquestar microservicios.
 - **Spike 3:** Costos y complejidad de escalabilidad: Analizar el impacto económico y técnico de escalar la plataforma en infraestructura en la nube.
- Documentar supuestos y riesgos potenciales
 - En el word 1

● 11. Implementación de Logging y Monitoreo Centralizado

- Descomponer cada épica en features más manejables:
 - **Configuración de sistema de logging unificado:** Establecer logs de errores, advertencias e información general en todas las partes críticas de la app.
 - **Integración de plataforma de monitoreo:** Usar herramientas como Datadog, New Relic, Grafana, Prometheus, etc.
 - **Alertas automáticas:** Definir reglas para que se generen alertas en caso de errores críticos o caídas de servicios.
 - **Dashboards de métricas:** Crear paneles que muestren métricas de salud del sistema (tiempos de respuesta, errores, uptime, etc.).
 - **Documentación de políticas de logging y monitoreo:** Dejar pautas claras para el equipo sobre qué, cómo y cuándo loguear o configurar alertas.
- Documentar criterios de aceptación de alto nivel
 - Todos los logs deben centralizarse en un único sistema.
 - El sistema debe enviar alertas automáticas ante errores críticos o caídas.
 - Debe haber dashboards accesibles para monitorear métricas clave.
 - La configuración debe estar documentada y mantenida.
 - Los niveles de log (INFO, WARNING, ERROR) deben estar correctamente diferenciados.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Configuración de sistema de logging unificado** -> Must have
 - **Integración de plataforma de monitoreo** -> Must have
 - **Alertas automáticas** -> Must have
 - **Dashboards de métricas** -> Should have
 - **Documentación de políticas de logging y monitoreo** -> Should have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Configuración de sistema de logging unificado** -> Reach: 10 // Impact: 9 // Confidence: 9 // Effort: 4 // Total: $(10 \times 9 \times 9) / 4 = 202,5$
 - **Integración de plataforma de monitoreo** -> Reach: 9 // Impact: 9 // Confidence: 8 // Effort: 4 // Total: $(9 \times 9 \times 8) / 4 = 162$
 - **Alertas automáticas** -> Reach: 9 // Impact: 10 // Confidence: 8 // Effort: 3 // Total: $(9 \times 10 \times 8) / 3 = 240$

- **Dashboards de métricas** -> Reach: 8 // Impact: 7 // Confidence: 8 // Effort: 5 // Total: $(8 \times 7 \times 8) / 5 = 89,6$
 - **Documentación de políticas de logging y monitoreo** -> Reach: 7 // Impact: 6 // Confidence: 8 // Effort: 4 // Total: $(7 \times 6 \times 8) / 4 = 84$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Estimar el tamaño relativo de cada feature
 - **Configuración de sistema de logging unificado** -> M
 - **Integración de plataforma de monitoreo** -> M
 - **Alertas automáticas** -> S
 - **Dashboards de métricas** -> M
 - **Documentación de políticas de logging y monitoreo** -> S
 - Identificar features que requieren investigación técnica (spikes)
 - **Spike 1:** ¿Qué stack de monitoreo y logging se adapta mejor al presupuesto y a la arquitectura? (Ej: Datadog, New Relic, ELK Stack, Grafana + Loki, etc.)
 - **Spike 2:** ¿Cómo manejar la retención de logs y evitar costos excesivos?
 - **Spike 3:** ¿Qué métricas mínimas deben ser monitoreadas para considerar el sistema saludable?
 - Documentar supuestos y riesgos potenciales
 - En el word 1

• 12. Gestión de géneros musicales para administradores

- Descomponer cada épica en features más manejables:
 - **Listado de géneros musicales:** Permitir al administrador visualizar todos los géneros musicales disponibles en el sistema.
 - **Creación de nuevos géneros:** Permitir al administrador agregar nuevos géneros musicales con nombre y descripción.
 - **Edición de géneros existentes:** Permitir modificar el nombre o descripción de un género musical ya registrado.
 - **Eliminación de géneros musicales:** Permitir eliminar géneros que no estén en uso por usuarios o eventos.
 - **Validación de uso antes de eliminar:** Verificar si un género está siendo usado antes de permitir su eliminación.
 - **Actualización en tiempo real de formularios de usuario:** Asegurar que los cambios en la lista de géneros se reflejen automáticamente en formularios de registro y eventos.
- Documentar criterios de aceptación de alto nivel:
 - El administrador debe poder crear, ver, editar y eliminar géneros musicales.
 - No se debe permitir eliminar géneros que estén en uso por usuarios o eventos activos.
 - Los cambios realizados en los géneros deben reflejarse inmediatamente en los formularios disponibles para los usuarios.
 - No se deben permitir nombres duplicados o campos vacíos en los géneros.
- Priorizar según importancia para el negocio

- Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Listado de géneros musicales** -> Must have
 - **Creación de nuevos géneros** -> Must have
 - **Edición de géneros existentes** -> Must have
 - **Eliminación de géneros musicales** -> Should have
 - **Validación de uso antes de eliminar** -> Must have
 - **Actualización en tiempo real de formularios de usuario** -> Must have
- Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Listado de géneros musicales** -> Reach: 8 // Impact: 7 // Confidence: 9 // Effort: 3 // Total: $(8 \times 7 \times 9) / 3 = 168$
 - **Creación de nuevos géneros** -> Reach: 7 // Impact: 8 // Confidence: 8 // Effort: 3 // Total: $(7 \times 8 \times 8) / 3 = 149.3$
 - **Edición de géneros existentes** -> Reach: 6 // Impact: 6 // Confidence: 7 // Effort: 3 // Total: $(6 \times 6 \times 7) / 3 = 84$
 - **Eliminación de géneros musicales** -> Reach: 4 // Impact: 5 // Confidence: 7 // Effort: 3 // Total: $(4 \times 5 \times 7) / 3 = 46.6$
 - **Validación de uso antes de eliminar** -> Reach: 5 // Impact: 9 // Confidence: 7 // Effort: 4 // Total: $(5 \times 9 \times 7) / 4 = 78.75$
 - **Actualización en tiempo real de formularios de usuario** -> Reach: 9 // Impact: 9 // Confidence: 7 // Effort: 5 // Total: $(9 \times 9 \times 7) / 5 = 113.4$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Listado de géneros musicales → S
 - Creación de nuevos géneros → M
 - Edición de géneros existentes → S
 - Eliminación de géneros musicales → S
 - Validación de uso antes de eliminar → M
 - Actualización en tiempo real de formularios de usuario → M
- Identificar features que requieren investigación técnica (spikes):
 - **Spike 1:** Validar impacto de sincronización en formularios y caché del frontend.
 - **Spike 2:** Evaluar si es necesario mantener un histórico de cambios en géneros musicales.
- Documentar supuestos y riesgos potenciales
 - En el word 1
- **13. Gestión de tipos de evento como tabla de referencia**
 - Descomponer cada épica en features más manejables:
 - **Listado de tipos de evento:** Permitir al administrador visualizar todos los tipos de evento actualmente disponibles.
 - **Creación de nuevos tipos de evento:** Permitir al administrador agregar nuevos tipos de evento con nombre y descripción.
 - **Edición de tipos de evento existentes:** Permitir modificar nombre o descripción de un tipo de evento registrado.
 - **Eliminación de tipos de evento:** Permitir eliminar tipos de evento que no estén asociados a eventos activos o futuros.
 - **Validación de uso antes de eliminar:** Verificar que el tipo de evento no esté en uso antes de permitir su eliminación.

- **Actualización automática en formulario de creación de eventos:** Reflejar cambios en tiempo real en el formulario usado por organizadores.
- Documentar criterios de aceptación de alto nivel:
 - El administrador debe poder crear, ver, editar y eliminar tipos de evento desde un panel de control.
 - No se debe permitir eliminar un tipo de evento si está vinculado a eventos futuros o activos.
 - Los cambios deben reflejarse de forma inmediata en el formulario de creación de eventos usado por organizadores.
 - No se deben permitir nombres duplicados ni campos vacíos.
- Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Listado de tipos de evento** -> Must have
 - **Creación de nuevos tipos de evento** -> Must have
 - **Edición de tipos de evento existentes** -> Must have
 - **Eliminación de tipos de evento** -> Should have
 - **Validación de uso antes de eliminar** -> Must have
 - **Actualización automática en formulario de eventos** -> Must have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Listado de tipos de evento** -> Reach: 8 // Impact: 7 // Confidence: 9 // Effort: 3
-> Total: $(8 \times 7 \times 9) / 3 = 168$
 - **Creación de nuevos tipos de evento** -> Reach: 7 // Impact: 8 // Confidence: 8 // Effort: 3 -> Total: $(7 \times 8 \times 8) / 3 = 149.3$
 - **Edición de tipos de evento existentes** -> Reach: 6 // Impact: 6 // Confidence: 7 // Effort: 3 -> Total: $(6 \times 6 \times 7) / 3 = 84$
 - **Eliminación de tipos de evento** -> Reach: 4 // Impact: 5 // Confidence: 7 // Effort: 3 -> Total: $(4 \times 5 \times 7) / 3 = 46.6$
 - **Validación de uso antes de eliminar** -> Reach: 5 // Impact: 9 // Confidence: 7 // Effort: 4 -> Total: $(5 \times 9 \times 7) / 4 = 78.75$
 - **Actualización automática en formulario de eventos** -> Reach: 9 // Impact: 9 // Confidence: 7 // Effort: 5 -> Total: $(9 \times 9 \times 7) / 5 = 113.4$
 - Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Listado de tipos de evento -> S
 - Creación de nuevos tipos de evento -> M
 - Edición de tipos de evento existentes -> S
 - Eliminación de tipos de evento -> S
 - Validación de uso antes de eliminar -> M
 - Actualización automática en formulario de eventos -> M
- Identificar features que requieren investigación técnica (spikes):
 - Spike 1: Evaluar impacto en formularios de creación de eventos y si requieren recarga/caché.
 - Spike 2: Analizar qué hacer con eventos históricos si se elimina un tipo de evento.

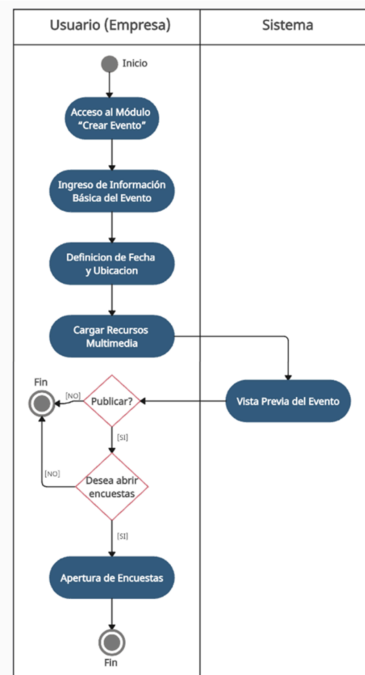
- Documentar supuestos y riesgos potenciales
 - En el word 1
- **14. Gestión del estado del evento**
 - Descomponer cada épica en features más manejables:
 - **Tabla de referencia de estados de evento:** Crear una tabla que contenga todos los posibles estados del evento (ej: Borrador, Activo, Finalizado, Cancelado).
 - **Asignación automática de estado según fechas o acciones:** Configurar lógica que actualice el estado del evento en base a reglas (ej: cambia a "Finalizado" si ya ocurrió).
 - **Control de visibilidad según estado:** Ocultar eventos cancelados o finalizados del listado público si corresponde.
 - **Control de edición según estado:** Restringir la edición de eventos que estén en estado finalizado o cancelado.
 - **Edición manual del estado por administradores:** Permitir a un administrador cambiar manualmente el estado de un evento desde un panel.
 - **Visualización clara del estado en las interfaces de usuario:** Mostrar el estado del evento de forma destacada tanto para organizadores como para asistentes.
 - Documentar criterios de aceptación de alto nivel:
 - Cada evento debe tener un estado asignado en base a una tabla de referencia.
 - El estado del evento debe impactar su visibilidad y posibilidad de edición.
 - El administrador debe poder cambiar manualmente el estado de un evento si es necesario.
 - El estado debe mostrarse de forma clara en todas las interfaces relevantes (listados, detalles, formularios).
 - La lógica automática debe actualizar estados en base a condiciones como fecha de finalización.
 - Priorizar según importancia para el negocio
 - Método MoSCoW: Must have (Debe), Should (Debería), Could (Podría) y Won't (No debe)
 - **Tabla de referencia de estados de evento** -> Must have
 - **Asignación automática de estado según fechas o acciones** -> Must have
 - **Control de visibilidad según estado** -> Must have
 - **Control de edición según estado** -> Must have
 - **Edición manual del estado por administradores** -> Should have
 - **Visualización clara del estado en las interfaces de usuario** -> Must have
 - Método RICE: Reach (Alcance), Impact (Impacto), Confidence (Confianza), Effort (Esfuerzo)
 - **Tabla de referencia de estados de evento** -> Reach: 9 // Impact: 8 // Confidence: 9 // Effort: 3 -> Total: $(9 \times 8 \times 9) / 3 = 216$
 - **Asignación automática de estado** -> Reach: 8 // Impact: 9 // Confidence: 8 // Effort: 4 -> Total: $(8 \times 9 \times 8) / 4 = 144$
 - **Control de visibilidad según estado** -> Reach: 9 // Impact: 8 // Confidence: 8 // Effort: 4 -> Total: $(9 \times 8 \times 8) / 4 = 144$
 - **Control de edición según estado** -> Reach: 8 // Impact: 7 // Confidence:

- 7 // Effort: 4 -> Total: $(8 \times 7 \times 7) / 4 = 98$
 - Edición manual del estado por administradores -> Reach: 5 // Impact: 6 // Confidence: 7 // Effort: 3 -> Total: $(5 \times 6 \times 7) / 3 = 70$
 - Visualización clara del estado en UI -> Reach: 10 // Impact: 7 // Confidence: 9 // Effort: 3 -> Total: $(10 \times 7 \times 9) / 3 = 210$
- Estimación inicial: T-shirt sizing (XS, S, M, L, XL) o Planning Poker (1, 2, 3, 5, 8, 13, 21 y 34)
 - Tabla de referencia de estados de evento -> S
 - Asignación automática de estado según fechas o acciones -> M
 - Control de visibilidad según estado -> S
 - Control de edición según estado -> S
 - Edición manual del estado por administradores -> S
 - Visualización clara del estado en UI -> S
- Identificar features que requieren investigación técnica (spikes):
 - **Spike 1:** Evaluar el mecanismo óptimo para cambiar automáticamente el estado (cron jobs, triggers, reglas de negocio).
 - **Spike 2:** Determinar cómo mantener sincronizado el estado entre backend y frontend sin errores.
- Documentar supuestos y riesgos potenciales
 - En el word 1

3.3 Diagrama Actividades y Estados

Proceso: Publicación de un Evento

Objetivo: Permitir que una empresa/organizador publique un evento para que los usuarios/clientes puedan verlo e interactuar con el

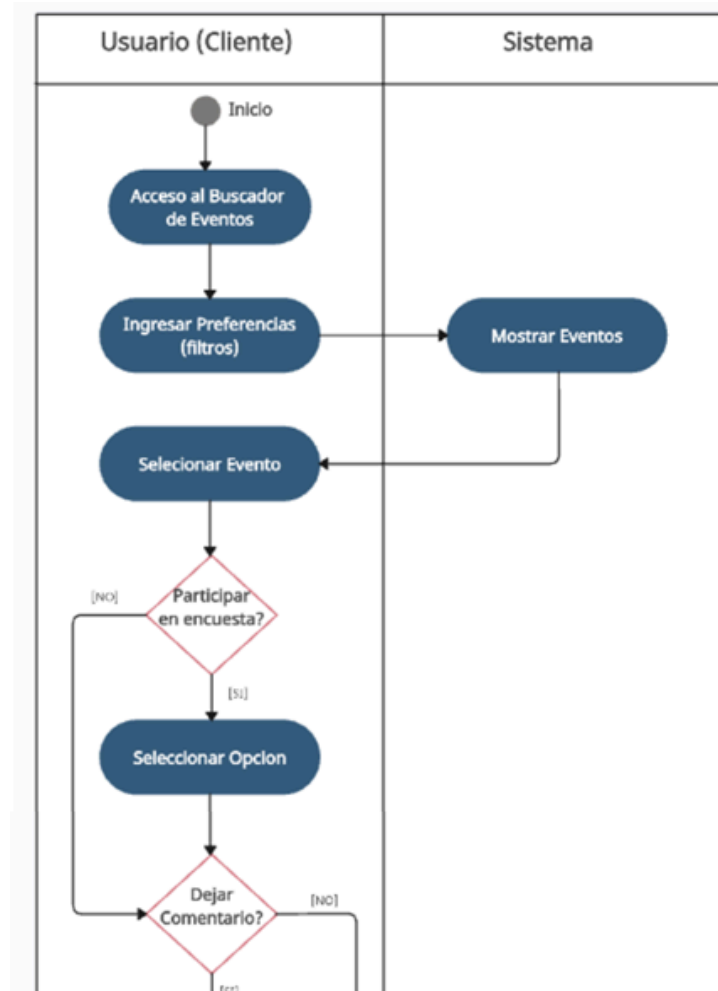


Etapas Detalladas (Secuencia de Actividades):

1. **Acceso al Módulo “Crear Evento”:** El usuario (empresa) accede a la opción para iniciar la creación de un nuevo evento desde el panel correspondiente.
2. **Ingreso de Información Básica del Evento:** el usuario deberá ingresar el título del evento, una descripción detallada (información que se considere importante).
3. **Definición de Fecha y Ubicación:** El usuario establece la fecha, hora de inicio/fin y la ubicación (física) del evento
4. **Cargar Recursos Multimedia:** Se agregan archivos como imágenes promocionales, videos o documentos relevantes al evento.
5. **Vista Previa:** El sistema genera una previsualización con toda la información ingresada para revisión antes de publicar.
6. **Publicación del Evento:** El usuario decide si desea publicar el evento. En caso afirmativo, el evento se vuelve visible para los clientes.
7. **Apertura de Encuestas (Opcional):** Si el organizador lo desea, puede abrir encuestas para que los usuarios participen en la elección de aspectos del evento como temática, música o preferencias generales.

Proceso: Interacción del Usuario con un Event

Objetivo: Permitir que un cliente explore, seleccione e interactúe con un evento publicado según sus intereses, brindando una experiencia personalizada previa al evento

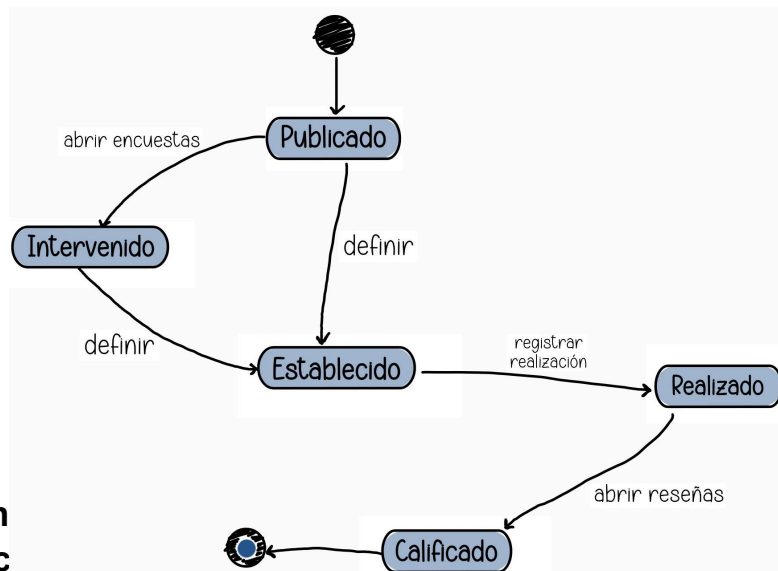


Etapas Detalladas (Secuencia de Actividades):

1. **Acceso al Buscador de Eventos:**
El usuario accede al módulo o sección del sistema destinado a la exploración de eventos disponibles.
2. **Ingreso de Preferencias (Filtros):** el usuario introduce criterios de búsqueda como tipo de evento, ubicación, fecha, temática, etc., para obtener resultados más relevantes.
3. **Mostrar Eventos:** el sistema procesa los filtros y muestra una lista de eventos que coinciden con los criterios seleccionados.
4. **Seleccionar Evento:** el usuario elige un evento específico de la lista para conocer más detalles y opciones de interacción.
5. **Participar en Encuesta:** si el evento tiene encuestas activas, el sistema ofrece al usuario participar. El usuario puede optar por no responder o seleccionar una opción disponible.
6. **Dejar Comentario:** el sistema consulta al usuario si desea dejar una opinión o sugerencia. En caso afirmativo, se habilita un campo para ingresar el comentario.

7. **Añadir a Favoritos:** el usuario puede marcar el evento como favorito para guardarlo y acceder fácilmente más adelante.
8. **Activar Notificaciones:** el sistema ofrece la opción de activar notificaciones para recibir alertas futuras sobre cambios, recordatorios o nuevas interacciones relacionadas con el evento.

Diagrama de Estado para “Evento”



Descripción

1. Public

El usuario (organizador) crea y publica el evento para que el resto de usuarios puedan visualizarlo.

2. **Intervenido:** un evento está en “Intervenido” cuando el usuario habilita las encuestas para el evento que creó. Permitiendo así que los usuarios puedan interactuar de distintas maneras.

3. **Establecido:** un evento está en “Establecido” cuando el usuario define cuando ya se terminó la interacción con el cliente (pre evento).

4. **Realizado:** el evento se encuentra en “Realizado” cuando ya transcurrió su fecha de realización.

5. **Calificado:** el evento se encuentra en “Calificado” cuando los distintos usuarios que asistieron al evento dejan una reseña/crítica sobre su experiencia.

3.4 Historias de Usuario por Feature

[HU-001] - Gestión de géneros musicales personalizados

Descripción:

Como usuario,
quiero seleccionar mis géneros musicales favoritos al registrarme
para recibir sugerencias de eventos y canciones alineadas a mis gustos.

Prioridad: MoSCoW (Must have)

Estimación: M (Medium)

Criterios de Aceptación:

1. Dado que soy un usuario, cuando registro estoy registrando mi perfil, entonces debo poder seleccionar mis géneros musicales favoritos.
2. Dado que he seleccionado mis géneros musicales, cuando el sistema me muestra eventos recomendados, entonces estos deben estar alineados con los géneros que he elegido.

Notas Técnicas:

- Integrar con la base de datos de géneros musicales disponibles.
- Asegurar que los géneros seleccionados por el usuario se guarden en su perfil.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-001] El sistema podría no ofrecer recomendaciones precisas si el proceso de selección de géneros presenta dificultades técnicas en base a su desarrollo.

[HU-002] - Sistema de recomendaciones de canciones por usuarios**Descripción:**

Como usuario,
quiero recomendar canciones para los eventos,
para contribuir a la playlist colaborativa del evento.

Prioridad: MoSCoW (Should have)

Estimación: L (Large)

Criterios de Aceptación:

1. Dado que soy un usuario registrado, cuando entro en un evento, entonces debo ver un campo para sugerir canciones dentro del género permitido.
2. Dado que he sugerido una canción, cuando otros usuarios vean la playlist del evento, entonces mi sugerencia debe estar visible.

Notas Técnicas:

- Crear una interfaz de usuario para que los usuarios puedan enviar sus canciones recomendadas.
- Integrar con APIs de música para mostrar las canciones disponibles para sugerir.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-002] La interfaz de sugerencias puede no ser atractiva o ser poco intuitiva si no está bien diseñada.

[HU-003] - Votación de canciones sugeridas por otros usuarios**Descripción:**

Como usuario,
quiero poder votar canciones recomendadas por otros usuarios,
para que las más populares suenen durante el evento.

Prioridad: MoSCoW (Must have)

Estimación: M (Medium)

Criterios de Aceptación:

1. Dado que soy un usuario registrado, cuando veo una lista de canciones sugeridas en el evento, entonces debo poder votar por mis canciones favoritas.
2. Dado que he votado, cuando se muestre la lista final de canciones, entonces las más votadas deben ser destacadas.

Notas Técnicas:

- Implementar un sistema de votación con un límite de votos por usuario.
- Mostrar en tiempo real el ranking de las canciones más votadas.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.
- [ID-RN-003] Solo usuarios autenticados pueden votar, comentar o marcar eventos como favoritos
- [ID-RN-005] Las empresas no podrán modificar votaciones ni comentarios de usuarios.

Riesgos Asociados:

- [ID-RIESGO-003] Los votos pueden ser manipulados si no se implementa una validación adecuada.

[HU-004] - Notificaciones y recordatorios de eventos

Descripción:

Como usuario,
quiero recibir notificaciones de nuevos eventos en base a mis preferencias,
para estar al tanto de los eventos de mis lugares favoritos.

Prioridad: MoSCoW (Must have)

Estimación: S (Small)

Criterios de Aceptación:

1. Dado que soy un usuario registrado y ya configuré mis preferencias, cuando se publique un evento que se alinee con mis gustos, entonces debo recibir una notificación.
2. Dado que un evento se acerca, cuando la fecha esté próxima, entonces debo recibir un recordatorio.

Notas Técnicas:

- Implementar un sistema de notificaciones permitiendo al usuario gestionarlas.

Reglas de Negocio Asociadas:

- [ID-RN-006] El sistema debe ser utilizado principalmente por navegadores de escritorio.
- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-004] La saturación de notificaciones podría generar molestias.

Wireframe Asociados:

- [Vista Inicial Usuario](#)

[HU-005] - Creación de eventos por parte de empresas organizadoras.

Descripción:

Como empresa organizadora,
quiero poder crear eventos y definir el género musical,
para que los usuarios puedan asistir y sugerir canciones.

Prioridad: MoSCoW (Must have)

Estimación: L (Large)

Criterios de Aceptación:

1. Dado que soy una empresa registrada, cuando accedo a la sección de creación de eventos, entonces debo poder definir el género musical, la fecha y la descripción del evento.
2. Dado que he creado un evento, cuando lo publico, entonces los usuarios deben poder ver y sugerir canciones para el evento.

Notas Técnicas:

- Crear un formulario para que los organizadores ingresen los detalles del evento.

Reglas de Negocio Asociadas:

- [ID-RN-002] El sistema solo permitirá la carga de eventos a cuentas registradas como "empresa".

Riesgos Asociados:

- [ID-RIESGO-005] El proceso de creación de eventos podría ser demasiado complejo si no se simplifica.

Wireframe Asociados:

- [Vista Inicial Empresa](#)

[HU-006] - Visualización del perfil del evento**Descripción:**

Como usuario,
quiero ver toda la información del evento,
para decidir si me interesa participar.

Prioridad: MoSCoW (Must have)

Estimación: M (Medium)

Criterios de Aceptación:

1. Dado que soy un usuario registrado, cuando accedo al perfil de un evento, entonces debo poder ver detalles como lugar, fecha, género musical, canciones sugeridas y asistentes.

Notas Técnicas:

- Implementar una base de datos con la información relevante de los eventos.
- Crear una interfaz visual atractiva para mostrar los detalles del evento de manera clara.

Reglas de Negocio Asociadas:

- [ID-RN-003] Solo usuarios autenticados pueden votar, comentar o marcar eventos como favoritos.

- [ID-RN-005] Las empresas deben adaptar sus eventos basándose en la retroalimentación del público.

Riesgos Asociados:

- [ID-RIESGO-006] La información del evento podría estar desactualizada si no se actualiza en tiempo real.

Wireframe Asociados:

- Wireframe1: [Vista Evento Empresa](#)
- Wireframe2: [Vista Evento Usuario](#)

Nota:**Archivos Adjuntos: (HU-001 hasta HU-006)**

- Diagramas especificados en las fases anteriores: Diagrama contextual, de requerimientos y de clases.
- Wireframes: [Vista General](#)

HISTORIAS DE USUARIOS POR FEATURE (Agregadas)***[HU-001] - Selección de géneros al momento de registrarse*****Historia de Usuario:**

Como usuario nuevo,
quiero seleccionar mis géneros musicales favoritos durante el registro,
para personalizar mis recomendaciones de eventos y canciones desde el inicio.

Prioridad: MoSCoW (Must Have)

Estimación: S (Small)

Criterios de Aceptación:

- Dado que soy un usuario nuevo, cuando me registre en la plataforma, entonces debo poder seleccionar uno o varios géneros musicales favoritos.
- Dado que he seleccionado mis géneros musicales, entonces debo ver que mi elección se guarda correctamente en mi perfil.

Notas Técnicas:

- Crear un formulario durante el proceso de registro que permita seleccionar géneros musicales a través de una interfaz clara (por ejemplo, checkboxes).
- La selección de géneros será almacenada en la base de datos asociada al perfil del usuario.
- Asegurar que al menos un género sea seleccionado antes de permitir que el usuario complete el registro.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-001] Posible dificultad en el diseño de una interfaz de usuario atractiva y fácil de usar para la selección de géneros.
- [ID-RIESGO-002] La base de datos podría no soportar eficientemente un alto número de géneros asociados si no se optimiza correctamente.

[HU-002] - Edición de géneros desde el perfil**Historia de Usuario:**

Como usuario registrado,
quiero poder cambiar mis géneros musicales favoritos en cualquier momento desde mi perfil,
para ajustar mis preferencias de eventos y canciones a medida que mis gustos cambian.

Prioridad: MoSCoW (Must Have)

Estimación: M (Medium)

Criterios de Aceptación:

- Dado que soy un usuario registrado, cuando accedo a mi perfil, entonces debo poder agregar, quitar o cambiar los géneros musicales favoritos.

- Dado que he editado mis géneros musicales, cuando guarde los cambios, entonces estos deben reflejarse correctamente en mi perfil y en las recomendaciones de eventos y canciones.

Notas Técnicas:

- Crear una sección en el perfil del usuario donde pueda visualizar y modificar sus géneros musicales favoritos.
- Actualizar la base de datos con los nuevos géneros seleccionados por el usuario.
- Integrar con el sistema de recomendaciones para que las sugerencias cambien dinámicamente según las nuevas preferencias.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-003] La interfaz podría no ser lo suficientemente intuitiva, lo que podría llevar a que los usuarios no usen esta funcionalidad.
- [ID-RIESGO-004] Los cambios de género pueden afectar la precisión de las recomendaciones si no se actualizan correctamente en tiempo real.

[HU-003] - Sistema de guardado de géneros asociados al usuario**Historia de Usuario:**

Como usuario registrado,
quiero que mis géneros musicales favoritos se guarden automáticamente en mi perfil,
para no tener que seleccionar mis géneros cada vez que inicie sesión.

Prioridad: MoSCoW (Must Have)

Estimación: M (Medium)

Criterios de Aceptación:

- Dado que soy un usuario registrado, cuando seleccione mis géneros musicales, estos deben guardarse de forma automática en mi perfil.
- Dado que he guardado mis géneros, cuando inicie sesión en el futuro, debo poder ver los géneros previamente seleccionados.

Notas Técnicas:

- Desarrollar la lógica de almacenamiento en la base de datos para los géneros seleccionados por cada usuario.
- Asegurar la sincronización correcta entre la base de datos y el perfil del usuario para que las preferencias se mantengan a lo largo del tiempo.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-005] Posibles problemas de sincronización de datos si la base de datos no se actualiza correctamente, lo que puede generar inconsistencias entre el perfil y las recomendaciones.

[HU-004] - Personalización de eventos y canciones basados en géneros favoritos**Historia de Usuario:**

Como usuario registrado,
quiero recibir recomendaciones de eventos y canciones basadas en mis géneros musicales favoritos,
para disfrutar de una experiencia más personalizada y acorde a mis gustos.

Prioridad: MoSCoW (Should Have)

Estimación: L (Large)

Criterios de Aceptación:

- Dado que soy un usuario registrado con géneros favoritos seleccionados, cuando acceda a las recomendaciones de eventos, entonces estos deben estar alineados con mis géneros musicales.
- Dado que soy un usuario registrado, cuando acceda a las recomendaciones de canciones, estas deben estar alineadas con mis géneros favoritos.

Notas Técnicas:

- Desarrollar un algoritmo de recomendación que priorice eventos y canciones basados en los géneros favoritos del usuario.
- Asegurarse de que el sistema de recomendaciones sea dinámico, es decir, que se actualice con cada modificación en los géneros seleccionados por el usuario.
- Integrar con APIs externas para obtener canciones relacionadas con los géneros preferidos.

Reglas de Negocio Asociadas:

- [ID-RN-004] Los usuarios deben configurar sus preferencias antes de recibir sugerencias personalizadas.

Riesgos Asociados:

- [ID-RIESGO-006] Las recomendaciones podrían no ser precisas si el algoritmo no tiene suficiente información sobre los géneros o si los géneros no se actualizan correctamente en la base de datos.

[HU-005] - Habilitar sugerencia de canciones**Descripción:**

Como usuario inscripto en un evento, quiero poder sugerir canciones dentro del género permitido del evento, para participar activamente en la creación de la playlist del evento y sentirme parte de la experiencia.

Prioridad: Alta
Estimación: L

Criterios de Aceptación:

- Dado que el usuario está autenticado e inscripto en un evento,
Cuando accede a la sección de sugerencias del evento,
Entonces puede ingresar el nombre de una canción y su artista, y enviarla como sugerencia.
- Dado que el evento tiene un género musical definido,
Cuando el usuario propone una canción,
Entonces el sistema valida que esté dentro del género permitido antes de aceptarla.
- Dado que existe un límite de sugerencias por usuario,
Cuando el usuario intenta enviar más canciones que el máximo permitido,
Entonces el sistema bloquea la acción y muestra un mensaje explicativo.

Notas Técnicas:

- Validar que el usuario esté inscripto en el evento antes de permitir sugerencias.
- UI: Formulario simple con campos "Nombre de la canción" y "Artista".

Reglas de Negocio Asociadas:

- [ID-RN-003] Solo los usuarios inscriptos en un evento pueden sugerir canciones.
- [ID-RN-005] Las canciones sugeridas deben coincidir con el género definido por el evento.
- [ID-RN-006] Cada usuario podrá sugerir hasta un máximo preconfigurado de canciones por evento.

Riesgos Asociados:

- [ID-RIESGO-002] El sistema podría recibir sugerencias irrelevantes si la validación del género no es precisa.

[HU-006] - Visualización pública de sugerencias

Descripción:

Como usuario inscripto en un evento,
quiero poder ver las canciones sugeridas por otros asistentes,
para descubrir nuevas opciones musicales y votar por las que me gusten.

Prioridad: Alta

Estimación: M

Criterios de Aceptación:

- Dado que el usuario está inscripto en un evento,
Cuando accede a la sección de sugerencias,
Entonces puede visualizar una lista con todas las canciones sugeridas para ese evento.
- Dado que otros usuarios han hecho sugerencias,
Cuando el usuario visualiza la lista,
Entonces puede ver el nombre de la canción, el artista y el nombre de quien la propuso.
- Dado que hay múltiples sugerencias,
Cuando el usuario navega por la lista,
Entonces puede ordenarla por más recientes o más votadas.

Notas Técnicas:

- La visualización puede hacerse con una lista dinámica.
- Incluir íconos o elementos UI que identifiquen fácilmente a los usuarios y sus propuestas.

Reglas de Negocio Asociadas:

- [ID-RN-008] Las sugerencias deben incluir nombre de la canción, artista y nombre del usuario que la propuso.

Riesgos Asociados:

- [ID-RIESGO-003] Posible sobrecarga visual si hay demasiadas sugerencias sin filtros u ordenamiento.

[HU-007] - Sistema de votación sobre canciones sugeridas

Descripción:

Como usuario inscripto en un evento,
quiero poder votar canciones sugeridas por otros asistentes
para influir en la selección de las canciones que sonarán durante el evento.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el usuario está autenticado e inscripto en el evento,
Cuando visualiza la lista de canciones sugeridas,
Entonces puede votar positivamente o negativamente cada canción una sola vez.
- Dado que un usuario ya votó por una canción,
Cuando intenta volver a votar sobre la misma,
Entonces el sistema le informa que ya ha emitido su voto y no permite modificarlo.
- Dado que las votaciones están habilitadas,
Cuando el usuario emite un voto,
Entonces el conteo de votos de la canción se actualiza en tiempo real.

Notas Técnicas:

- Validar que cada usuario vote solo una vez por sugerencia.

Reglas de Negocio Asociadas:

- [ID-RN-009] Cada usuario puede votar una sola vez por canción sugerida.
- [ID-RN-010] Los votos no pueden editarse ni eliminarse una vez emitidos.

Riesgos Asociados:

- [ID-RIESGO-004] Manipulación de votos por múltiples cuentas o bots.

[HU-008] - Visualización de ranking de canciones sugeridas

Descripción:

Como usuario inscripto en un evento,
quiero ver una lista ordenada por cantidad de votos positivos,
para conocer qué canciones tienen mayor aceptación por parte del público.

Prioridad: Should have

Estimación: S

Criterios de Aceptación:

- Dado que existen canciones sugeridas con votos,
Cuando el usuario accede a la sección de ranking,
Entonces el sistema muestra las canciones ordenadas de mayor a menor según su puntaje.
- Dado que se muestran múltiples canciones,
Cuando el usuario navega la lista,
Entonces puede ver el número total de votos positivos y negativos junto a cada una.

Notas Técnicas:

- El orden del ranking debe ser de mayor a menor, destacando los que mayores votos tuvieron.
- Incluir ícono de popularidad si se desea destacar las top 3.

Reglas de Negocio Asociadas:

- [ID-RN-011] Las canciones sugeridas deben ordenarse automáticamente por votos.

Riesgos Asociados:

- [ID-RIESGO-005] Confusión del usuario si no se muestra claramente la diferencia entre votos positivos y negativos.

[HU-009] - Cierre de votaciones previo al evento

Descripción:

Como administrador del evento,
quiero establecer un límite de tiempo para el cierre de las votaciones,
para poder definir la playlist final antes de que comience el evento.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que la fecha del evento se aproxima,
Cuando se alcanza el tiempo configurado para cierre,
Entonces el sistema bloquea automáticamente nuevas votaciones.
- Dado que el usuario intenta votar fuera de plazo,
Cuando accede a la sección de votación,
Entonces el sistema muestra un mensaje que indica que las votaciones ya están cerradas.

Notas Técnicas:

- Agregar una función para: fecha de cierre votación en los eventos.
- Ocultar o desactivar botones de votación luego del cierre.

Reglas de Negocio Asociadas:

- [ID-RN-012] El sistema debe impedir nuevas votaciones una vez superada la fecha límite.

Riesgos Asociados:

- [ID-RIESGO-006] Posible pérdida de votos si los usuarios no están informados del cierre anticipadamente.

[HU-010] - Configuración de notificaciones en perfil

Descripción:

Como usuario registrado,
quiero poder configurar qué tipo de notificaciones deseo recibir,
para controlar la información que me llega según mis preferencias.

Prioridad: Must have

Estimación: L (tamaño grande)

Criterios de Aceptación:

- Dado que el usuario accede a su perfil, cuando selecciona las opciones de notificaciones, entonces puede activar o desactivar tipos específicos como recordatorios, cambios o nuevos eventos.
- Dado que el usuario actualiza su configuración, cuando guarda los cambios, entonces estos se aplican inmediatamente en el sistema.

Notas Técnicas:

- Crear una sección de "Preferencias de notificación" en el perfil del usuario.
- Utilizar checkboxes para permitir activación/desactivación de tipos.
- Almacenar la configuración en la base de datos asociada al usuario.
- Controlar qué notificaciones se envían en base a estas preferencias.

Reglas de Negocio Asociadas:

- [ID-RN-012] Solo usuarios autenticados pueden configurar sus notificaciones.
- [ID-RN-013] Las preferencias se aplican a todas las sesiones activas del usuario.

Riesgos Asociados:

- [ID-RIESGO-007] Riesgo de que el sistema no respete las preferencias configuradas si la lógica no se aplica correctamente.

[HU-011] - Notificaciones de recordatorio de evento

Descripción:

Como usuario registrado,
quiero recibir un recordatorio 24-48 horas antes de un evento,
para asegurarme de no olvidarme de asistir.

Prioridad: Should have

Estimación: S (tamaño pequeño)

Criterios de Aceptación:

- Dado que el usuario está inscripto a un evento, cuando faltan entre 24 y 48 horas, entonces el sistema debe enviar una notificación con la fecha, hora y ubicación.
- Dado que el usuario tiene recordatorios desactivados, cuando se acerca la fecha del evento, entonces no se debe enviar ningún aviso.

Notas Técnicas:

- Crear cron job para verificar eventos próximos cada 1 hora.
- Integrar con servicio de envío de notificaciones (ej. Firebase, OneSignal, email o push).
- Incluir datos relevantes: nombre del evento, fecha, hora, ubicación.

Reglas de Negocio Asociadas:

- [ID-RN-014] Solo se enviarán recordatorios si el usuario está inscripto y tiene activa la opción correspondiente.

Riesgos Asociados:

- [ID-RIESGO-008] Riesgo de saturación de notificaciones si se programan múltiples eventos seguidos.

[HU-012] - Notificaciones de cambios en eventos

Descripción:

Como usuario registrado, quiero recibir una alerta inmediata cuando un evento al que estoy inscripto cambia, para mantenerme informado y adaptar mis planes.

Prioridad: Must have

Estimación: M (tamaño medio)

Criterios de Aceptación:

- Dado que el evento cambia de horario, sede o se cancela, cuando se actualiza la información, entonces se debe enviar una notificación inmediata a todos los usuarios inscriptos.
- Dado que el usuario no está inscripto en el evento, cuando se produce un cambio, entonces no debe recibir ninguna notificación.

Notas Técnicas:

- Detectar cambios en los campos clave (fecha, hora, ubicación, estado).
- Emitir trigger o evento del sistema para lanzar notificaciones.
- Incluir información actualizada y clara en el mensaje.

Reglas de Negocio Asociadas:

- [ID-RN-015] Toda modificación de evento debe disparar una evaluación de alertas.
- [ID-RN-016] Las notificaciones deben enviarse dentro de los 5 minutos posteriores al cambio.

Riesgos Asociados:

- [ID-RIESGO-009] Retrasos en la entrega de notificaciones por carga alta del sistema.

[HU-013] - Notificaciones de nuevos eventos basados en preferencias

Descripción:

Como usuario registrado,
quiero que el sistema me notifique sobre eventos nuevos que coincidan con mis gustos,
para poder inscribirme en eventos de mi interés sin tener que buscarlos manualmente.

Prioridad: Could have

Estimación: M (tamaño medio)

Criterios de Aceptación:

- Dado que se crea un evento nuevo, cuando coincide con las preferencias de género y/o ubicación de un usuario, entonces se debe enviar una notificación.
- Dado que el usuario no tiene activada la opción de recibir nuevas sugerencias, cuando se cree un evento, entonces no se debe enviar ninguna alerta.

Notas Técnicas:

- Filtrar nuevos eventos con coincidencias basadas en género y ubicación preferidas del usuario.
- Consultas periódicas o eventos reactivos al alta de nuevos eventos.
- Enviar notificación personalizada (ej. "Nuevo evento de tu interés en tu zona").

Reglas de Negocio Asociadas:

- [ID-RN-017] Las preferencias deben estar actualizadas y habilitadas para recibir recomendaciones.
- [ID-RN-018] Solo se enviarán notificaciones si el evento fue publicado públicamente.

Riesgos Asociados:

- [ID-RIESGO-010] Baja precisión en el algoritmo de coincidencia de preferencias puede generar spam.

[HU-014] - Visualizar lugar del evento

Descripción:

Como usuario interesado en un evento, quiero ver el nombre y dirección del lugar, y un mapa opcional, para saber con claridad dónde se realizará el evento.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que el usuario está visualizando el perfil de un evento, Cuando accede a la sección de detalles, Entonces puede ver el nombre del lugar y la dirección completa del evento.

Notas Técnicas:

- Asegurarse de que las direcciones se validen correctamente en el backend para evitar errores en la visualización.

Reglas de Negocio:

- [ID-RN-019] Todo evento debe tener un lugar visible con nombre y dirección.

Riesgos:

- [ID-RIESGO-011] Puede haber inconsistencias si falta información de dirección o está mal cargada.

[HU-015] - Visualizar fecha y hora del evento**Descripción:**

Como usuario que está evaluando asistir a un evento, quiero ver claramente la fecha y hora de inicio, para organizar mi asistencia.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que el evento tiene una fecha definida,
Cuando el usuario accede al perfil del evento,
Entonces puede ver la fecha completa (día, mes, año) y la hora de inicio del evento.

Notas Técnicas:

- La fecha y hora deben estar en un formato claro y entendible, dependiendo de la zona horaria del usuario.

Reglas de Negocio:

- [ID-RN-020] La fecha y hora del evento deben mostrarse siempre en un formato claro y localizable para el usuario.

Riesgos:

- [ID-RIESGO-013] Errores de formato pueden causar confusión
- [ID-RIESGO-014] Horarios incorrectos o inconsistentes podrían llevar a malentendidos.

[HU-016] - Visualizar géneros musicales del evento

Descripción:

Como usuario con preferencias musicales,
quiero ver los géneros musicales asociados a un evento,
para decidir si coincide con mis gustos.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que un evento tiene géneros musicales asignados,
Cuando el usuario visualiza el perfil del evento,
Entonces puede ver la lista de géneros musicales del evento.
- Dado que hay múltiples géneros musicales,
Cuando se visualiza la lista,
Entonces los géneros deben estar claramente diferenciados

Notas Técnicas:

- Los géneros musicales deben ser parte de la base de datos del evento, y la visualización debe ser dinámica según los géneros asociados al evento.
- Utilizar iconos o colores diferenciados para mejorar la experiencia del usuario en la visualización de los géneros.

Reglas de Negocio:

- [ID-RN-022] Cada evento debe tener al menos un género musical definido y visible para el usuario.

Riesgos:

- [ID-RIESGO-015] Géneros mal categorizados pueden confundir o desmotivar a los usuarios.
- [ID-RIESGO-016] Falta de visualización clara puede reducir el interés del usuario.

[HU-017] - Visualizar canciones sugeridas y votadas

Descripción:

Como usuario interesado en la música del evento,
quiero ver qué canciones se han sugerido y cuántos votos tienen,
para conocer el ambiente musical previsto.

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que hay canciones sugeridas para el evento,
Cuando el usuario accede a la lista de canciones,
Entonces puede ver una lista de títulos de canciones sugeridas, junto con su cantidad de votos (positivos y negativos).
- Dado que hay múltiples canciones sugeridas,
Cuando el usuario visualiza la lista,
Entonces las canciones deben estar ordenadas de mayor a menor según su cantidad de votos.

Notas Técnicas:

- Los votos deben ser gestionados en el backend para asegurar que las canciones se muestren en orden correcto. Puede implementarse con un sistema de puntuación o conteo de votos.
- Incluir botones o controles interactivos para que los usuarios voten de manera sencilla y rápida.

Reglas de Negocio:

- [ID-RN-023] Las canciones sugeridas deben estar ordenadas por relevancia (votos).
- [ID-RN-024] Deben visualizarse los votos asociados a cada canción (puede ser visual o numérico).

Riesgos:

- [ID-RIESGO-017] Gran cantidad de canciones puede ralentizar la carga o dificultar la navegación.
- [ID-RIESGO-018] Mal manejo de votos podría llevar a inconsistencias en el orden o datos erróneos.

[HU-018] - Visualizar lista de asistentes confirmados

Descripción:

Como usuario que evalúa asistir a un evento,
quiero ver cuántas personas asistirán y quiénes son,
para conocer el interés general y si hay gente que conozco.

Prioridad: Could have

Estimación: M

Criterios de Aceptación:

- Dado que hay usuarios confirmados para un evento,
Cuando el usuario accede al perfil del evento,
Entonces puede ver la cantidad total de asistentes confirmados.
- Dado que el evento tiene muchos asistentes,
Cuando el usuario visualiza la lista de asistentes,
Entonces se puede mostrar una lista breve con algunos nombres de los asistentes.

Notas Técnicas:

- La lista de asistentes debe manejarse de manera eficiente, especialmente si el número de usuarios es elevado.
- Implementar mecanismos para garantizar la privacidad de los usuarios si se elige mostrar nombres.

Reglas de Negocio:

- [ID-RN-025] El sistema debe mostrar cuántas personas confirmaron asistencia.
- [ID-RN-026] Mostrar nombres/avatars es opcional y configurable.

Riesgos:

- [ID-RIESGO-019] Mostrar nombres podría generar conflictos con la privacidad de usuarios.
- [ID-RIESGO-020] Si hay muchos asistentes, la carga puede volverse lenta o generar errores.

[HU-019] - Formulario de creación de eventos

Descripción:

Como organizador,

quiero tener un formulario para crear eventos,
para ingresar toda la información relevante de manera sencilla.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el organizador accede a la sección de creación de eventos,
Cuando abre el formulario,
Entonces debe ver los campos de nombre, descripción, fecha, hora, lugar y género musical.
- Dado que el organizador llena el formulario,
Cuando hace clic en el botón de guardar,
Entonces el sistema debe validar que todos los campos obligatorios estén completos y mostrar un mensaje de error si falta alguno.

Notas Técnicas:

- Asegurarse de validar todos los campos, especialmente los obligatorios.
- El formulario debe ser sencillo y claro, con mensajes de error intuitivos.

Reglas de Negocio:

- [ID-RN-027] El formulario debe exigir los campos nombre, fecha, hora, lugar y género musical como obligatorios.
- [ID-RN-028] El organizador debe poder guardar el evento solo si todos los campos obligatorios están completos.

Riesgos:

- [ID-RIESGO-021] Si no se validan correctamente los campos obligatorios, el organizador podría guardar eventos incompletos.
- [ID-RIESGO-022] El formulario puede resultar confuso si los mensajes de error no son claros.

[HU-020] - Selector de género musical

Descripción:

Como organizador,
quiero poder seleccionar uno o varios géneros musicales para el evento,
para definir el tipo de música que se escuchará.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que el organizador está en el formulario de creación de eventos,
Cuando accede a la sección de género musical,
Entonces debe poder seleccionar uno o más géneros musicales de una lista predefinida.
- Dado que el organizador selecciona los géneros,
Cuando guarda el evento,
Entonces el sistema debe guardar los géneros seleccionados para el evento.

Notas Técnicas:

- Utilizar un componente de selección múltiple para permitir que el organizador elija uno o varios géneros.

Reglas de Negocio:

- [ID-RN-029] El organizador debe poder seleccionar uno o más géneros musicales para cada evento.
- [ID-RN-030] Los géneros musicales deben provenir de una lista predefinida en la base de datos.

Riesgos:

- [ID-RIESGO-023] Los géneros pueden no estar bien categorizados, lo que podría causar confusión.
- [ID-RIESGO-024] Puede haber problemas con la escalabilidad si los géneros no están bien organizados en la base de datos.

[HU-021] - Sistema de guardado de eventos

Descripción:

Como organizador,
quiero que mi evento se guarde correctamente en la base de datos,
para que otros usuarios puedan ver los detalles del evento.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el organizador ha llenado el formulario de creación de evento,
Cuando hace clic en guardar,
Entonces el sistema debe guardar todos los datos del evento en la base de datos,
vinculándolos al organizador.
- Dado que el evento se guarda correctamente,
Cuando otro usuario busca eventos,
Entonces debe poder ver el evento creado por el organizador.

Notas Técnicas:

- Verificar que todos los datos se han guardado correctamente y que el evento está asociado al organizador.

Reglas de Negocio:

- [ID-RN-031] El evento debe estar vinculado al organizador que lo crea en la base de datos.
- [ID-RN-032] Los eventos deben guardarse con todos los datos completados y deben estar disponibles para la visualización pública.

Riesgos:

- [ID-RIESGO-025] Si la base de datos no está correctamente configurada, el evento podría no guardarse correctamente.

- [ID-RIESGO-026] El sistema podría tener fallos si no se gestionan correctamente las relaciones entre organizadores y eventos.

[HU-022] - Gestión de propuestas de canciones

Descripción:

Como organizador,
quiero que los usuarios puedan sugerir y votar canciones para el evento,
para poder personalizar la lista musical en base a los gustos del público.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que el organizador ha creado el evento,
Cuando el evento está publicado,
Entonces los usuarios pueden sugerir canciones para el evento.
- Dado que los usuarios sugieren canciones,
Cuando una canción recibe suficientes votos,
Entonces el organizador podrá ver las canciones más populares en una lista.

Notas Técnicas:

- Implementar un sistema de votación donde cada usuario pueda votar por las canciones propuestas. Asegurarse de que el sistema permita votar solo una vez por canción.
- Almacenar las sugerencias y votos en una base de datos que esté asociada con cada evento.

Reglas de Negocio:

- [ID-RN-033] Los usuarios deben poder sugerir y votar por canciones en cualquier evento.
- [ID-RN-034] Solo el organizador podrá ver las canciones más votadas y decidir cuál incluir en el evento.

Riesgos:

- [ID-RIESGO-027] Si el sistema de votación no está bien implementado, los usuarios podrían votar múltiples veces, afectando la integridad de los datos.
- [ID-RIESGO-028] Los organizadores pueden no estar contentos con la selección de canciones si el sistema de votación no refleja correctamente el gusto del público.

[HU-023] - Validaciones básicas en el formulario

Descripción:

Como organizador,
quiero que el formulario de creación de eventos valide los campos obligatorios antes de guardarlo,
para asegurar que no se queden eventos incompletos.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que el organizador está completando el formulario de creación de evento,
Cuando intenta guardar el evento,
Entonces el sistema debe validar que todos los campos obligatorios (nombre, fecha, hora, lugar, género musical) estén completos.
- Dado que alguno de los campos obligatorios está vacío,
Cuando el organizador intenta guardar el evento,
Entonces el sistema debe mostrar un mensaje de error para los campos faltantes.

Notas Técnicas:

- Utilizar validaciones del lado del cliente y servidor para asegurarse de que todos los campos estén completos.
- Mostrar mensajes de error claros y útiles si el organizador intenta guardar el evento con campos vacíos.

Reglas de Negocio:

- [ID-RN-035] El formulario debe validar que todos los campos obligatorios estén completos antes de permitir guardar el evento.

Riesgos:

- [ID-RIESGO-029] La validación incorrecta de los campos obligatorios podría permitir que el evento se guarde incompleto, lo que afectaría la calidad de la aplicación.

[HU-024] - Botón de reproducción junto a cada sugerencia**Descripción:**

Como *usuario*,
quiero *ver un botón de reproducción junto a cada sugerencia de canción*,
para *poder escuchar fácilmente la música antes de votar o sugerirla*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un usuario visualiza una lista de canciones sugeridas, cuando la interfaz se carga, entonces cada canción debe mostrar un botón de reproducción.
- Dado que un usuario presiona el botón de reproducción, cuando hay disponibilidad desde alguna fuente (Spotify o YouTube), entonces se debe comenzar a reproducir el audio/video correspondiente.

Notas Técnicas:

- Implementar botón visible junto a cada entrada de canción en el frontend.
- Asociar eventos a acciones de reproducción vía API según disponibilidad.
- Estilo visual no debe interferir con el flujo principal de navegación.

Reglas de Negocio Asociadas:

- [ID-RN-036] Las canciones sugeridas deben poder reproducirse sin necesidad de salir de la app.
- [ID-RN-037] Solo los usuarios autenticados pueden interactuar con sugerencias.

Riesgos Asociados:

- [ID-RIESGO-030] Posibles problemas de UI si el diseño del botón interfiere con otros elementos.
- [ID-RIESGO-031] Latencia al cargar previews desde fuentes externas.

[HU-025] - Integración con API de Spotify**Descripción:**

*Como usuario con cuenta de Spotify,
quiero escuchar un preview de 30 segundos o canción completa,
para decidir si me gusta y quiero votarla o no.*

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el usuario tiene sesión activa en Spotify, cuando presiona el botón de reproducción, entonces debe escucharse el preview o canción completa desde Spotify.
- Dado que la canción no tiene preview disponible, cuando se detecta la falta, entonces debe dispararse el sistema de fallback.

Notas Técnicas:

- Usar Spotify Web API para autenticación y previews (/tracks/{id} endpoint).
- Requiere token OAuth del usuario.
- Manejo de errores si el contenido no está disponible por región o permisos.

Reglas de Negocio Asociadas:

- [ID-RN-038] Las canciones sugeridas deben poder reproducirse sin necesidad de salir de la app.
- [ID-RN-039] El contenido reproducido debe respetar los permisos de cuenta del usuario.

Riesgos Asociados:

- [ID-RIESGO-032] Spotify limita contenido por región o tipo de cuenta.
- [ID-RIESGO-033] El usuario puede no tener cuenta de Spotify o no estar logueado.

[HU-026] - Integración con API de YouTube

Descripción:

Como *usuario*,
quiero *reproducir canciones sugeridas desde YouTube como alternativa*,
para *escuchar la canción completa si no tengo Spotify*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el usuario no tiene cuenta de Spotify, cuando presiona el botón de reproducción, entonces debe buscarse y reproducirse el video desde YouTube.
- Dado que se encuentra un video correspondiente, cuando se integra, entonces debe mostrarse embebido en la interfaz sin abrir una nueva pestaña.

Notas Técnicas:

- Usar YouTube Data API para búsqueda y reproducción embebida (iframe).
- Considerar integración con youtube-iframe-player.
- Validar disponibilidad del video y evitar contenido inapropiado.

Reglas de Negocio Asociadas:

- [ID-RN-040] Las canciones sugeridas deben poder reproducirse sin necesidad de salir de la app.
- [ID-RN-041] Debe evitarse la reproducción de contenido no musical o inadecuado.

Riesgos Asociados:

- [ID-RIESGO-034] Resultados inexactos en la búsqueda automática de canciones.
- [ID-RIESGO-035] Publicidad o contenido irrelevante en los videos embebidos.

[HU-027] - Sistema de fallback**Descripción:**

Como usuario,
quiero que la app intente reproducir desde YouTube si Spotify no está disponible,
para asegurarme de siempre poder escuchar la canción.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que la canción no está disponible en Spotify, cuando el usuario intenta reproducir, entonces debe buscarse automáticamente en YouTube.
- Dado que YouTube tampoco devuelve un resultado válido, cuando eso ocurre, entonces debe notificarse al usuario que la canción no puede reproducirse.

Notas Técnicas:

- Condicional en backend o frontend que detecte ausencia de preview en Spotify.
- Lógica para consulta automática a YouTube en segundo plano.
- Feedback al usuario ante fallo completo de ambas fuentes.

Reglas de Negocio Asociadas:

- [ID-RN-042] El sistema debe ofrecer siempre al menos una alternativa de escucha si existe.
- [ID-RN-043] Las canciones sugeridas deben poder reproducirse sin necesidad de salir de la app.

Riesgos Asociados:

- [ID-RIESGO-036] Errores en la lógica de fallback que impidan reproducir desde cualquier fuente.
- [ID-RIESGO-037] Exceso de llamadas API y límites superados (rate limiting).

[HU-028] - Control básico de reproducción**Descripción:**

Como *usuario*,
quiero *poder pausar, reanudar y controlar el volumen de la reproducción*,
para *tener una experiencia fluida y cómoda mientras navego la app*.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que una canción está reproduciéndose, cuando el usuario presiona el botón de pausa, entonces la canción debe detenerse.
- Dado que el usuario ajusta el volumen, cuando lo hace desde la app, entonces debe cambiarse sin afectar otros sonidos del dispositivo.

Notas Técnicas:

- Controles UI integrados en el componente reproductor (play/pause, volumen).
- Reutilizar componentes de Spotify/YouTube si ya los incluyen o implementar overlay simple.
- Mantener diseño accesible para usuarios con dificultades visuales o auditivas.

Reglas de Negocio Asociadas:

- [ID-RN-044] Las canciones sugeridas deben poder reproducirse sin necesidad de salir de la app.

Riesgos Asociados:

- [ID-RIESGO-038] Reproducción simultánea de múltiples canciones.

- [ID-RIESGO-039] Incompatibilidades de control de volumen entre plataformas.

[HU-029] - Definir categorías de logros

Descripción:

*Como equipo de producto,
quiero definir tipos de badges (asistencia, sugerencias, votos, etc.)
para establecer las condiciones que premian la participación del usuario.*

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado un conjunto de acciones posibles de los usuarios, Cuando se identifican patrones de participación, Entonces se debe asignar una categoría de badge correspondiente.
- Dado que un badge es creado, Cuando se visualiza en la documentación del sistema, Entonces debe incluir su nombre, descripción y criterios.

Notas Técnicas:

- Crear una tabla de categorías y condiciones asociadas.
- Definir niveles si aplica (ej. "Asistente novato", "Asistente experto").
- Guardar la definición en base de datos para uso dinámico.

Reglas de Negocio Asociadas:

- [ID-RN-045] Cada tipo de participación debe tener al menos un badge asignable.
- [ID-RN-046] Los criterios para obtener cada badge deben estar predefinidos y visibles para los usuarios.

Riesgos Asociados:

- [ID-RIESGO-041] Confusión de usuarios si los criterios de los logros no son claros o accesibles.

[HU-030] - Asignación automática de badges

Descripción:

Como *sistema*,
quiero *asignar automáticamente badges a los usuarios cuando alcanzan ciertos hitos*
para *recompensar su participación sin intervención manual*.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que un usuario alcanza un hito, Cuando cumple los criterios establecidos, Entonces el sistema le asigna el badge correspondiente.
- Dado que un badge es asignado, Cuando el usuario accede a su perfil, Entonces el nuevo badge debe ser visible.

Notas Técnicas:

- Trigger automático al actualizar las estadísticas del usuario.
- Revisión diaria/semanal de participación mediante cron jobs o eventos en tiempo real.
- Notificación automática al usuario cuando recibe un nuevo badge.

Reglas de Negocio Asociadas:

- [ID-RN-047] Los badges deben ser asignados sin intervención manual del administrador.
- [ID-RN-048] Un usuario no puede recibir el mismo badge más de una vez, salvo que tenga niveles.

Riesgos Asociados:

- [ID-RIESGO-042] Errores de lógica que asignen badges incorrectamente o en exceso.

[HU-031] - Visualización de badges en el perfil

Descripción:

Como *usuario*,
quiero *ver mis logros y badges en mi perfil público*
para *mostrar mi nivel de participación en la comunidad*.

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que un usuario accede a su perfil, Cuando lo visualiza desde la app, Entonces debe ver los badges obtenidos con nombre e ícono.
- Dado que otro usuario accede al perfil de alguien más, Cuando consulta su historial, Entonces también debe ver sus logros públicos.

Notas Técnicas:

- Modificar la vista del perfil para incluir sección "Logros".
- Mostrar íconos distintivos por cada badge.
- Posibilidad de ordenar o filtrar por tipo de badge.

Reglas de Negocio Asociadas:

- [ID-RN-049] Todos los badges obtenidos deben estar visibles en el perfil público del usuario.
- [ID-RN-050] Los badges deben incluir nombre, ícono, y breve descripción al pasar el mouse o tocar.

Riesgos Asociados:

- [ID-RIESGO-043] Sobrecarga visual o pérdida de usabilidad si se muestran demasiados badges.

[HU-031] - Tracking de asistencia a eventos

Descripción:

Como *sistema*,
quiero *registrar automáticamente los eventos a los que asiste cada usuario*
para *utilizar esa información en futuras recomendaciones personalizadas*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un usuario marca asistencia o escanea código en un evento, Cuando el sistema procesa la acción, Entonces debe guardarse el evento en su historial.
- Dado que el usuario visualiza su historial, Cuando accede desde su perfil, Entonces debe ver la lista de eventos a los que asistió.

Notas Técnicas:

- Asociación de usuario con eventos en tabla intermedia.
- Requiere validación por código QR o check-in en la app.

Reglas de Negocio Asociadas:

- [ID-RN-051] Cada asistencia confirmada debe quedar registrada con fecha y evento.
- [ID-RN-052] Un evento solo puede registrarse una vez por usuario.

Riesgos Asociados:

- [ID-RIESGO-044] Fallos en el check-in automático pueden generar datos incompletos o inexactos.

[HU-032] - Registro de votos y géneros favoritos**Descripción:**

Como *sistema*,
quiero *registrar los géneros y canciones que vota o elige el usuario*
para *identificar sus preferencias musicales*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un usuario vota una canción o género, Cuando la acción se confirma, Entonces se actualizan sus preferencias en el perfil.
- Dado que se registra una nueva preferencia, Cuando se acumulen suficientes votos, Entonces debe reflejarse como género favorito.

Notas Técnicas:

- Guardar estadísticas por usuario (géneros más votados).
- Puede utilizarse para ranking interno de afinidad por género.

Reglas de Negocio Asociadas:

- [ID-RN-053] Las preferencias musicales deben actualizarse con cada voto registrado.
- [ID-RN-054] Los géneros favoritos deben calcularse dinámicamente según votos recientes.

Riesgos Asociados:

- [ID-RIESGO-045] Preferencias sesgadas si el usuario solo vota esporádicamente.

[HU-033] - Motor de recomendación básico

Descripción:

Como sistema de recomendación, quiero analizar historial de asistencia y preferencias musicales para sugerir eventos personalizados a cada usuario.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que un usuario tiene historial de asistencia y votos, Cuando el sistema calcula recomendaciones, Entonces debe sugerir eventos relacionados con sus gustos.
- Dado que el usuario no tiene historial, Cuando se accede al sistema, Entonces las sugerencias deben basarse solo en los géneros favoritos.

Notas Técnicas:

- Algoritmo basado en filtrado por contenido simple (sin machine learning).
- Utiliza datos cruzados de asistencia y votos.

Reglas de Negocio Asociadas:

- [ID-RN-055] Las sugerencias deben basarse en datos reales del usuario y no en rankings globales.
- [ID-RN-056] Si no hay historial, se priorizan géneros favoritos como fuente principal.

Riesgos Asociados:

- [ID-RIESGO-046] Recomendaciones poco relevantes si el usuario tiene pocos datos registrados.

[HU-034] - Sistema de recomendaciones en el inicio**Descripción:**

Como *usuario*,
quiero *ver eventos recomendados al ingresar a la app o en una sección destacada*
para *encontrar más fácilmente eventos acordes a mis intereses*.

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que el usuario abre la app, Cuando carga la pantalla de inicio, Entonces debe ver un carrusel o sección con eventos recomendados.
- Dado que el usuario no tiene recomendaciones, Cuando se accede, Entonces debe mostrarse un mensaje de “No hay suficientes datos aún” o sugerencias generales.

Notas Técnicas:

- Requiere integración del motor de recomendación con frontend.
- Priorizar eventos próximos en el tiempo entre los recomendados.

Reglas de Negocio Asociadas:

- [ID-RN-057] La sección de recomendaciones debe estar visible en la pantalla de inicio para todos los usuarios logueados.
- [ID-RN-058] Las recomendaciones deben ser actualizadas al menos una vez por día.

Riesgos Asociados:

- [ID-RIESGO-047] Saturación visual de la pantalla de inicio si se mezcla con otras secciones destacadas.

[HU-034] - Actualización dinámica de preferencias

Descripción:

Como *sistema*,
quiero *ajustar las recomendaciones en tiempo real conforme el usuario interactúa con la app*
para *reflejar con mayor precisión sus intereses actuales*.

Prioridad: Could have

Estimación: L

Criterios de Aceptación:

- Dado que el usuario interactúa frecuentemente con ciertos géneros, Cuando vota o asiste a nuevos eventos, Entonces las recomendaciones se actualizan dinámicamente.
- Dado que el usuario cambia su patrón de participación, Cuando se detecta un desvío significativo, Entonces el motor debe reflejar nuevas preferencias.

Notas Técnicas:

- Uso de eventos en tiempo real o batch diario.
- Análisis por ventanas de tiempo (últimos 30 días, 60 días).

Reglas de Negocio Asociadas:

- [ID-RN-059] El sistema debe ajustar las preferencias si cambia el patrón de asistencia o votación.
- [ID-RN-060] Las actualizaciones deben realizarse sin afectar el rendimiento de la app.

Riesgos Asociados:

- [ID-RIESGO-048] Alta carga computacional si se recalculan las recomendaciones constantemente.

[HU-035] - Configuración de sistema de logging unificado**Descripción:**

*Como equipo de desarrollo,
quiero establecer un sistema de logs unificado en toda la app
para centralizar errores, advertencias e información relevante del sistema.*

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que se ejecuta una acción crítica en la app, Cuando ocurre un evento relevante, Entonces debe generarse un log clasificado por nivel (INFO, WARNING, ERROR).
- Dado que se detecta un error, Cuando se genera el log, Entonces debe enviarse al sistema centralizado para su almacenamiento.

Notas Técnicas:

- Debe implementarse tanto en backend como en frontend.

Reglas de Negocio Asociadas:

- [ID-RN-061] Todos los logs deben ser recolectados en un repositorio centralizado.
- [ID-RN-062] Los niveles de log deben estar correctamente definidos y diferenciados.

Riesgos Asociados:

- [ID-RIESGO-049] Logs dispersos o mal categorizados dificultan la trazabilidad de errores.

[HU-036] - Integración de plataforma de monitoreo**Descripción:**

Como *administrador del sistema*,
quiero *integrar una plataforma de monitoreo*
para *visualizar y analizar en tiempo real el estado de la aplicación*.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que el sistema está en funcionamiento, Cuando ocurre una variación en el rendimiento, Entonces la plataforma debe reflejarlo en tiempo real.
- Dado que se produce un error, Cuando la métrica se dispara, Entonces debe estar visible desde la consola del sistema.

Notas Técnicas:

- Seleccionar herramienta según compatibilidad con infraestructura actual.
- Configurar agentes o exporters según la plataforma elegida.

Reglas de Negocio Asociadas:

- [ID-RN-063] Todas las métricas deben ser enviadas a una única plataforma de monitoreo aprobada.
- [ID-RN-064] El sistema debe estar disponible para el equipo de DevOps y soporte técnico.

Riesgos Asociados:

- [ID-RIESGO-050] Integración deficiente con los servicios actuales puede generar métricas incompletas o erróneas.

[HU-037] - Alertas automáticas**Descripción:**

Como *sistema*,
quiero *enviar alertas automáticas ante errores críticos o caídas*
para que *el equipo pueda intervenir rápidamente y minimizar impacto*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que ocurre un error crítico, Cuando se supera un umbral definido, Entonces debe enviarse una alerta automática por correo o canal interno.
- Dado que un servicio se cae, Cuando no responde en cierto tiempo, Entonces debe activarse una notificación inmediata.

Notas Técnicas:

- Integración con sistemas de mensajería como Slack, Microsoft Teams o email.
- Configuración de umbrales y tiempos de respuesta por tipo de alerta.

Reglas de Negocio Asociadas:

- [ID-RN-065] Todas las alertas deben activarse ante condiciones críticas predefinidas.
- [ID-RN-066] Las alertas deben contener información útil para facilitar la resolución.

Riesgos Asociados:

- [ID-RIESGO-051] Alertas mal configuradas pueden generar spam o ignorarse por fatiga de notificaciones.

[HU-038] - Dashboards de métricas**Descripción:**

*Como miembro del equipo técnico,
quiero visualizar dashboards con métricas clave del sistema
para tener visibilidad constante sobre el estado y el rendimiento.*

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que se accede al panel de control, Cuando el usuario entra, Entonces debe poder ver métricas como uptime, errores recientes y tiempos de respuesta.
- Dado que hay varias métricas, Cuando se configuren dashboards, Entonces deben segmentarse por módulos o servicios (API, base de datos, etc.).

Notas Técnicas:

- Compatible con Grafana o similar para visualización.
- Debe actualizarse en tiempo real o con intervalo de 1-5 minutos.

Reglas de Negocio Asociadas:

- [ID-RN-067] Debe haber al menos un dashboard con métricas globales del sistema.
- [ID-RN-068] Los dashboards deben ser accesibles por roles autorizados únicamente.

Riesgos Asociados:

- [ID-RIESGO-052] Métricas mal presentadas pueden generar falsas interpretaciones o acciones innecesarias.

[HU-039] - Documentación de políticas de logging y monitoreo**Descripción:**

Como miembro del equipo de desarrollo y mantenimiento, quiero tener documentación clara sobre qué y cómo loguear o configurar alertas para mantener coherencia, trazabilidad y facilidad de mantenimiento en todo el equipo.

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que un desarrollador trabaja con nuevos módulos, Cuando consulta las políticas de logging, Entonces debe entender qué nivel de log usar y cuándo.
- Dado que el sistema evoluciona, Cuando se actualizan las reglas, Entonces la documentación debe mantenerse sincronizada.

Notas Técnicas:

- Repositorio compartido (ej. Notion, Confluence, Markdown en Git).
- Debe incluir ejemplos de buenas prácticas y convenciones adoptadas.

Reglas de Negocio Asociadas:

- [ID-RN-069] Toda política de logging debe estar disponible y visible para el equipo técnico.
- [ID-RN-070] La documentación debe ser revisada y actualizada con cada sprint si hay cambios.

Riesgos Asociados:

- [ID-RIESGO-053] Falta de documentación clara puede generar inconsistencias y dificultar el debugging.

[HU-040] - Listado de géneros musicales

Descripción:

Como *administrador*,
quiero *visualizar un listado de todos los géneros musicales disponibles*
para *poder gestionarlos fácilmente desde una vista centralizada*.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que el administrador accede al panel de géneros, Cuando se carga la página, Entonces debe mostrarse un listado con todos los géneros registrados.
- Dado que existen muchos géneros, Cuando se visualizan, Entonces debe ser posible ordenarlos o filtrarlos por nombre.

Notas Técnicas:

- Implementar paginación o scroll infinito si hay muchos registros.
- Mostrar nombre y descripción del género en la tabla/listado.

Reglas de Negocio Asociadas:

- [ID-RN-071] Solo los administradores autenticados pueden acceder al listado completo.
- [ID-RN-072] Los géneros deben mostrarse en orden alfabético por defecto.

Riesgos Asociados:

- [ID-RIESGO-054] Listado lento o sin filtros podría dificultar la gestión si hay muchos géneros.

[HU-041] - Creación de nuevos géneros

Descripción:

Como *administrador*,
quiero *poder agregar nuevos géneros musicales con nombre y descripción*
para *mantener el catálogo actualizado y completo*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el administrador completa el formulario, Cuando envía los datos, Entonces debe crearse un nuevo género si el nombre no está duplicado.
- Dado que el formulario se valida, Cuando un campo está vacío o es inválido, Entonces debe mostrarse un mensaje de error adecuado.

Notas Técnicas:

- Validar unicidad del nombre y evitar campos vacíos.
- Se recomienda usar inputs controlados con feedback inmediato.

Reglas de Negocio Asociadas:

- [ID-RN-073] No se deben permitir nombres de género duplicados.
- [ID-RN-074] Todos los campos obligatorios deben ser validados antes de guardar.

Riesgos Asociados:

- [ID-RIESGO-055] La falta de validaciones puede generar datos inconsistentes o duplicados.

[HU-042] - Edición de géneros existentes

Descripción:

Como *administrador*,
quiero *modificar el nombre o descripción de un género existente*
para *mantener la información actualizada y coherente*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el administrador edita un género, Cuando guarda los cambios, Entonces debe actualizarse correctamente en el sistema.
- Dado que se modifica el nombre, Cuando el nuevo nombre ya existe, Entonces debe impedirse el guardado y mostrarse una advertencia.

Notas Técnicas:

- Reutilizar lógica de validación del formulario de creación.
- Implementar lógica para controlar cambios redundantes (sin modificar nada).

Reglas de Negocio Asociadas:

- [ID-RN-075] No se pueden duplicar nombres de géneros tras la edición.
- [ID-RN-076] Todos los cambios deben reflejarse inmediatamente en el sistema.

Riesgos Asociados:

- [ID-RIESGO-056] Ediciones erróneas pueden impactar en la selección de géneros en otras partes del sistema.

[HU-043] - Eliminación de géneros musicales**Descripción:**

Como *administrador*,
quiero *poder eliminar géneros musicales que no estén en uso*
para *mantener el sistema limpio y evitar redundancias innecesarias*.

Prioridad: Should have
Estimación: M

Criterios de Aceptación:

- Dado que el administrador intenta eliminar un género, Cuando el género no está en uso, Entonces debe eliminarse correctamente.
- Dado que el género está siendo utilizado, Cuando se intenta eliminar, Entonces debe bloquearse la acción con una advertencia.

Notas Técnicas:

- Mostrar mensaje de confirmación antes de eliminar.
- Incluir conteo de uso (n° de eventos/usuarios relacionados) antes de permitir la acción.

Reglas de Negocio Asociadas:

- [ID-RN-077] Solo se pueden eliminar géneros que no estén asociados a usuarios o eventos activos.
- [ID-RN-078] La eliminación debe ser lógica (soft delete) si es necesario mantener trazabilidad.

Riesgos Asociados:

- [ID-RIESGO-057] Eliminación accidental de géneros puede afectar formularios y datos históricos.

[HU-044] - Validación de uso antes de eliminar

Descripción:

Como *sistema*,
quiero *verificar si un género está en uso antes de eliminarlo*
para *evitar errores en formularios y mantener integridad en los datos*.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que el administrador inicia el proceso de eliminación, Cuando el sistema detecta uso activo, Entonces debe impedir la eliminación con una advertencia clara.
- Dado que el género no está en uso, Cuando se valida, Entonces debe autorizar la eliminación.

Notas Técnicas:

- Consultar referencias cruzadas (eventos, usuarios, preferencias).
- Mostrar al administrador un detalle opcional del uso (cantidad de registros asociados).

Reglas de Negocio Asociadas:

- [ID-RN-079] La validación de uso debe realizarse antes de confirmar cualquier eliminación.
- [ID-RN-080] Los géneros en uso no pueden ser eliminados bajo ninguna circunstancia.

Riesgos Asociados:

- [ID-RIESGO-058] Fallas en la validación pueden causar errores en cascada en la aplicación.

[HU-045] - Actualización en tiempo real de formularios de usuario

Descripción:

Como *usuario*,
quiero *ver actualizada la lista de géneros musicales en los formularios de eventos y preferencias*
para *poder seleccionar siempre entre las opciones más recientes*.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que se agrega, edita o elimina un género, Cuando un usuario accede a un formulario, Entonces la lista debe estar actualizada automáticamente.
- Dado que un género fue eliminado, Cuando ya no es válido, Entonces no debe mostrarse como opción en formularios.

Notas Técnicas:

- Considerar uso de WebSockets o actualización dinámica mediante llamadas a la API.
- Controlar caché o almacenamiento local para evitar datos obsoletos.

Reglas de Negocio Asociadas:

- [ID-RN-081] Los formularios deben reflejar siempre el estado actual de la base de géneros musicales.
- [ID-RN-082] Los géneros eliminados deben desaparecer de la lista sin afectar las selecciones válidas previas.

Riesgos Asociados:

- [ID-RIESGO-059] Formularios desactualizados pueden llevar a errores de validación o selección inválida.

[HU-046] - Listado de tipos de evento**Descripción:**

Como administrador,
quiero ver un listado con todos los tipos de evento existentes
para poder gestionarlos desde un panel centralizado.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que accedo al panel de tipos de evento, Cuando el sistema carga la sección, Entonces se debe mostrar un listado con nombre y descripción de cada tipo.

- Dado que hay muchos tipos registrados, Cuando visualizo el listado, Entonces debería poder filtrar y buscar por nombre.

Notas Técnicas:

- Usar paginación o scroll infinito si hay muchos elementos.
- Implementar filtros por texto y orden alfabético.

Reglas de Negocio Asociadas:

- [ID-RN-083] Todos los tipos de evento registrados deben ser visibles para los administradores.

Riesgos Asociados:

- [ID-RIESGO-060] El listado podría volverse poco manejable sin paginación ni filtros adecuados.

[HU-047] - Creación de nuevos tipos de evento

Descripción:

Como administrador,
quiero poder crear un nuevo tipo de evento
para que los organizadores lo seleccionen al configurar sus eventos.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que ingreso un nombre y descripción válidos, Cuando envío el formulario, Entonces el nuevo tipo debe guardarse correctamente.

- Dado que el nombre está duplicado o está vacío, Cuando intento guardar, Entonces debe mostrar un mensaje de error.

Notas Técnicas:

- Validación en frontend y backend para evitar duplicados y campos vacíos.
- Agregar mensaje de éxito o error tras guardar.

Reglas de Negocio Asociadas:

- [ID-RN-084] Los nombres de tipo de evento deben ser únicos y no pueden estar vacíos.

Riesgos Asociados:

- [ID-RIESGO-061] Malas validaciones podrían generar duplicidad o errores de carga.

[HU-048] - Edición de tipos de evento existentes**Descripción:**

Como administrador,
quiero poder editar el nombre o descripción de un tipo de evento
para corregir errores o mejorar la claridad de las categorías.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que selecciono un tipo existente, Cuando lo edito y guardo cambios válidos, Entonces se debe actualizar la información correctamente.
- Dado que el nuevo nombre está duplicado, Cuando intento guardar, Entonces debe mostrarse un mensaje de error.

Notas Técnicas:

- Controlar el impacto de los cambios en los eventos ya registrados.

- Mostrar campos prellenados con la información actual.

Reglas de Negocio Asociadas:

- [ID-RN-085] Los cambios deben reflejarse sin afectar eventos que ya usan ese tipo.

Riesgos Asociados:

- [ID-RIESGO-062] Cambios abruptos pueden generar confusión si no se sincronizan correctamente.

[HU-049] - Eliminación de tipos de evento**Descripción:**

Como administrador,
quiero poder eliminar un tipo de evento no vinculado
para mantener la base limpia y evitar confusión.

Prioridad: Should have

Estimación: M

Criterios de Aceptación:

- Dado que un tipo de evento no está asociado a eventos futuros o activos, Cuando intento eliminarlo, Entonces debe borrarse correctamente.
- Dado que un tipo está en uso, Cuando intento eliminarlo, Entonces debe aparecer un mensaje de bloqueo.

Notas Técnicas:

- Consultar la existencia de relaciones antes de permitir la eliminación.
- Mostrar advertencia de impacto antes de confirmar.

Reglas de Negocio Asociadas:

- [ID-RN-086] No se puede eliminar un tipo de evento vinculado a eventos futuros o activos.

Riesgos Asociados:

- [ID-RIESGO-063] Eliminaciones erróneas podrían generar referencias inválidas.

[HU-050] - Validación de uso antes de eliminar**Descripción:**

Como administrador,
quiero que el sistema valide si un tipo de evento está en uso
para evitar errores o inconsistencias al eliminar.

Prioridad: Must have

Estimación: S

Criterios de Aceptación:

- Dado que intento eliminar un tipo de evento, Cuando está en uso, Entonces el sistema debe bloquear la acción con un mensaje claro.
- Dado que no está en uso, Cuando lo elimino, Entonces debe borrarse de forma segura.

Notas Técnicas:

- Verificar asociaciones en base de datos antes de ejecutar el delete.
- Manejo de errores en backend y mensajes al usuario.

Reglas de Negocio Asociadas:

- [ID-RN-087] El sistema debe proteger los datos vinculados a eventos en curso o programados.

Riesgos Asociados:

- [ID-RIESGO-064] La eliminación sin validación puede romper relaciones en la base de datos.

[HU-051] - Actualización automática en formulario de creación de eventos**Descripción:**

Como organizador,
quiero que la lista de tipos de evento se actualice automáticamente en el formulario de creación
para poder elegir entre las opciones más recientes sin errores.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que se agrega, edita o elimina un tipo de evento, Cuando accedo al formulario de creación, Entonces la lista debe estar actualizada automáticamente.
- Dado que un tipo fue eliminado, Cuando ya no es válido, Entonces no debe mostrarse como opción.

Notas Técnicas:

- Considerar WebSockets o consultas a la API para mantener la lista actualizada.
- Controlar caché o almacenamiento local.

Reglas de Negocio Asociadas:

- [ID-RN-088] Los formularios de creación deben reflejar siempre el estado actual de los tipos de evento.
- [ID-RN-089] Los tipos eliminados no deben aparecer en formularios, pero deben respetarse los eventos que ya los tenían.

Riesgos Asociados:

- [ID-RIESGO-065] Formularios desactualizados podrían generar selecciones inválidas o errores en la carga.

[HU-052] - Tabla de referencia de estados de evento**Descripción:**

Como administrador, quiero contar con una tabla de referencia que contenga todos los posibles estados de evento para poder gestionar y asignar fácilmente el estado de los eventos.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que accedo a la tabla de estados, Cuando se visualiza, Entonces debe mostrar todos los estados posibles como "Borrador", "Activo", "Finalizado", "Cancelado", entre otros.
- Dado que un nuevo estado se agrega, Cuando se guarda, Entonces debe reflejarse en todas las opciones de eventos.

Notas Técnicas:

- La tabla debe ser fácilmente actualizable por el administrador.
- Los estados deben estar definidos claramente para evitar confusión.

Reglas de Negocio Asociadas:

- [ID-RN-090] Los estados de los eventos deben estar previamente definidos en una tabla de referencia.

Riesgos Asociados:

- [ID-RIESGO-066] Cambios incorrectos en la tabla pueden causar inconsistencias en los eventos.

[HU-053] - Asignación automática de estado según fechas o acciones**Descripción:**

Como sistema,
quiero asignar automáticamente el estado de un evento según reglas predefinidas
para facilitar la gestión sin intervención manual.

Prioridad: Must have

Estimación: L

Criterios de Aceptación:

- Dado que un evento alcanza la fecha de finalización, Cuando pasa ese día, Entonces el estado se debe cambiar automáticamente a "Finalizado".
- Dado que un evento es cancelado por el organizador, Cuando se realiza la acción, Entonces el estado se debe actualizar a "Cancelado".

Notas Técnicas:

- Definir las reglas de cambio de estado basadas en fechas y acciones (como cancelaciones o finalización de eventos).
- Implementar un servicio que ejecute estas actualizaciones automáticamente.

Reglas de Negocio Asociadas:

- [ID-RN-091] El sistema debe cambiar el estado de un evento según las reglas de fechas y acciones.

Riesgos Asociados:

- [ID-RIESGO-067] Si las reglas no se aplican correctamente, los eventos pueden estar en un estado incorrecto.

[HU-054] - Control de visibilidad según estado

Descripción:

Como sistema,
quiero ocultar eventos cancelados o finalizados del listado público
para asegurar que solo los eventos activos sean visibles para los usuarios.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un evento está en estado "Cancelado" o "Finalizado", Cuando un usuario accede al listado de eventos, Entonces estos eventos no deben ser visibles.
- Dado que un evento está en estado "Activo", Cuando se visualiza el listado, Entonces debe mostrarse normalmente.

Notas Técnicas:

- Implementar lógica en la API para controlar qué eventos deben ser visibles según su estado.
- Asegurarse de que los eventos no se muestren en el frontend si su estado es "Finalizado" o "Cancelado".

Reglas de Negocio Asociadas:

- [ID-RN-092] Los eventos con estado "Cancelado" o "Finalizado" no deben ser visibles para el público.

Riesgos Asociados:

- [ID-RIESGO-068] Los eventos incorrectamente ocultos pueden generar confusión o frustración entre los usuarios.

[HU-055] - Control de edición según estado

Descripción:

Como sistema,
quiero restringir la edición de eventos que estén en estado "Finalizado" o "Cancelado"
para evitar cambios innecesarios en eventos que ya no están activos.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un evento está en estado "Finalizado" o "Cancelado", Cuando un organizador o administrador intenta editarlo, Entonces debe aparecer un mensaje de advertencia que indique que no se puede editar.
- Dado que un evento está en estado "Borrador" o "Activo", Cuando se intenta editar, Entonces se debe permitir la modificación.

Notas Técnicas:

- Validar el estado del evento antes de permitir la edición.
- Mostrar mensajes claros sobre las restricciones de edición.

Reglas de Negocio Asociadas:

- [ID-RN-093] Los eventos "Finalizados" o "Cancelados" no deben ser editables.

Riesgos Asociados:

- [ID-RIESGO-069] Permitir la edición en estados inadecuados podría generar inconsistencias o cambios indeseados.

[HU-056] - Edición manual del estado por administradores

Descripción:

Como administrador,
quiero poder cambiar manualmente el estado de un evento
para corregir el estado si es necesario o por razones de gestión.

Prioridad: Should have

Estimación: S

Criterios de Aceptación:

- Dado que soy un administrador, Cuando accedo a un evento, Entonces debo poder cambiar su estado a "Borrador", "Activo", "Finalizado" o "Cancelado".
- Dado que cambio el estado, Cuando guardo la modificación, Entonces el nuevo estado debe reflejarse en todas las interfaces de usuario.

Notas Técnicas:

- Asegurarse de que la edición manual del estado solo sea posible para administradores.
- El cambio de estado debe reflejarse en la base de datos y en las interfaces relevantes.

Reglas de Negocio Asociadas:

- [ID-RN-094] Solo los administradores deben tener la capacidad de cambiar manualmente el estado de un evento.

Riesgos Asociados:

- [ID-RIESGO-070] Cambios manuales podrían generar inconsistencias si no se manejan correctamente.

[HU-057] - Visualización clara del estado en las interfaces de usuario

Descripción:

Como usuario o organizador,
quiero ver el estado de un evento de forma clara en las interfaces de usuario
para entender rápidamente si el evento está activo, finalizado o cancelado.

Prioridad: Must have

Estimación: M

Criterios de Aceptación:

- Dado que un evento está en cualquier estado, Cuando lo visualizo en el listado o en el detalle del evento, Entonces el estado debe ser claramente visible y destacado.
- Dado que un evento está "Finalizado" o "Cancelado", Cuando lo visualizo, Entonces debe estar indicado de manera destacada para que no haya confusión.

Notas Técnicas:

- Usar colores, íconos o etiquetas para hacer el estado fácilmente reconocible.
- Asegurarse de que el estado sea visible en todos los lugares donde se muestren eventos.

Reglas de Negocio Asociadas:

- [ID-RN-095] El estado de los eventos debe ser visible en todas las interfaces de usuario relevantes.

Riesgos Asociados:

- [ID-RIESGO-071] Una visualización poco clara del estado podría generar malentendidos o errores.

3.5 División de historias

Criterios considerados:

- Se dividieron las historias **Large** (L) en subhistorias más pequeñas, que representan entre **1 y 3 días de trabajo**.
- Se procuró que cada historia cumpla con los criterios **INVEST**: Independiente, Negociable, Valiosa, Estimable, Small, Testable

División de historias grandes:

[HU-002] - Sistema de recomendaciones de canciones por usuarios

Estimación original: L (Large) → Subdividida en 3 subhistorias

- **[HU-002a] Interfaz para sugerir canciones**
Como usuario, quiero acceder a un campo para sugerir canciones al ingresar a un evento, para participar en la playlist.
Estimación: S (Small)
Dependencia: ninguna
- **[HU-002b] Validación de género al sugerir**
Como usuario, quiero que solo pueda sugerir canciones del género permitido del evento, para asegurar que la playlist sea coherente con el estilo del evento.
Estimación: S (Small)
Dependencia: HU-002a
- **[HU-002c] Visualización de sugerencias en playlist**
Como usuario, quiero ver mi sugerencia reflejada en la playlist del evento, para confirmar que fue registrada correctamente.
Estimación: S (Small)
Dependencia: HU-002a

[HU-005] - Creación de eventos por parte de empresas organizadoras

Estimación original: L (Large) → Subdividida en 3 subhistorias

- **[HU-005a] Formulario de creación de evento**
Como empresa, quiero poder acceder a un formulario para poder ingresar los datos básicos del evento, como el nombre, la descripción, la fecha y la ubicación.
Estimación: M (Medium)
Dependencia: ninguna
- **[HU-005b] Definición de género musical del evento**
Como empresa, quiero definir el género musical al crear el evento, para que los

usuarios puedan sugerir canciones relevantes para ese género.

Estimación: S (Small)

Dependencia: HU-005a

- **[HU-005c] Confirmación y publicación del evento**

Como empresa, quiero recibir una confirmación visual al crear el evento, para asegurarse de que se haya publicado correctamente y que los usuarios pueden verlo y sugerir canciones.

Estimación: S (Small)

Dependencia: HU-005a

3.6 Subdivisión Técnica

Equipo Técnico (6 personas):

- Backend: 2 personas
- Frontend Web: 2 personas
- QA (Testing): 1 persona
- Documentación/Coordinación Técnica: 1 persona (apoya donde se necesite)

Subtareas por área:

Modelado de Datos (Backend)

- Definir entidades: Usuario, Evento, Empresa, Comentario, Preferencia
- Diseñar relaciones entre entidades
- Crear la base de datos inicial

Endpoints / API (Backend)

- POST /registro: registro de usuarios
- POST /login: inicio de sesión
- GET /eventos: listado de eventos
- GET /eventos/{id}: detalle de evento
- POST /eventos: subir nuevo evento (empresa)
- POST /eventos/{id}/comentarios: comentar evento
- POST /eventos/{id}/votar: votar preferencias
- GET /empresa/{id}/estadisticas: ver estadísticas

Interfaz UI/UX (Frontend)

- Diseño de pantallas: Inicio, Login, Registro, Listado de Eventos, Panel de Empresa
- Componente de eventos con botones de votar y comentar
- Diseño responsive
- Integración con API

Test Unitarios / Funcionales (QA)

- Validación de formularios
- Testeo de endpoints
- Flujo completo de usuario
- Automatización básica

Documentación Técnica

- Diagrama de base de datos
- Lista de endpoints con ejemplos
- Manual para correr el proyecto
- Guía de colaboración

3.7 Planificación de Capacidad Inicial

Velocidad Inicial Estimada

- Equipo de 6 personas.
- Estimación: 30 puntos por sprint.
- Duración del sprint: 1 semana.

Composición del Equipo

- 2 Backend
- 2 Frontend
- 1 QA Tester
- 1 Coordinador/Documentador

Planificación de los Primeros 3 Sprints

Sprint 1: Fundaciones (Semana 1)

- Historia 1: Registro/Login de usuarios
- Historia 2: Base de datos y relaciones
- Historia 3: Pantallas de login/registro
- Historia 4: Endpoint de autenticación
- Historia 5: Documentación inicial y pruebas

Objetivo: El usuario puede registrarse e iniciar sesión.

Sprint 2: Interacción del Usuario (Semana 2)

- Historia 1: Visualizar eventos
- Historia 2: Votar preferencias

- Historia 3: Comentar eventos
- Historia 4: Pruebas funcionales
- Historia 5: Documentación de flujo de usuario

Objetivo: El usuario puede votar y comentar en eventos.

Sprint 3: Panel Empresa y Estadísticas (Semana 3)

- Historia 1: Publicar evento
- Historia 2: Ver estadísticas del evento
- Historia 3: Diseño de panel de empresa
- Historia 4: Tests de funcionalidad completa
- Historia 5: Ajustes finales y documentación

Objetivo: Las empresas pueden publicar eventos y ver retroalimentación.

Métricas de Éxito por Sprint

- Sprint 1: Registro/login funcional sin errores
- Sprint 2: Eventos visibles, votación y comentarios funcionales
- Sprint 3: Publicación de eventos y visualización de estadísticas