**Reflexión Individual – Adrián González**

1. ¿Qué criterios usaron para decidir qué entidades y relaciones debían formar parte del modelo?

El modelo de base de datos se diseñó considerando los requisitos fundamentales de un sistema de reservas deportivas. Las principales entidades fueron:

* Entidades centrales: Usuario, Cancha y Reserva (core del negocio)
* Entidades de soporte: Instalación, Deporte, Horario, Servicio Adicional (datos maestros)
* Entidades transaccionales: Pago, Valoración (operaciones del sistema)

Las relaciones se establecieron basándose en:

* Relaciones 1:N (una instalación tiene muchas canchas)
* Relaciones N:M (reservas con servicios adicionales) resueltas con tablas puente
* Jerarquías (ciudad → instalación → cancha)

Cabe mencionar que no se modeló un sistema complejo de roles de usuarios (solo usuarios básicos). Los horarios se simplificaron a días de semana y horas, sin considerar temporadas y no se incluyó gestión de equipos o torneos, aunque podría ser relevante para un sistema deportivo

1. ¿Qué tan adecuadas fueron las claves primarias y foráneas que definieron en su diseño?

Claves primarias:

* Se utilizó consistentemente IDs autoincrementales (SERIAL/IDENTITY) como PKs
* Ventajas: Simplicidad, consistencia, independencia del contenido de los datos
* Limitación: No son significativas por sí mismas (requieren joins para interpretación)

Claves foráneas:

* Correctamente definidas para mantener integridad referencial
* Se usó ON DELETE CASCADE en relaciones fuertes (ej: usuario → teléfono)
* Se usó ON DELETE RESTRICT donde era necesario preservar integridad (ej: reserva → cancha)

Podría considerarse claves naturales compuestas en algunos casos (ejemplo: nombre de ciudad + instalación)

1. ¿En qué medida aplicaron la normalización? ¿Qué beneficios y limitaciones experimentaron?

1FN:

* Todas las tablas cumplen con 1FN (valores atómicos, sin grupos repetidos)

Ejemplo: Teléfonos de usuario en tabla separada (usuario\_telefono)

2FN:

* Cumplimiento completo al eliminar dependencias parciales
* Todos los atributos no clave dependen completamente de la PK

3FN:

* Mayoría de tablas cumplen 3FN al eliminar dependencias transitivas.
* Excepción potencial: La tabla 'reserva' tiene 'duración' que podría calcularse (pero se justifica por el trigger que la mantiene)

Se aceptó redundancia controlada en 'duración' para mejorar rendimiento en consultas. También se crearon tablas puente (cancha\_horario, reserva\_servicio) para relaciones N:M. Además, no se normalizó más allá de 3FN para mantener simplicidad

1. ¿Qué restricciones y reglas del negocio implementaron directamente en la base de datos y por qué?

CHECK constraints:

* Validación de rangos (puntuación 1-5 en valoraciones)
* Validación de horas (hora\_fin > hora\_inicio)
* Precios no negativos

NOT NULL:

* Campos obligatorios como nombre de usuario, email, etc.
* Previene datos incompletos

UNIQUE:

* Para evitar duplicados (email de usuario, combinación cancha-horario)
* Garantiza unicidad donde el negocio lo requiere

DEFAULT values:

* Estados por defecto ('activo' para usuarios, 'pendiente' para reservas)
* Fechas automáticas (fecha\_registro, fecha\_creacion)

Triggers:

Estos ayudan al cálculo automático de duración de reservas, prevención de solapamientos de reservas (regla de negocio crítica) y el cálculo automático del total de pagos.

1. ¿Qué ventajas o desventajas identificas del modelo que construyeron al momento de hacer consultas complejas?

Ventajas:

* Estructura clara facilita joins entre tablas relacionadas
* Normalización reduce redundancia y facilita actualizaciones
* Índices implícitos en PKs/FKs mejoran rendimiento de joins
* Datos consistentes permiten consultas agregadas confiables

Desventajas:

* Algunas consultas pueden requerir múltiples joins (ej: reportes que cruzan usuario-cancha-servicios)
* Falta de índices adicionales podría afectar rendimiento con mucho datos
* La tabla de horarios podría ser compleja para consultas de disponibilidad

1. ¿Qué cambiarían en el diseño de la base de datos si tuvieran que escalar este sistema a un entorno de producción?

La escalabilidad, se podrían implementar índices adicionales en campos de filtro frecuente (fecha\_reserva, estado), con eso se llega a considerar particionamiento por fechas para la tabla de reservas.

El rendimiento se puede mejorar al añadir índices a campos de búsqueda (nombre de cancha, email de usuario) y optimizar consultas frecuentes con vistas materializadas.

También se puede mantener la integridad Implementar backups y replicación al añadir triggers para auditoría de cambios críticos y considerar transacciones explícitas en operaciones complejas

Ni tan mal, se podría decir que el modelo actual es sólido para el alcance definido, pero requeriría adaptaciones para escalar a un entorno de producción con mayores volúmenes de datos y usuarios concurrentes. Las principales mejoras estarían en índices, particionamiento y gestión de concurrencia.