HISTORIA DE USUARIO

Sigue la siguiente estructura y agrega tareas de acuerdo con el número de semanas de cada módulo.

Nombre de la HU:	**Administración de Eventos y Venues con Relaciones y Transacciones Optimizadas**
Objetivo de la HU	• Optimizar el acceso y la persistencia de datos dentro de la arquitectura hexagonal, aplicando relaciones avanzadas con JPA/Hibernate, control de transacciones y consultas eficientes mediante JPQL y Specifications. Además, incorporar migraciones versionadas para mantener la base de datos sincronizada entre entornos en el módulo de **Eventos y Venues**.

TASK 1

1. Relaciones y Ciclo de Vida de Entidades

2. Descripción de la Tarea:

3. Implementar relaciones **OneToMany**, **ManyToOne** y, de ser necesario, **ManyToMany** entre las entidades principales:
    1. Un **Venue** puede tener muchos **Eventos** → OneToMany.
    2. Un **Evento** pertenece a un solo **Venue** → ManyToOne.
    3. Opcional: un Evento puede tener múltiples categorías/etiquetas → ManyToMany.
4. Configurar correctamente la propiedad de la relación:
    1. mappedBy,
    2. cascade,
    3. orphanRemoval,
    4. columna foránea en Evento (venue_id).
5. Analizar y aplicar estrategias de carga **Lazy** y **Eager**, priorizando Lazy para evitar

sobrecarga cuando se cargan los Venues.

6. Revisar el **ciclo de vida de las entidades** (persist, merge, detach, remove) y su impacto en transacciones al crear, modificar o borrar un Evento asociado a un Venue.

TASK 2

1. Optimización de Consultas

2. Descripción de la Tarea:

3. Reemplazar consultas nativas o ineficientes por **JPQL** y **Specifications**, aplicadas al dominio de Eventos y Venues.
   Ejemplos:
   - Buscar eventos por venue
   - Buscar eventos por fecha o rango
   - Filtrar venues por capacidad o ubicación
4. Implementar filtros dinámicos para consultas:
   - filtro por estado del evento (activo, cancelado)
   - filtro por fecha de inicio/fin
   - filtro por venue específico
   - filtro por categoría si aplica
5. Detectar y reducir el problema de **N+1 queries** mediante:
   - @EntityGraph
   - join fetch
   - configuración de fetchType y batchSize.
6. Verificar la mejora del rendimiento mediante logs SQL y métricas del entorno comparando antes/después.

K 3

1. Transaccionalidad y Migraciones con Flyway

2. Descripción de la Tarea:

3. Aplicar la anotación **@Transactional** en los casos de uso de la capa de aplicación:
    - Diferenciar entre transacciones de lectura (readOnly = true) y escritura.
    - Analizar aislamiento y propagación dependiendo del flujo:
        - REQUIRED, REQUIRES_NEW, etc.
4. Implementar migraciones con **Flyway**:
    - Crear la carpeta resources/db/migration.
    - Incluir scripts SQL versionados:
        - V1__init.sql → creación de tablas Evento, Venue.
        - V2__relaciones.sql → relaciones, llaves foráneas e índices.
        - V3__ajustes.sql → ajustes adicionales que requiera el módulo.
    - Validar ejecución automática al iniciar la aplicación.
5. Probar replicabilidad en entornos limpios:
    - H2 en memoria
    - MySQL local (Docker o instalación local)

Criterios de Aceptación

- Relaciones OneToMany, ManyToOne y ManyToMany entre Eventos y Venues correctamente configuradas y funcionales.
- Se usa Lazy/Eager adecuadamente y no hay problemas N+1 en los listados de Eventos por Venue.
- Consultas implementadas con JPQL o Specifications, sin consultas nativas innecesarias.
- Correcta aplicación de @Transactional según el tipo de operación.
- Migraciones con Flyway versionadas y reproducibles en otros entornos.
- El dominio se mantiene limpio y desacoplado de JPA/Spring.
- El rendimiento de consultas mejora perceptiblemente (menos consultas, menos

carga de relaciones).

Story Points: 10

Cierre de la actividad

El proyecto debe demostrar manejo avanzado de persistencia, transacciones y optimización de consultas dentro del módulo de Eventos y Venues.
Las relaciones deben estar correctamente definidas y optimizadas, las consultas ajustadas y las migraciones deben garantizar consistencia entre entornos.

La entrega debe incluir:

- Código funcional
- Migraciones Flyway
- Documentación breve de los cambios
- Evidencia de mejora en el rendimiento de consultas y reducción de consultas N+1.