

Universidad Nacional de Costa Rica Sede Regional Chorotega

Carrera

Ingeniería en Sistemas

Curso

Diseño e Implementación de Base de Datos

Tema:

Gestor FIFA Datos

Nombre del Equipo:

DCA Developers

Integrantes

Cristhian Altamirano Montes

Ariana Hernández Peña

Danny Orlando Abarca Alvarado

Profesor

Francisco Coulon Ollivier

Documento explicativo: Tecnologías utilizadas y retos para una versión final

Para el desarrollo de la solución planteada en el proyecto de **Herramienta de Mantenimiento FIFA**, se utilizaron diversas tecnologías enfocadas en la manipulación, normalización, almacenamiento y consulta de datos estructurados. A continuación se describen dichas tecnologías y las decisiones detrás de su uso, así como los principales desafíos enfrentados y posibles mejoras para una versión final.

Tecnologías utilizadas

La herramienta se construyó con base en una **base de datos relacional** implementada en **SQL Server**, un sistema de gestión de bases de datos robusto y ampliamente utilizado en entornos empresariales. SQL Server fue seleccionado debido a su capacidad de manejar relaciones complejas entre tablas, aplicar restricciones de integridad referencial mediante claves foráneas, y soportar funcionalidades avanzadas como vistas, procedimientos almacenados, expresiones calculadas y triggers si se desea escalar a futuro.

Los datos provienen de archivos públicos alojados en Kaggle, específicamente de los datasets de FIFA 22. Estos datasets fueron extraídos en formato .csv y procesados previamente para eliminar inconsistencias, columnas innecesarias, y estructuras repetitivas. Este preprocesamiento se realizó utilizando herramientas como **Excel** que permitieron limpiar los datos, normalizarlos y adaptarlos al esquema de la base de datos.

Posteriormente, se generaron scripts de inserción (INSERT INTO) para poblar las tablas Jugadores, Clubes, Nacionalidades, Posiciones, Habilidades y las tablas intermedias para relaciones muchos a muchos, como Jugador_Posicion y Jugador_Habilidad. También se diseñaron procedimientos almacenados para insertar evaluaciones y generar cálculos automáticos de ratings, así como vistas que permiten consultar jugadores según club o nacionalidad.

Todo el desarrollo fue orientado al uso exclusivo de **T-SQL (Transact-SQL)**, el dialecto de SQL específico de SQL Server. Se evitaron herramientas gráficas para simular un entorno de trabajo por consola o script puro, en línea con los objetivos académicos del laboratorio.

Retos enfrentados y propuestas para una versión final

Uno de los principales retos fue la **carga inicial de datos desde archivos CSV**, debido a variaciones regionales en la codificación (como UTF-8 con o sin BOM), separadores (coma o punto y coma) y columnas invisibles generadas por herramientas como Excel (Unnamed: 0). Esto se resolvió eliminando columnas innecesarias, asegurando codificación correcta y generando scripts de inserción manual cuando BULK INSERT fallaba debido a limitaciones de permisos en SQL Server.

Otro desafío importante fue la **integridad referencial**. La presencia de claves foráneas requería asegurar que los datos en las tablas dependientes (como Jugador_Habilidad) solo hicieran referencia a jugadores o habilidades previamente insertadas. Esto obligó a implementar filtros y validaciones para evitar conflictos de integridad.

En una versión final, se sugiere incorporar un **módulo de validación previa de datos** antes de insertarlos, ya sea a través de procedimientos almacenados o mediante una herramienta intermedia escrita en Python o C#. También sería conveniente automatizar la carga de archivos CSV usando scripts programáticos que generen dinámicamente los INSERT INTO en lugar de depender exclusivamente de SQL manual.

Finalmente, un reto común en bases de datos de este tipo es la **actualización periódica de información**, como ratings semanales o cambios de club. Para esto, se propuso y se implementó una tabla Evaluaciones con un campo calculado que simula el cálculo automático del nuevo rating de un jugador con base en su rendimiento anterior, su puntuación en SofaScore y una nota equilibradora.

En resumen, el sistema cumple con los objetivos académicos del proyecto, emplea buenas prácticas de modelado relacional y demuestra la capacidad de trabajar con datos complejos y relaciones múltiples. Para una versión final más robusta, sería ideal incorporar un backend que permita interacción dinámica con la base de datos, así como validación avanzada, trazabilidad de cambios y mecanismos de seguridad para proteger la integridad de los datos.