Sistema Gestor de Grand Slam Tennis - Documentación Técnica

Bibliotecas Utilizadas

El sistema utiliza las siguientes bibliotecas principales:

iostream

Proporciona funcionalidades básicas de entrada y salida estándar en C++. Se utiliza para la interacción con el usuario a través de la consola, permitiendo mostrar información (`cout`) y recibir datos (`cin`).

pqxx/pqxx

Biblioteca oficial de C++ para PostgreSQL. Ofrece una interfaz orientada a objetos para interactuar con bases de datos PostgreSQL, permitiendo realizar conexiones, ejecutar consultas y manejar transacciones de manera segura y eficiente.

cstdlib

Proporciona funciones generales de utilidad, incluyendo gestión de procesos y sistema. En este proyecto se utiliza principalmente para la función `system()` que permite ejecutar comandos del sistema operativo.

Funciones de Utilidad

Funciones de Interfaz

limpiarPantalla()

void limpiarPantalla() {

system("clear");

}

Limpia la pantalla de la consola para mantener una interfaz limpia y organizada.

limpiarEntrada()

void limpiarEntrada() {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

Limpia el buffer de entrada para evitar problemas en la lectura secuencial de datos.

esperarEntrada()

void esperarEntrada() {

cout << "\nPresione Enter para continuar...";

cin.get();

}

Pausa la ejecución del programa hasta que el usuario presione Enter.

Funciones de Base de Datos

Conexión y Configuración

conectar()

Establece la conexión con la base de datos PostgreSQL con los siguientes aspectos:

- Configura parámetros de conexión (base de datos, usuario, contraseña, host, puerto)

- Verifica la conexión exitosa

- Maneja errores de conexión

comprobarTablas()

Inicializa la estructura de la base de datos creando las siguientes tablas principales:

1. Countries: Países

2. Venues: Sedes de torneos

3. Categories: Categorías de competición

4. Players: Información de jugadores

5. Referees: Árbitros

6. Tournaments: Torneos

7. TournamentEditions: Ediciones específicas de torneos

8. Matches: Partidos individuales

9. MatchParticipants: Participantes en partidos

Funciones Auxiliares de Base de Datos

ejecutarSQL()

bool ejecutarSQL(connection &c, const string &query) {

try {

work W(c);

W.exec(query);

W.commit();

return true;

} catch (const exception &e) {

cerr << "Error en la consulta SQL: " << e.what() << endl;

return false;

}

}

Wrapper para ejecutar consultas SQL de manera segura, manejando transacciones y excepciones.

Funciones de Consulta

Tipos de Consultas Implementadas

1. consultarTorneos(): Muestra información histórica de torneos

2. consultarJugadores(): Directorio de jugadores con estadísticas

3. consultarArbitros(): Información de árbitros

4. consultarPartidosPorTorneo(): Detalle de partidos en un torneo específico

5. consultarHistorialEntrenadores(): Historial de entrenadores de jugadores

6. consultarGananciasJugador(): Ganancias y desempeño de un jugador

Funciones de Registro

Tipos de Registros

1. registrarPartido(): Registro detallado de partidos con:

- Validación de categorías

- Selección de jugadores

- Registro de resultados

- Manejo de premios

2. registrarJugador(): Registro de nuevos jugadores con:

- Validación de datos personales

- Registro de nacionalidades

- Verificación de unicidad

3. registrarTorneo(): Registro de nuevas ediciones de torneos

Sistema de Menú

menuPrincipal()

Implementa la interfaz principal del sistema con:

- Menú organizado por categorías

- Opciones de consulta y registro

- Validación robusta de entradas

- Manejo de errores

Consideraciones de Diseño

1. Estructura Modular

- Funciones específicas para cada tarea

- Separación clara de responsabilidades

- Código reutilizable y mantenible

2. Manejo de Errores

- Uso consistente de try-catch

- Mensajes de error descriptivos

- Recuperación graciosa ante fallos

3. Interfaz de Usuario

- Menús claros y organizados

- Validación robusta de entrada

- Feedback visual con colores y emojis

4. Seguridad de Datos

- Uso de transacciones para operaciones críticas

- Validación de datos de entrada

- Protección contra pérdida de datos

5. Usabilidad

- Interfaz intuitiva

- Mensajes claros

- Confirmaciones en operaciones importantes

Funcionalidades Destacadas

- Gestión completa de torneos de Grand Slam

- Registro detallado de partidos, jugadores y árbitros

- Consultas avanzadas con múltiples criterios

- Seguimiento de ganancias y desempeño

- Sistema de categorías flexible

Voy a analizar en detalle las consultas SQL y las validaciones empleadas en cada función de consulta del sistema Grand Slam Tennis:

Consultas SQL Detalladas y Validaciones

1. consultarTorneos()

Consulta SQL Principal:

SELECT te.year, t.name as tournament, v.name as venue,

c.name as country, te.start\_date, te.end\_date,

COUNT(DISTINCT m.match\_id) as total\_matches

FROM TournamentEditions te

JOIN Tournaments t ON te.tournament\_id = t.tournament\_id

JOIN Venues v ON te.venue\_id = v.venue\_id

JOIN Countries c ON t.country\_id = c.country\_id

LEFT JOIN Matches m ON te.edition\_id = m.edition\_id

GROUP BY te.year, t.name, v.name, c.name, te.start\_date, te.end\_date

ORDER BY te.year DESC, t.name;

Validaciones y Características:

- Uso de LEFT JOIN para incluir torneos sin partidos

- Conteo de partidos usando COUNT(DISTINCT)

- Ordenamiento por año descendente y nombre de torneo

- Manejo de NULL con COALESCE en caso de datos faltantes

2. consultarJugadores()

Consulta SQL Principal:

SELECT p.player\_id, p.name, p.birth\_date,

COALESCE(string\_agg(DISTINCT c.name, ', '), 'Sin nacionalidad') as nationalities,

COUNT(DISTINCT mp.match\_id) as matches\_played,

COUNT(DISTINCT CASE WHEN mp.is\_winner THEN mp.match\_id END) as matches\_won

FROM Players p

LEFT JOIN PlayerNationalities pn ON p.player\_id = pn.player\_id

LEFT JOIN Countries c ON pn.country\_id = c.country\_id

LEFT JOIN MatchParticipants mp ON p.player\_id = mp.player\_id

GROUP BY p.player\_id, p.name, p.birth\_date

ORDER BY p.name;

Validaciones y Características:

- Manejo de múltiples nacionalidades con string\_agg()

- LEFT JOIN para incluir jugadores sin partidos o nacionalidades

- Conteo de partidos jugados y ganados

- Uso de CASE WHEN para contar victorias

3. consultarArbitrosPorTorneo()

Consulta SQL Principal (fragmento):

WITH ArbitroStats AS (

SELECT

r.referee\_id,

r.name as referee\_name,

c.name as country,

COUNT(DISTINCT m.match\_id) as total\_matches,

COUNT(DISTINCT cat.category\_id) as categories\_count,

string\_agg(DISTINCT cat.name, ', ') as categories,

COUNT(DISTINCT CASE WHEN m.round = 'Final' THEN m.match\_id END) as finals\_count,

MIN(m.date) as first\_match,

MAX(m.date) as last\_match

FROM Referees r

JOIN Countries c ON r.country\_id = c.country\_id

JOIN Matches m ON r.referee\_id = m.referee\_id

JOIN Categories cat ON m.category\_id = cat.category\_id

JOIN TournamentEditions te ON m.edition\_id = te.edition\_id

JOIN Tournaments t ON te.tournament\_id = t.tournament\_id

WHERE t.tournament\_id = $1 AND te.year = $2

GROUP BY r.referee\_id, r.name, c.name

ORDER BY total\_matches DESC, referee\_name

)

Validaciones y Características:

- Uso de CTE (Common Table Expression) para cálculos complejos

- Múltiples agregaciones:

- Conteo de partidos

- Categorías arbitradas

- Partidos de finales

- Rango de fechas de arbitraje

- Filtrado por torneo y año específicos

- Ordenamiento por número de partidos

4. consultarHistorialEntrenadores()

Consulta SQL Principal (fragmento):

WITH CoachStats AS (

SELECT

c.coach\_id,

c.name as coach\_name,

co.name as country,

ch.start\_date,

ch.end\_date,

COUNT(DISTINCT m.match\_id) as matches\_during\_period,

COUNT(DISTINCT CASE WHEN mp.is\_winner THEN m.match\_id END) as matches\_won,

SUM(CASE WHEN mp.is\_winner THEN m.winner\_prize ELSE m.loser\_prize END) as total\_earnings,

string\_agg(DISTINCT cat.name, ', ') as categories

FROM Coaches c

JOIN Countries co ON c.country\_id = co.country\_id

JOIN CoachingHistory ch ON c.coach\_id = ch.coach\_id

LEFT JOIN MatchParticipants mp ON ch.player\_id = mp.player\_id

LEFT JOIN Matches m ON mp.match\_id = m.match\_id

AND m.date BETWEEN ch.start\_date AND COALESCE(ch.end\_date, CURRENT\_DATE)

LEFT JOIN Categories cat ON m.category\_id = cat.category\_id

WHERE ch.player\_id = $3

GROUP BY c.coach\_id, c.name, co.name, ch.start\_date, ch.end\_date

)

Validaciones y Características:

- CTE con cálculos estadísticos complejos

- Cálculo de:

- Partidos durante período de entrenamiento

- Victorias

- Ganancias totales

- Uso de COALESCE para manejar períodos de entrenamiento activos

- Filtrado de partidos dentro del período de entrenamiento

5. consultarGananciasJugador()

Consulta SQL Principal (fragmento):

WITH PlayerEarnings AS (

SELECT

m.match\_id,

cat.name as category,

m.round,

m.date,

CASE WHEN mp.is\_winner THEN m.winner\_prize ELSE m.loser\_prize END as prize,

mp.is\_winner,

string\_agg(DISTINCT CASE

WHEN mp2.team\_number = mp.team\_number THEN p2.name

END, ' / ') as teammate,

string\_agg(DISTINCT CASE

WHEN mp2.team\_number != mp.team\_number THEN p2.name

END, ' / ') as opponents,

mp.score

FROM Matches m

JOIN Categories cat ON m.category\_id = cat.category\_id

JOIN TournamentEditions te ON m.edition\_id = te.edition\_id

JOIN MatchParticipants mp ON m.match\_id = mp.match\_id

JOIN MatchParticipants mp2 ON m.match\_id = mp2.match\_id

JOIN Players p2 ON mp2.player\_id = p2.player\_id

WHERE te.tournament\_id = $1

AND te.year = $2

AND mp.player\_id = $3

AND p2.player\_id != $3

GROUP BY m.match\_id, cat.name, m.round, m.date, mp.is\_winner, mp.score,

m.winner\_prize, m.loser\_prize

)

Validaciones y Características:

- CTE para cálculo detallado de ganancias

- Manejo de equipos en diferentes categorías

- Cálculo dinámico de premios según resultado

- Agregación de información de compañeros y oponentes

Validaciones Generales Empleadas

1. Validaciones de Entrada

- Verificación de rangos de IDs

- Comprobación de existencia de registros

- Validación de formatos de fecha

2. Validaciones de Integridad

- Uso de JOIN para asegurar relaciones entre tablas

- Manejo de NULL y valores por defecto

- Uso de COALESCE para manejar casos de datos incompletos

3. Validaciones de Consistencia

- Comprobación de coherencia en fechas de torneos

- Validación de género en categorías

- Verificación de disponibilidad de jugadores

Técnicas SQL Avanzadas

- Uso de Common Table Expressions (CTE)

- Subconsultas correlacionadas

- Agregaciones complejas

- Funciones de ventana

- Joins múltiples

- Manejo de agregaciones con string\_agg()