# FUL5 Entrega N°2 Escaray Maite

#### Descripción de la temática de la base de datos

Para este proyecto se buscó diseñar e implementar una base de datos para la gestión de reservas de canchas de fútbol 5. La base de datos busca digitalizar y optimizar la administración y gestión de turnos, clientes y pagos en distintos complejos deportivos que ofrecen este servicio.

#### Introducción

He seleccionado este proyecto ya que en Argentina, el fútbol 5 es una actividad diaria entre grupos de amigos/as que actualmente se están expandiendo cada vez más. Semanalmente, grupos de amigos/as reservan canchas por hora en complejos que están por los distintos barrios de las ciudades. Hoy en día, muchos de estos complejos, para no decir todos, utilizan medios informales como whatsapp o llamados para gestionar las reservas y anotan estos datos en papel o en alguna planilla básica de excel.

#### Objetivo

El objetivo principal y central es diseñar y construir una base de datos relacional que permita:

- Registrar las distintas canchas y sus características.
- Gestionar las reservas por parte de clientes.
- Asociar pagos y medios de pago.
- Tener informes contables y analíticos.
- Brindar soporte para la logística operativa de los complejos deportivos.
- El diseño incluirá distintas sedes de el complejo deportivo que llamaremos a modo de ejemplo: FUL5.

# Situación problemática

Bajo la modalidad informal con la que operan los complejos de futbol 5, esto genera distintas problemáticas como:

- Superposición de turnos.
- Ausencia y bajas en la reserva.

- Pérdida de información de los clientes para comunicarse en caso de que por ejemplo se olviden algo, se deba modificar su reserva, haga falta el pago, etc.
- Dificultad para hacer seguimiento de las señas y los pagos completos.
- Dificultad para hacer seguimiento a tipos de clientes, por ejemplo, un cliente que suele reservar y no aparecer luego o cancelar las reservas a último momento, debe registrarse y ser tenido en cuenta para futuras reservas.

#### Modelo de negocio

# Link para visualizar el cursograma: Cursograma FUL5.drawio (1).pdf

FUL5 es un complejo deportivo (ficticio) que ofrece múltiples canchas (techadas y al aire libre) distribuidas en 4 sedes dentro de la ciudad de La Plata. Los clientes suelen reservar por la duración de 1 hora, seleccionando día, hora, cancha y medio de pago. El complejo gestiona las reservas y el control de turnos desde una interfaz conectada a la base de datos.

# Cada reserva incluye:

- a) El cliente que realiza la reserva.
- b) La sede y la cancha seleccionada.
- c) La fecha y hora.
- d) El monto a abonar.
- e) El medio de pago utilizado (efectivo, transferencia, billeteras virtuales, etc.).
- f) Por último, el estado de la reserva (en espera, finalizada, ausente, cancelada, etc).

Cada sede tiene personal responsable de la gestión operativa, pero la información se centraliza en una única base de datos. Esto permite tener control sobre todas las sedes, detectar superposición de horarios, generar reportes contables, y planificar horarios pico.

#### Se debe contemplar:

- Un administrador por sede.
- Clientes frecuentes o clientes esporádicos.
- Canchas con distintas características en distintas sedes.
- Diversos medios de pago (efectivo, transferencia, apps).

# Diagramas de entidad relación.

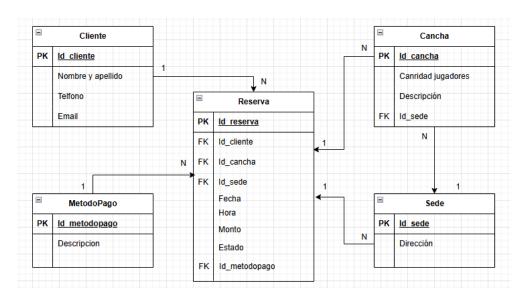
#### Link para visualizar el DER: DER FUL5.drawio.pdf

La estructura está compuesta por cinco entidades:

- 1) Cliente: esta entidad contiene la información básica de las personas que realizan la reserva.
- 2) Reserva: es la entidad central del modelo que registra cada alquiler realizado por un cliente.
- **3)** Cancha: contiene cada cancha disponible en las distintas sedes para alquilar, con sus especificaciones.
- 4) Sede: representa cada una de las ubicaciones físicas del complejo.
- 5) Medio pago: los distintos métodos de pago que puede elegir un cliente.

Cada entidad se relaciona con otra a través de claves primarias y foráneas. Las relaciones entre las entidades son las siguientes:

- Cliente Reserva: relacion 1 a N (un cliente puede hacer muchas reservas)
- Cancha Reserva: relación 1 a N (una cancha puede ser reservada muchas veces pero, cada reserva tiene una sola cancha asignada).
- Sede Cancha: relación 1 a N (una sede tiene muchas canchas).
- Sede Reserva: relación 1 a N (una sede puede tener muchas reservas).
- Método Pago Reserva: 1 a N (un método de pago puede ser asociado a muchas reservas).



# Listado de las tablas que comprenden la base de datos

CLIENTES				
Campo	Descripción	Nombre completo	Tipo de dato	Tipo de clave
Id_cliente	Número único para identificar a cada cliente	ld del cliente	INT AUTO_INCREMENT NOT NULL	PK
Nombre	Nombre cliente	Nombre	VARCHAR(100) NOT NULL	
Apellido	Apellido cliente	Nombre	VARCHAR(100) NOT NULL	
Teléfono	teléfono de contacto del cliente	Teléfono	VARCHAR(20) NOT NULL	
Email	email de contacto del cliente	Email	VARCHAR(100)	UNIQUE INDEX

SEDE				
Campo	Descripción	Nombre completo	Tipo de dato	Tipo de clave
ld_sede	Número único para identificar cada sede	ld de la sede	INT	PK
Nombre	Identificador de la sede	Nombre de la sede	VARCHAR(100) NOT NULL	
Dirección	Dirección física de cada sede	Dirección física	VARCHAR(150)	

CANCHA				
Campo	Descripción	Nombre completo	Tipo de dato	Tipo de clave
ld_cancha	Número único para identificar cada cancha	Id de la cancha	INT AUTO_INCREMENT	PK
Cantidad jugadores	Cada cancha tendrá una cantidad de jugadores	Cantidad de jugadores por cancha	INT	
Descripción	Especificaciones de cada cancha como por ejemplo, si es techada o al aire libre	Descripción específica de cada cancha	VARCHAR(200)	
ld_sede	Para identificar en que sede se encuentra cada cancha	ld de la sede	INT AUTO_INCREMENT	FK

MÉTODO DE PAGO				
Campo	Descripción	Nombre completo	Tipo de dato	Tipo de clave
ld_metodop ago	Número único para identificar cada medio de pago	Id del método de pago	INT NOT NULL PRIMARY KEY AUTO_INCREMENT	PK
Descripción	datos del método de pago como CBU, alias, etc.	Descripcion del metodo de pago	VARCHAR(500)	

RESERVA				
Campo	Descripción	Nombre completo	Tipo de dato	Tipo de clave
ld_reserva	Número único para identificar cada reserva	Id de la reserva	INT AUTO_INCREMENT NOT NULL	PK
ld_cliente	Número único para identificar cada cliente	Id del cliente	INT AUTO_INCREMENT NOT NULL	FK
ld_cancha	Número único para identificar cada cancha	Id de la cancha	INT AUTO_INCREMENT NOT NULL	FK
ld_sede	Número único para identificar cada sede	ld de la sede	INT AUTO_INCREMENT	FK
Fecha	Fecha de alquiler de la cancha	Fecha de alquiler	DATE	
Hora	Hora de inicio de uso de la cancha	Hora de alquiler	TIME	
Monto	Monto total adeudado (si pagan seña, este monto se actualiza y disminuye)	Monto pagado	DECIMAL(10,2)	
Estado	En espera, completada, ausente, cancelada	Estado de la reserva	VARCHAR(40)	
ld_metodop ago	Número único para identificar cada método de pago	ld del método de pago	INT AUTO_INCREMENT NOT NULL	FK

#### Listado de Vistas

#### 1) Vista reservas detalle

Esta vista muestra el detalle completo de cada reserva que se sube al sistema. Su objetivo es facilitar el alcance y la visualización de la información de una reserva sin tener que realizar múltiples JOINs en las tablas. Esta vista no modifica datos, solo consulta.

- Tablas involucradas:
  - Reserva
  - Cliente
  - Sede
  - o Cancha
  - MetodoPago
- Campos de la vista:
  - o Id\_reserva
  - Cliente
  - Sede
  - o Cancha
  - Metodo\_pago
  - o Fecha
  - o Hora
  - Estado

#### 2) Vista Clientes reservas

Esta vista hace un listado de todos los clientes y la cantidad de reservas realizadas por cada uno. Su objetivo es poder extraer datos útiles para analizar y realizar informes como, por ejemplo, evaluar la participación y frecuencia de reservas por cliente. Estos análisis pueden servir para promociones, fidelizaciones o incluso para

tomar medidas restrictivas con clientes que suelen cancelar o ausentarse a las reservas.

- Tablas involucradas:
  - Cliente
  - Reserva
- Campos de la vista:
  - Id\_cliente
  - Cliente
  - Cantidad\_reservas

# 3) Vista reservas mensuales

Esta vista nos proporciona un resumen mensual de todas las reservas realizadas (no incluye las reservas canceladas u ausentes), mostrando además el total facturado para cada mes, por ende, nos brinda información valiosa a nivel contable. El objetivo es facilitar el análisis contable y operativo por mes, obteniendo estadísticas e informes sobre la cantidad de reservas y la facturación sin tener que hacer consultas complejas.

- Tablas involucradas:
  - Reserva
- Campos de la vista:
  - Mes: formato YYYY/MM
  - Cantidad\_reservas: cantidad total de reservas en el mes
  - Total\_facturado: suma de los montos cobrados en cada reserva del mes.

#### Listado de Funciones

# 1) Total gastado por cliente

Esta función nos proporciona el monto total de dinero que fue gastado por un cliente en reservas, al día de la fecha en que se ejecuta la función. Tiene por

objetivo brindar un KPI u métrica económica por cliente para tomar decisiones en cuanto a promociones o segmentación.

- Tablas involucradas:
  - Reserva
- Datos que utiliza:
  - Id cliente
  - o Monto de la reserva

# 2) Esta cancha está ocupada

A través de esta función podemos verificar si una cancha ya está asignada a una reserva en una fecha y hora específica. El objetivo es evitar superposición de turnos y conflictos al momento de registrar una nueva reserva en el sistema. Esta función es utilizada como una herramienta interna, es decir, que no es llamada por el usuario cuando necesite sino que, cuando realizar el AgregarReserva (stored Procedure detallado más adelante), antes de insertar una nueva reserva, el sistema ejecuta esto y lanza un error si la cancha ya está ocupada.

- Tablas involucradas:
  - Reserva
- Datos que utiliza:
  - Cancha id
  - Fecha
  - o Hora

#### 3) Es cliente frecuente

Esta función determina y asigna si un cliente es considerado frecuente teniendo en cuenta la cantidad de reservas confirmadas que haya realizado. El objetivo es identificar clientes habituales para posibles beneficios o tratos preferenciales para futuras reservas. El criterio definido es que si un cliente tiene 3 o más reservas confirmadas, se lo considera cliente frecuente.

- Tablas involucradas:
  - Reserva
- Datos que utiliza:
  - Id\_cliente
  - Estado
- Retorna:
  - TRUE si el cliente es frecuente
  - o FALSE si no lo es

#### Listado de Stored Procedures

#### 1) Agregar reserva

Agrega una nueva reserva, verificando previamente si la cancha está libre. El objetivo es automatizar la lógica de registro de reservas, validando la disponibilidad para evitar superposición y duplicación de turnos. A través de este Stored Procedure se logra mejorar la integridad del sistema.

- Tablas involucradas:
  - Reserva (inserta datos)
- Datos que utiliza:
  - Utiliza la función EstaCanchaOcupada
- Retorna:
  - "La cancha ya está reservada en ese horario" si ya está ocupada.

# 2) Historial de reservas por cliente

Nos brinda todas las reservas hechas por un cliente específico. Tiene como objetivo acceder de forma rápida y confiable al historial de un cliente.

- Datos y tablas que utiliza:
  - Utiliza la vista Vista\_reservas\_detalle que, a su vez, accede a múltiples tablas (detalladas anteriormente)
  - Utiliza como parámetro el Id\_cliente

#### 3) Cancelar reservas pasadas

Actualiza de forma automática el estado de las reservas vencidas, es decir, aquellas que quedaron "en espera" y no fueron concretadas, a "Cancelada" cuando la fecha de reserva ya pasó. Tiene como objetivo mantener el sistema actualizado para evitar que queden reservas pendientes en el sistema.

- Tablas involucradas:
  - Reserva (modifica el campo Estado)
- Datos que utiliza:
  - Estado
  - o Fecha
- Condición aplicada:
  - Si la fecha de la reserva es menor al día actual y su estado es "En espera" se cambia al estado "Cancelada".

# Listado de Triggers

#### 1) before\_insert\_reserva

Este trigger se ejecuta automáticamente antes de insertar una reserva en la tabla Reserva. Tiene como objetivo evitar registros incompletos. Si el campo estado viene vacío al momento de crear una reserva, el trigger lo completa automáticamente con el valor por defecto 'En espera'.

- Tablas involucradas:
  - Reserva (modifica el campo Estado)
- Situación en la que se acciona:
  - Cada vez que se realiza un INSERT en la tabla reserva.

# 2) after\_delete\_reserva

Este trigger se ejecuta después de eliminar una reserva en la tabla Reserva. Tiene como objetivo informar al usuario del sistema que una reserva fue eliminada.

Para eso, se emite un mensaje utilizando un Signal con un mensaje dinámico que indica el Id\_reserva eliminado. De esta manera, permite confirmar en tiempo real que se eliminó una reserva aportando un sistema de alerta simple sin necesidad de crear tablas adicionales.

- Tablas involucradas:
  - Reserva
- Situación en la que se acciona:
  - o Cada vez que se ejecuta un DELETE en la tabla reserva.