

# Trabajo Práctico Especial

Instituto Tecnológico de Buenos Aires - Base de Datos I (72.37)

Grupo 19

**Nicolás Guillermo Arancibia Carabajal**

narancibiacarabajal@itba.edu.ar

64.481

**Nicole Yael Salama**

nsalama@itba.edu.ar

64.488

**Juan Pablo Birsa**

jbirsa@itba.edu.ar

64.500

**Augusto Barthelemy Solá**

abarthelemysola@itba.edu.ar

64.502

22 de noviembre de 2024

## 1 Objetivo del trabajo

El objetivo principal de este Trabajo Práctico Especial consiste en la investigación y posterior implementación de herramientas de SQL Avanzado, como **PSM** y **Triggers**, ya que estas ofrecen funcionalidades que no pueden ser abordadas de forma estándar (como el uso de claves primarias o foráneas).

En particular, estas herramientas avanzadas nos permitieron preservar una dependencia funcional que se había perdido durante la descomposición de un esquema para cumplir con la **Forma Normal de Boyce-Codd** (BCNF).

## 2 Roles del equipo

Las tareas dentro del Trabajo Práctico fueron estructuradas de manera secuencial y colaborativa, con una coordinación que permitió avanzar desde la investigación hasta la integración final del sistema. Aunque cada integrante asumió una responsabilidad específica, el equipo trabajó de manera conjunta para garantizar la consistencia del trabajo en el desarrollo.

- **Juan Pablo Birsa y Nicolás Arancibia:** Lideraron la investigación sobre la importación de los datos. Esto incluyó analizar el tipo de archivo proporcionado (.csv), entender su estructura y distribución, y determinar la mejor forma de cargarlo en las tablas de la base de datos utilizando la función `\COPY` de PostgreSQL.
- **Nicole Salama:** Implementó la asignación de dorsales correspondientes a cada jugador. Su trabajo fue fundamental para mantener la integridad de los datos en esta etapa crucial del proyecto.
- **Augusto Barthelemy:** Diseñó e implementó el trigger que validaba la dependencia funcional **Equipo Dorsal**  $\rightarrow$  **Jugador**, protegiendo la integridad del sistema.

- **Nicolás Arancibia y Nicole Salama:** Desarrollaron las funciones necesarias para filtrar y analizar los datos a partir de una fecha específica.
- **Augusto Barthelemy:** Supervisó el funcionamiento global del proyecto, verificando la correcta integración de todas las partes del sistema.
- **Juan Pablo Birsa:** Encargado de la documentación final del proyecto, detallando cada paso del proceso y la integración de las diferentes partes del sistema.

### 3 Proceso de importación de los datos

El proceso de importación de los datos se realizó utilizando el comando `COPY`, una herramienta que PostgreSQL provee para transferir datos desde un archivo externo, como por ejemplo un `.txt` o un `.csv`. Esta funcionalidad fue presentada previamente durante el desarrollo del "Trabajo Práctico 10: Importación / Exportación". En particular, el comando `COPY` es una operación que requiere privilegios de **superuser**, ya que interactúa directamente con el sistema de archivos del servidor de la base de datos. Para evitar esta limitación, utilizamos su variante `\COPY`, que se ejecuta desde el cliente y no requiere estos privilegios elevados.

Por otro lado, dado que en el archivo `.csv` los datos están separados por `;`, se especificó un `DELIMITER` con este mismo carácter. Además, la primera fila del archivo, que contiene los nombres de los atributos de cada jugador, fue ignorada utilizando la opción `HEADER TRUE`.

Asimismo, para asegurar que los espacios en blanco en los archivos CSV se interpreten como valores `NULL` en la base de datos, se utilizó la opción `NULL ''`. Esto permite que cualquier celda vacía o que contenga solo un espacio en blanco sea tratada como un valor `NULL` en la tabla.

Listing 1: Comando de importación de datos

```

1 \COPY futbolista(nombre, posicion, edad, altura, pie,
2 fichado, equipo_anterior, valor_mercado, equipo)
3 FROM 'path/jugadores-2022.csv'
4 WITH (FORMAT csv, DELIMITER ';', NULL '',
5 HEADER TRUE, ENCODING 'UTF8');
```

### 4 Investigaciones realizadas

Durante el desarrollo del trabajo, fue necesario investigar y resolver diversas dificultades técnicas que surgieron en el proceso.

Al momento de importar los datos, a partir de la función `\COPY`, es necesario especificar el **ENCODING** correspondiente al archivo de tipo `.csv` con el que estábamos trabajando. Esto se resolvió utilizando la herramienta `file`, disponible en sistemas `Unix/Linux`, que permite identificar el formato del archivo y su encoding.

Por otro lado, al momento de asignar los dorsales a cada jugador, la solución más conveniente fue utilizar **Arrays**, tanto para los dorsales que ya habían sido ocupados como para obtener los dorsales posibles para cada posición. Para entender cómo utilizar este tipo de dato correctamente, fue necesario consultar la [Documentación de PostgreSQL Arrays](#).

### 5 Dificultades encontradas y sus soluciones

Como se mencionó anteriormente, cada dificultad que surgió fue abordada mediante investigación y soluciones específicas.

- **Importación de datos:** La especificación del **ENCODING** se resolvió mediante la herramienta `file`.
- **Asignación de dorsales:** Se implementó una solución basada en **Arrays** para gestionar eficientemente los dorsales disponibles y ocupados.
- **Generación de reportes:** Se utilizó la función `REPEAT()` para mejorar la visualización de los datos en pantalla.

## 6 Conclusión

El Trabajo Práctico Especial permitió aplicar herramientas avanzadas de SQL, como **PSM** y **Rriggers**, para resolver limitaciones que no pueden ser abordadas con mecanismos estándar y garantizar la preservación de dependencias funcionales tras la normalización a **BCNF**. Además, la importación eficiente de datos mediante `\COPY`, junto con el uso de estructuras como **Arrays** y funciones como `REPEAT()`, optimizó la asignación de dorsales y la presentación de reportes, asegurando consistencia y robustez en el sistema. El trabajo en equipo fue fundamental para resolver los problemas técnicos y completar el proyecto de manera exitosa.