

## Entrega 2 - Proyecto Final

### Sistema de Reservas de Canchas

#### Alumno: Grebosz

Esta entrega incluye los avances del proyecto, incorporando mejoras y ampliaciones respecto a la primera entrega. Se detallan las vistas, funciones, procedimientos almacenados, triggers e instrucciones de importación, todo presentado con formato claro y estilizado para facilitar su lectura y evaluación.

Entrega 2 - Proyecto Final

Sistema de Reservas de Canchas

Alumno: Grebosz

## Introducción

### Objetivos:

- Evitar solapamientos
- Automatizar procesos
- Mejorar la experiencia de uso

### Situación Problemática:

### Modelo de Negocio:

Conecta jugadores con dueños de canchas, gestionando reservas y pagos.

Listado de Vistas - view\_reservation\_details: Objetivo: Reunir datos de reserva, jugador, cancha, dueño y pago para reportes. Tablas: reservation, user\_player, soccer\_field, user\_field, payment.

- view\_field\_availability: Objetivo: Consultar disponibilidad de canchas en un rango (@from / @to). Tablas: soccer\_field, reservation.

Verifica si existe solapamiento con reservas activas. Tablas: reservation. Objetivo: Reutilizable por procedimientos y queries.

validando solapamientos y opcionalmente registra pago. Tablas: reservation, payment.

Beneficio: Centraliza la lógica y mantiene transaccionalidad.

- sp\_seed\_more\_data(): Descripción: Genera datos de prueba adicionales (idempotente).

Tablas: user\_player, user\_field, soccer\_field. Beneficio: Facilita pruebas repetibles.

reservas solapadas. Tablas: reservation. Objetivo: Mantener integridad de horarios.

- trg\_payment\_after\_insert: Descripción: Confirma reserva automáticamente al registrar pago. Tablas: payment, reservation. Objetivo: Automatizar estado luego de pago.

procedures, triggers) - Entrega2\_insercion\_datos.sql (datos de prueba y llamadas a procedures)

'ReservasCanchas.sql' si aún no existe la base y tablas iniciales. 3. Ejecutar

'Entrega2\_creacion\_objetos.sql' para crear vistas, funciones, procedures y triggers. 4.

Establecer variables si desea probar la vista de disponibilidad: SET @from =

'2025-09-10 00:00:00'; SET @to = '2025-09-11 00:00:00'; Luego: SELECT \* FROM

view\_field\_availability; 5. Ejecutar 'Entrega2\_insercion\_datos.sql' para insertar datos de

prueba y probar sp\_create\_reservation. 6. Revisar que no existan errores; los procedures y triggers contienen señales/rollbacks en caso de conflicto.

Notas: - Todos los objetos fueron diseñados para ser idempotentes o para fallar de forma controlada ante violaciones (por ejemplo solapamientos). - El trigger original

'trg\_reservation\_no\_overlap' fue redefinido por claridad y para asegurar consistencia con sp\_create\_reservation.

## Notas Finales e Instrucciones

- Todos los objetos SQL fueron diseñados para ser idempotentes.
- Los triggers y procedimientos incluyen validaciones y rollback ante conflictos.

### Instrucciones de Importación:

1. Abrir cliente MySQL / Workbench.
2. Ejecutar 'ReservasCanchas.sql' para crear base y tablas.
3. Ejecutar 'Entrega2\_creacion\_objetos.sql' para crear vistas, funciones, procedures y triggers.
4. Establecer variables para probar disponibilidad:  

```
SET @from = '2025-09-10 00:00:00';  
SET @to = '2025-09-11 00:00:00';  
SELECT * FROM view_field_availability;
```
5. Ejecutar 'Entrega2\_insercion\_datos.sql' para insertar datos de prueba.
6. Verificar que no existan errores.