



Facultad de
Ciencias
UNAM

PrediLigaMX: Modelo Predictivo Basado en Montecarlo, Web Scraping y Bases de Datos para Partidos de la Liga MX

Brandon Arturo Delgadillo Zúñiga
brandondz2002@ciencias.unam.mx

GitHub:
<https://github.com/HitBrann>

Profesor. Miguel Ángel Pérez León
<https://github.com/Jugernaut>
Ayudante. Jonathan Natael Ramírez Montes
<https://github.com/NataelM>

16 de junio de 2023

Resumen: En este proyecto se busca crear un modelo predictivo de resultados de partidos de fútbol de la Liga MX, para esto se usará un modelo probabilístico basado en métodos de Montecarlo, además de que se usará Web Scraping para obtener los datos de la página oficial de la Liga MX y footystats.org, finalmente se usará una base de datos para almacenar y manejar los datos obtenidos.

1. Introducción:

Primeramente, la idea principal de este proyecto nos fue proporcionada por el ayudante Jonathan Natael Ramírez Montes, quien nos proporcionó un modelo probabilístico para predecir resultados de partidos de fútbol, el cual se basaba en calcular mediante un archivo de datos la probabilidad de que un equipo gane, empate o pierda un partido, y posteriormente, mediante el uso de una distribución de probabilidad de Poisson, se calculaba la probabilidad de que un equipo anotara cierta cantidad de goles, con lo cual se calculaba la probabilidad de que un equipo ganara, empatara o perdiera un partido, y con esto se podía predecir el resultado de un partido. El proyecto original puede ser encontrado en el siguiente enlace: https://github.com/NataelM/soccer_model

Sin embargo, el proyecto original tenía algunos problemas, el primero de ellos es que los datos que se usaron para crear el modelo eran muy desactualizados, por lo que decidimos obtener nuestros propios datos de la página oficial de la Liga MX mediante el uso de Web Scraping, además de que decidimos obtener datos de la página footystats.org, la cual cuenta con datos de partidos de fútbol de muchas ligas, incluyendo la Liga MX, lo cual nos permitió obtener más datos y así tener un modelo más preciso. Decidimos usar una base de datos para almacenar los datos obtenidos, esto con el fin de que el

programa no tenga que obtener los datos cada vez que se quiera hacer una predicción, sino que los obtenga una vez y los guarde en la base de datos, y cuando se quiera hacer una predicción, el programa obtenga los datos de la base de datos, además de que esto nos permite tener mayor seguridad de que los datos no se van a perder o a modificar.

Como detalle adicional decidimos que lo mejor era hacer del proyecto original el cual era un conjunto de celdas de Jupyter Notebook, un programa de Python, por lo que decidimos crear una interfaz accesible para el usuario, la cual se puede usar desde la terminal, esto con el fin de que el usuario no tenga que preocuparse tanto por el código, sino que solo tenga que ejecutar el programa y seguir las instrucciones que este le proporcione además de que nos permitía tratar de evitar el mayor número de errores o acciones que pudieran dañar el programa aunque mantuvimos la estructura que se nos proporcionó en clase para la organización de las carpetas y archivos.

La información de la base de datos necesaria para la ejecución del programa puede ser configurada por el usuario, esto con el fin de que el usuario pueda usar la base de datos que desee, el programa puede ser ejecutado en cualquier sistema operativo que cuente con Python 3.11 o superior, además de que el programa cuenta con un archivo de requerimientos para que el usuario pueda instalar las librerías necesarias para la ejecución del programa.

Para realizar todo lo anteriormente mencionado, se reestructuró prácticamente todo el código del proyecto original, se agregaron nuevas funciones y se modificaron otras, se hizo uso de las librerías Selenium y webdriver-manager para el Web Scraping, se usó la librería mysql.connector para la conexión con la base de datos, además para el desarrollo de las simulaciones por Montecarlo se usó una reestructuración del código del usuario **TacticsBadger** en github, el cual puede ser encontrado en el siguiente enlace: <https://github.com/TacticsBadger/MonteCarloFootballMatchSim> con lo cual se obtuvo un modelo probabilístico más preciso en combinación con los datos obtenidos de las páginas web mencionadas anteriormente.

El modelo predictivo no funciona para años anteriores a la temporada 2021-2022, esto debido a que vimos innecesario obtener datos de años anteriores, pero el programa puede ser modificado para que funcione con años anteriores a la temporada 2021-2022 simplemente normalizando los datos de la base de datos, para que coincidan con los de el programa, además de que se puede modificar el programa para que funcione con otras ligas de fútbol, aunque esto requeriría de un poco más de trabajo y más Web Scraping.

1.1. Objetivos:

- **Objetivo General:** Crear un modelo predictivo de resultados de partidos de fútbol de la Liga MX.
- **Objetivos Específicos:**
 - Obtener datos de la página oficial de la Liga MX.[✓]
 - Obtener datos de la página footystats.org.[✓]
 - Crear una base de datos con los datos obtenidos.[✓]
 - Adaptar el modelo original a nuestro proyecto.[✓]
 - Limpiar los datos obtenidos.[✓]
 - Analizar los datos obtenidos.[✓]
 - Crear un modelo probabilístico para predecir resultados de partidos de fútbol de la Liga MX. [Mejorable]
 - Crear una interfaz para el usuario.[✓]

1.2. Justificación:

La razón por la que se optó por hacer este proyecto es porque nos pareció interesante la idea de poder predecir resultados de partidos de fútbol, además de que nos pareció un buen reto el tener que adaptar el proyecto original a nuestro proyecto, así como el implementar el uso de técnicas de Web Scraping y el uso de una base de datos, técnicas que son muy útiles en el mundo laboral actualmente, aparte del hecho de que nos permitió aprender a usar nuevas librerías de Python.

También nos pareció interesante el margen de mejora que tenía el concepto original, ya que nos permitió hacer uso de nuestra creatividad para mejorarlo, además de que nos permitió entender su lógica y su funcionamiento, lo cual nos permitió aprender más sobre el tema e implementar nuestras propias ideas.

1.3. Marco Teórico:

1.3.1. Montecarlo:

Montecarlo es un método que se usa para aproximar soluciones de problemas mediante el uso de números aleatorios, este método se basa en la ley de los grandes números, la cual dice que la media de los resultados de un experimento aleatorio tiende a acercarse al valor esperado cuando el número de repeticiones del experimento es muy grande, por lo que este método se basa en la generación de números aleatorios para simular un experimento aleatorio, el cual se repite un gran número de veces, y se calcula la media de los resultados obtenidos, la cual se acerca al valor esperado.

En este caso se utiliza para generar variables aleatorias que representan los goles que anotan los equipos en un partido de fútbol, esto se hace mediante el uso de una distribución de Poisson, la cual se basa en la siguiente fórmula:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (1)$$

Donde λ es el parámetro de la distribución, el cual para este caso es el valor del xG de cada equipo.

1.3.2. xG:

xG es una métrica que se usa para medir la calidad de un disparo, esta métrica se basa en la probabilidad de que un disparo termine en gol, esta probabilidad se calcula mediante el uso de un modelo estadístico que toma en cuenta factores como la distancia del disparo, el ángulo del disparo, la posición del jugador que dispara, entre otros factores, este modelo estadístico se basa en datos de partidos de fútbol reales, por lo que se puede considerar que es un modelo probabilístico, en este caso el xG fue obtenido de la página web footystats.org.

1.3.3. Liga MX:

La Liga MX es la liga de fútbol profesional más importante de México, esta liga se juega en dos torneos por año, el torneo de apertura y el torneo de clausura, cada torneo se juega con 18 equipos, los cuales juegan entre sí en un formato de todos contra todos, al final de la temporada regular los 8 mejores equipos pasan a la liguilla, la cual se juega en formato de eliminación directa, en el cual los equipos se enfrentan en partidos de ida y vuelta, el equipo que anote más goles en el marcador global pasa a la siguiente ronda, en caso de que el marcador global termine empatado, el equipo que anote más goles de visitante pasa a la siguiente ronda, en caso de que el marcador global y el marcador de visitante terminen empatados, se juega un tiempo extra, en caso de que el marcador siga empatado, se define al ganador mediante una tanda de penales.

1.3.4. Web Scrapping:

El concepto más básico de Web Scraping es el de extraer datos de una página web, como cuando se copia y pega información de una página web a un documento de Word, sin embargo, el Web Scraping va más allá de eso, ya que permite extraer datos de una página web de forma automática, esto se hace mediante el uso de varios scripts que se encargan de extraer los datos de la página web, y guardarlos en un archivo, lo cual permite obtener los datos de una página web de forma más rápida y sencilla, además de que se puede programar para que se ejecute de forma automática, por lo que se puede obtener información de una página web de forma periódica, sin necesidad de que alguien tenga que hacerlo manualmente lo cual ahorra tiempo y evita errores humanos. Para este proyecto se utilizó la librería Selenium de Python para hacer Web Scraping.

1.3.5. Bases de Datos:

Una base de datos es un conjunto de datos relacionales que se organizan de forma que se pueda acceder a ellos de forma sencilla, además de que se pueden agregar, modificar o eliminar datos de forma eficiente, esto permite almacenar grandes cantidades de datos de forma organizada y es posible acceder a ellos mediante consultas, lo cual permite obtener información específica de los datos almacenados. Para este proyecto se utilizó la librería mysql.connector de Python para hacer consultas a la base de datos.

1.3.6. Interfaz de Usuario:

Una interfaz de usuario es un medio de comunicación entre el usuario y un sistema informático, esta interfaz permite al usuario interactuar con el sistema informático de forma sencilla, ya que permite que el usuario pueda darle instrucciones al sistema informático, y que el sistema informático pueda mostrarle información al usuario, esto se logra mediante el uso de elementos gráficos como botones, cajas de texto, imágenes, etc. Para este proyecto se utilizó una interfaz de usuario hecha completamente para ser utilizada desde la terminal lo cual ahorra recursos y permite que el programa se ejecute de forma más rápida, tratando de mantener la interfaz lo más sencilla posible para que sea fácil de usar.

1.3.7. Programación concurrente:

La programación concurrente es un paradigma de programación que se basa en la ejecución de múltiples procesos al mismo tiempo, esto permite que se puedan ejecutar múltiples tareas a la vez, lo cual permite que el programa se ejecute de forma más rápida, ya que se pueden ejecutar múltiples tareas simultáneamente, en lugar de ejecutar una tarea a la vez, hacemos mención de este paradigma de programación ya que se utilizó para hacer que el programa se ejecute de forma más rápida y eficiente, además de que permitió realizar múltiples simulaciones al mismo tiempo, lo cual permitió hacer pruebas de efectividad del programa sin tener que esperar tanto tiempo para obtener los resultados.

2. Obtención de Datos:

Cuando se trata de obtener datos de una página web, lo más sencillo es hacer uso de una API, sin embargo no todas las páginas web cuentan con una API pública, por lo que en este caso se optó por hacer uso de Web Scrapping, basándonos en diversos scripts que extraen automáticamente toda la información que nos fuese útil de la página web de la liga MX, y de footystats.org.

Cuando percibimos el paso del tiempo sobre el archivo de datos del proyecto original, nos dimos cuenta que era mejor idea obtener nuestros propios datos, ya que los datos del proyecto original se obtuvieron en el 2018 y buscábamos obtener predicciones más actuales, por lo que se optó por obtener los datos de la página web de la liga MX, la cual cuenta con datos de todos los partidos de la liga MX desde el 2011 hasta el presente, se obtuvieron los datos de todos los partidos de la liga MX desde el 2021 hasta el 2022, esto se realizó mediante el uso de un script como el antes mencionado, también se obtuvieron los datos de los prtidos xG del año anterior al presente, esto se realizó mediante el uso de un script que se encarga de extraer los datos de la página web footystats.org.

También se obtuvieron los datos de la tabla general de la liga MX, en la sección de la página web de la liga MX que muestra la tabla general, se puede ver la tabla general de la liga MX, la cual muestra la posición de cada equipo en la tabla general, el número de partidos jugados, el número de partidos ganados, el número de partidos empatados, el número de partidos perdidos, el número de goles a favor, el número de goles en contra, la diferencia de goles, y el número de puntos, por lo que se obtuvieron los datos de la tabla general de la liga MX, también esto se realizó mediante el uso de un script. Y en esa misma página web se obtuvieron los datos de la tabla de ofensiva y defensiva de la liga MX, la cual muestra la posición de cada equipo en la tabla, el número de goles a favor y el número de goles en contra.

Finalmente para fines de comparación, se obtuvieron los datos de la tabla de partidos de la liga MX del año presente, esto se realizó reciclando el script que se encarga de extraer los datos de la página web de partidos de la liga MX, y guardarlos en un archivo aparte, esto con el fin de poder comparar los datos obtenidos de la tabla de partidos de la liga MX del año presente, con el modelo aplicado a los datos obtenidos de la tabla de partidos de la liga MX del año anterior al presente, y así poder comprobar que tan efectivo es el modelo de predicción.

2.1. Web Scrapping: Selenium

Para realizar todo lo anterior se hizo uso de Selenium, el cual es una herramienta que permite automatizar tareas en páginas web, esto se logra mediante el uso de un script que se encarga de realizar las tareas de forma automática como si fuera un usuario real, esto permite que se puedan ejecutar tareas repetitivas de forma automática, su uso puede requerir conocimientos de JavaScript para poder hacer tareas más complejas.

El algoritmo de Web Scrapping se puede dividir en 6 partes:

- Configuración: En esta parte se configura el navegador y se instala el driver.
- Inicialización: En esta parte se inicializa el navegador, y se abre la página web.
- Eliminación de elementos: En esta parte se eliminan los elementos innecesarios de la página web como banners de publicidad, o elementos que no son necesarios para extraer los datos y que podrían interferir con la tarea de extracción de datos.
- Interacción con la página web: En esta parte se interactúa con la página web, clickeando botones, llenando formularios, etc.

- Extracción de datos: En esta parte se extraen los datos de la página web, y se guardan en un archivo, este paso y los dos anteriores se pueden repetir varias veces, dependiendo de la cantidad de datos que se quieran extraer.
- Finalización: En esta parte se cierra el navegador y se procede a realizar las tareas necesarias con los datos extraídos.

En total utilizamos 3 scripts de Web Scrapping, uno para extraer los datos de la tabla de partidos de la liga MX, otro para extraer los datos de la tabla general, ofensiva y defensiva de la liga MX, y otro para extraer los datos xG de la liga MX, los cuales se ejecutan de forma automática, y guardan los datos en un archivo CSV, el cual se puede utilizar para realizar las tareas necesarias con los datos extraídos, cosa que hacemos en el siguiente paso.

Estadística Histórica

Estadística Patrocinada por: **FOOTBALL™**

Por Temporada: 2023-2024 Por Torneo: Apertura **Buscar**

TABLA GENERAL

Pos	Club	Local								Visitante							
		JJ	JG	JE	JP	GF	GC	Dif	PTS	JJ	JG	JE	JP	GF	GC	Dif	PTS
1	América	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Atlas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Atlético de San Luis	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Tijuana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Cruz Azul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 1: Página de Tabla General/Ofensiva/Defensiva de la liga MX

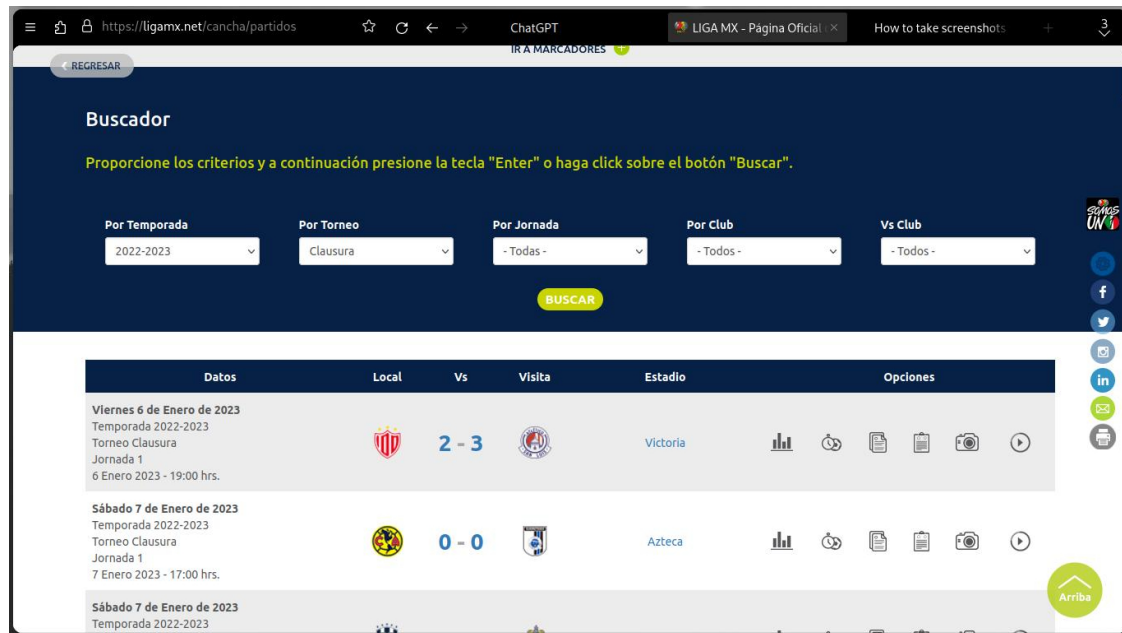


Figura 2: Página de Tabla de Partidos de la liga MX

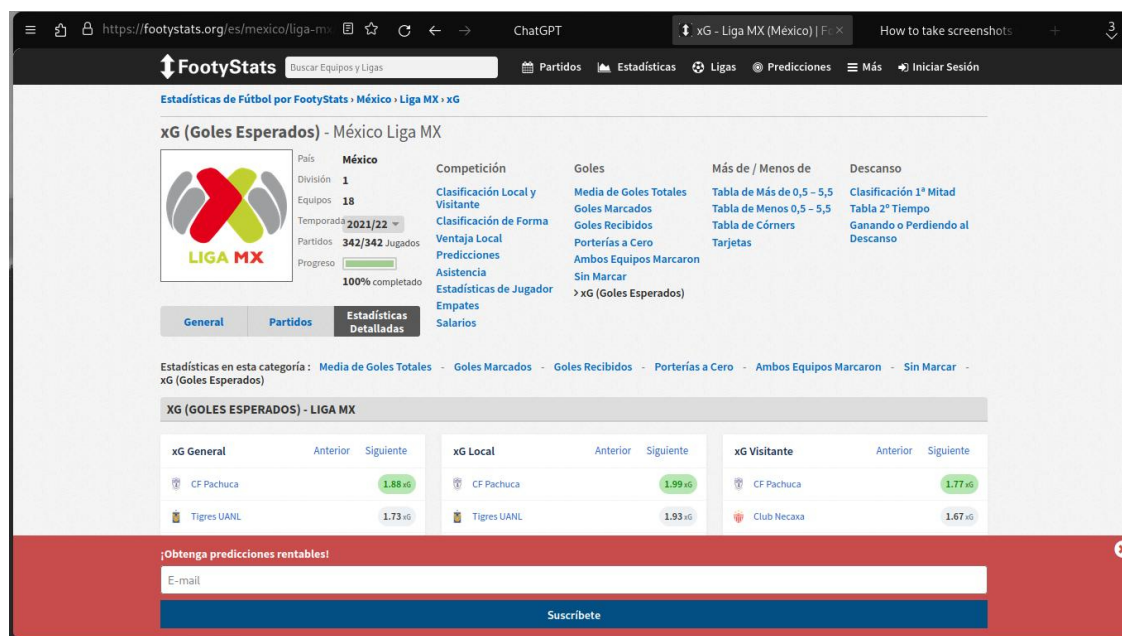


Figura 3: Página de xG de la liga MX

3. Base de Datos: MySQL

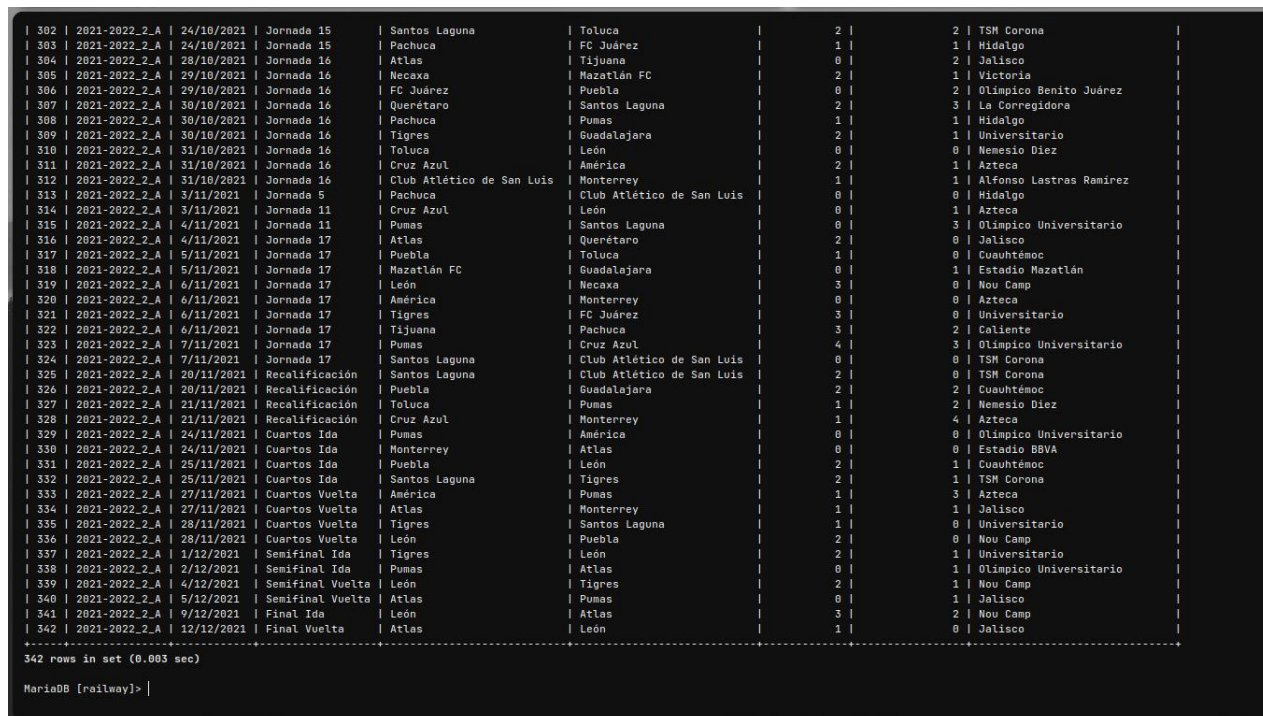
3.1. Creación de la Base de Datos

En la actualidad, ocupamos una base de datos MySQL para almacenar los datos obtenidos de la página web de la liga MX, aunque en un principio se utilizaba una base de datos con posibilidad de conectarse remotamente, ahora usa una base de datos local, esto se debe a que la base de datos remota podría ser vulnerada, por lo que alguien podría acceder a los datos de la base de datos, y modificarlos, lo cual podría afectar el funcionamiento del modelo de predicción, se optó por permitir configurar la base de datos mediante un archivo de configuración (JSON), en lugar de tener que modificar el programa de la base de datos cada vez que se quiera cambiar la configuración de la base de datos, esto permite que se pueda configurar la base de datos de forma más rápida y sencilla, además de que en el primer uso del programa, te permite configurar la base de datos. Así cada usuario puede utilizar su propia base de datos local, y no tiene que preocuparse por que alguien pueda acceder a los datos de su base de datos.

3.2. Creación de las Tablas y Subida de Datos

Configuramos los scripts para que creen la base de datos correspondiente, y las tablas necesarias para almacenar los datos, una vez hecho eso se procede a leer los datos de los archivos CSV generados en nuestro Web Scrapping, y se suben los datos a la base de datos, una vez hecho eso, ya se pueden realizar las tareas necesarias con los datos extraídos.

Debido a esto es necesario que el usuario de MySQL tenga permisos de creación de base de datos y tablas, además de permisos de lectura y escritura en la base de datos.



302	2021-2022_2_A	24/10/2021	Jornada 15	Santos Laguna	Toluca	2	2	TSM Corona	
303	2021-2022_2_A	24/10/2021	Jornada 15	Pachuca	FC Juárez	1	1	Hidalgo	
304	2021-2022_2_A	29/10/2021	Jornada 16	Atlas	Tijuana	0	0	Jalisco	
305	2021-2022_2_A	29/10/2021	Jornada 16	Necaxa	Mazatlán FC	2	1	Victoria	
306	2021-2022_2_A	29/10/2021	Jornada 16	FC Juárez	Puebla	0	2	Olimpico Benito Juárez	
307	2021-2022_2_A	30/10/2021	Jornada 16	Querétaro	Santos Laguna	2	3	La Corregidora	
308	2021-2022_2_A	30/10/2021	Jornada 16	Pachuca	Pumas	1	1	Hidalgo	
309	2021-2022_2_A	30/10/2021	Jornada 16	Tigres	Guadalajara	2	1	Universitario	
310	2021-2022_2_A	31/10/2021	Jornada 16	Toluca	León	0	0	Nemesio Diez	
311	2021-2022_2_A	31/10/2021	Jornada 16	Cruz Azul	América	2	1	Azteca	
312	2021-2022_2_A	31/10/2021	Jornada 16	Club Atlético de San Luis	Monterrey	1	1	Alfonso Lastras Ramirez	
313	2021-2022_2_A	3/11/2021	Jornada 5	Pachuca	Club Atlético de San Luis	0	0	Hidalgo	
314	2021-2022_2_A	3/11/2021	Jornada 11	Cruz Azul	León	0	1	Azteca	
315	2021-2022_2_A	4/11/2021	Jornada 11	Pumas	Santos Laguna	0	3	Olimpico Universitario	
316	2021-2022_2_A	4/11/2021	Jornada 11	Atlas	Querétaro	2	0	Jalisco	
317	2021-2022_2_A	5/11/2021	Jornada 17	Puebla	Toluca	1	0	Cuauhtémoc	
318	2021-2022_2_A	5/11/2021	Jornada 17	Mazatlán FC	Guadalajara	0	1	Estadio Mazatlán	
319	2021-2022_2_A	6/11/2021	Jornada 17	León	Necaxa	3	0	Nou Camp	
320	2021-2022_2_A	6/11/2021	Jornada 17	América	Monterrey	0	0	Azteca	
321	2021-2022_2_A	6/11/2021	Jornada 17	Tigres	FC Juárez	3	0	Universitario	
322	2021-2022_2_A	6/11/2021	Jornada 17	Tijuana	Pachuca	3	2	Caliente	
323	2021-2022_2_A	7/11/2021	Jornada 17	Pumas	Cruz Azul	4	3	Olimpico Universitario	
324	2021-2022_2_A	7/11/2021	Jornada 17	Santos Laguna	Club Atlético de San Luis	0	0	TSM Corona	
325	2021-2022_2_A	20/11/2021	Recalificación	Santos Laguna	Club Atlético de San Luis	2	0	TSM Corona	
326	2021-2022_2_A	20/11/2021	Recalificación	Puebla	Guadalajara	2	2	Cuauhtémoc	
327	2021-2022_2_A	21/11/2021	Recalificación	Toluca	Pumas	1	2	Nemesio Diez	
328	2021-2022_2_A	21/11/2021	Recalificación	Cruz Azul	Monterrey	1	4	Azteca	
329	2021-2022_2_A	24/11/2021	Cuartos Ida	Pumas	América	0	0	Olimpico Universitario	
330	2021-2022_2_A	24/11/2021	Cuartos Ida	Monterrey	Atlas	0	0	Estadio BBVA	
331	2021-2022_2_A	25/11/2021	Cuartos Ida	Puebla	León	2	1	Cuauhtémoc	
332	2021-2022_2_A	25/11/2021	Cuartos Ida	Santos Laguna	Tigres	2	1	TSM Corona	
333	2021-2022_2_A	27/11/2021	Cuartos Vuelta	América	Pumas	1	3	Azteca	
334	2021-2022_2_A	27/11/2021	Cuartos Vuelta	Atlas	Monterrey	1	1	Jalisco	
335	2021-2022_2_A	28/11/2021	Cuartos Vuelta	Tigres	Santos Laguna	1	0	Universitario	
336	2021-2022_2_A	28/11/2021	Cuartos Vuelta	León	Puebla	2	0	Nou Camp	
337	2021-2022_2_A	1/12/2021	Semifinal Ida	Tigres	León	2	1	Universitario	
338	2021-2022_2_A	2/12/2021	Semifinal Ida	Pumas	Atlas	0	1	Olimpico Universitario	
339	2021-2022_2_A	4/12/2021	Semifinal Vuelta	León	Tigres	2	1	Nou Camp	
340	2021-2022_2_A	5/12/2021	Semifinal Vuelta	Atlas	Pumas	0	1	Jalisco	
341	2021-2022_2_A	9/12/2021	Final Ida	León	Atlas	3	2	Nou Camp	
342	2021-2022_2_A	12/12/2021	Final Vuelta	Atlas	León	1	0	Jalisco	

342 rows in set (0.003 sec)

MariaDB [railway]> |

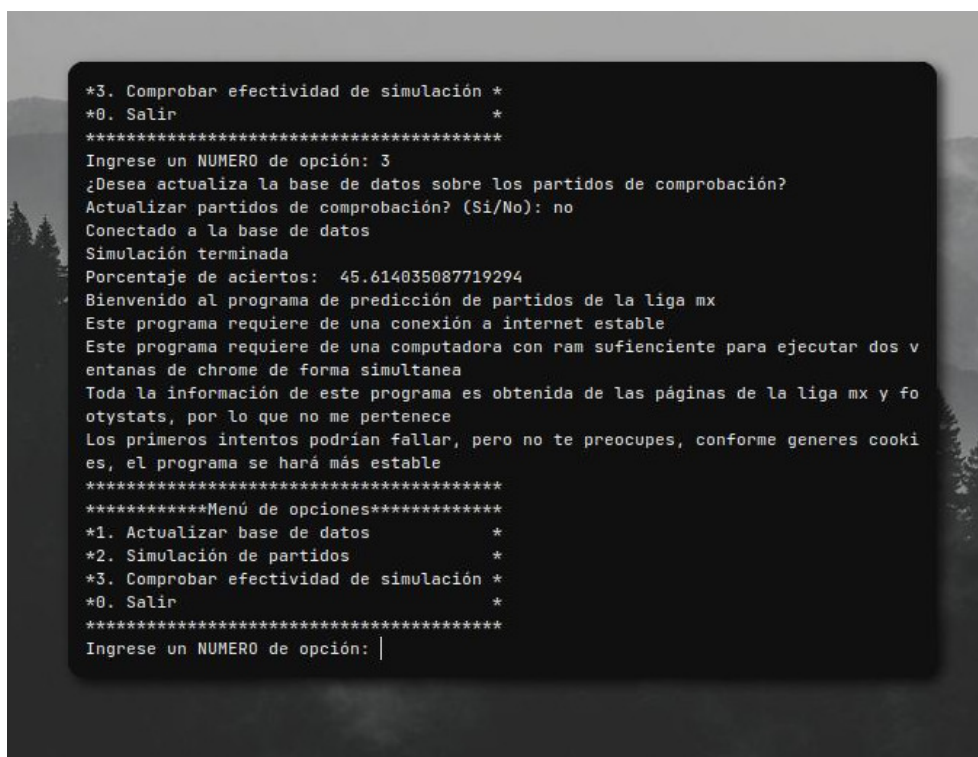
Figura 4: Tabla PARTIDOS en la base de datos

4. Simulación de Partidos por Montecarlo

Para poder realizar las simulaciones de partidos por métodos de Montecarlo, nuestro programa lee nuestra base de datos y obtiene los valores de xG de cada equipo como local o como visitante, según se le asigne en la simulación, y los utiliza como lambda de una distribución de Poisson, y con esta distribución se generan los goles de cada equipo con un máximo de 6 goles por equipo por partido, para luego comparar los goles de cada equipo y determinar el ganador del partido, se repite el proceso para cada partido que se simula el número de veces que se le indique al programa, así se obtiene la probabilidad de que cada equipo gane, empate o pierda, finalmente nuestro programa recomienda apostar a los equipos con mayor probabilidad de ganar, y así se puede comprobar que tan efectivo es el modelo de predicción.

4.1. Efectividad del Modelo de Predicción

Para comprobar que tan efectivo es el modelo de predicción, hacemos que nuestro programa lea los resultados de la tabla de partidos comparando los goles de cada equipo y así obteniendo el resultado sobre si el equipo ganó, empató o perdió. Luego de cada partido tomamos el equipo local y el equipo visitante, para proceder a simularlos con nuestro modelo de predicción, luego comparamos los resultados de la simulación con los resultados reales, con esto podemos comprobar que tan efectivo es el modelo de predicción haciendo uso de todos los partidos de la tabla contando cuantos aciertos y cuantos errores tuvo el modelo de predicción.



```
*3. Comprobar efectividad de simulación *
*0. Salir *
*****
Ingrese un NUMERO de opción: 3
¿Desea actualiza la base de datos sobre los partidos de comprobación?
Actualizar partidos de comprobación? (Si/No): no
Conectado a la base de datos
Simulación terminada
Porcentaje de aciertos: 45.614035087719294
Bienvenido al programa de predicción de partidos de la liga mx
Este programa requiere de una conexión a internet estable
Este programa requiere de una computadora con ram suficiente para ejecutar dos v
entanas de chrome de forma simultanea
Toda la información de este programa es obtenida de las páginas de la liga mx y fo
otystats, por lo que no me pertenece
Los primeros intentos podrian fallar, pero no te preocupes, conforme generes cooki
es, el programa se hará más estable
*****
*****Menú de opciones*****
*1. Actualizar base de datos *
*2. Simulación de partidos *
*3. Comprobar efectividad de simulación *
*0. Salir *
*****
Ingrese un NUMERO de opción: |
```

Figura 5: Resultados de la comprobación de la efectividad del modelo de predicción

5. Resultados Finales

5.1. Resultados:

Gracias a la automatización de los procesos, se logró obtener una base de datos con los datos actualizados de la liga MX, además se logró realizar las simulaciones de partidos por métodos de Montecarlo, y así se logró comprobar que tan efectivo es el modelo de predicción, gracias a esto se logró comprobar que el modelo de predicción es ligeramente superior al propuesto en el proyecto original, ya que el modelo de predicción propuesto en el proyecto original tenía una efectividad de cerca del 40 % mientras que el modelo de predicción propuesto en este proyecto tiene una efectividad de cerca del 45 %, lo cual es una mejora ligeramente significativa, esto se debe a que gracias al uso de los datos xG, pues se logra tener una mejor aproximación de los goles que anotará cada equipo ya que se toman en cuenta más factores que afectan la cantidad de goles que anota un equipo, y por lo tanto se logra tener una mejor aproximación de los resultados de los partidos.

Sin embargo pensamos que se puede mejorar el modelo de predicción, empezando por dejar de depender de una distribución de Poisson, ya que esta distribución no es la mejor para modelar los goles de un partido de fútbol, ya que esta distribución no toma en cuenta diversos factores que afectan la cantidad de goles que anota un equipo, como por ejemplo, el clima, la altura, el estado de la cancha, el estado físico de los jugadores, etc. Por lo que se podría mejorar el modelo de predicción haciendo uso de una distribución que tome en cuenta estos factores, y así se podría tener una mejor aproximación de los resultados de los partidos.

También creemos que se pueden implementar técnicas de ciencias de datos y Machine Learning para mejorar el modelo de predicción, ya que se podrían utilizar los datos de los partidos anteriores para entrenar un modelo de Machine Learning que prediga los resultados de los partidos, y así se podría tener un modelo de predicción más efectivo, ya que se tomarían en cuenta más factores que afectan el resultado de un partido de fútbol, y así se podría tener una mejor aproximación de los resultados de los partidos.

5.2. Conclusiones:

Gracias a este proyecto, logramos aprender a utilizar diversas herramientas de programación, como por ejemplo, el uso de Selenium para realizar Web Scrapping, bases de JavaScript, el uso de MySQL para almacenar los datos extraídos y manipularlos, el uso de la librería NumPy para realizar las simulaciones de partidos por métodos de Montecarlo, y el uso de otras tantas librerías de Python para realizar diversas tareas, como por ejemplo, el uso de la librería Pandas para leer los datos de los archivos CSV generados en nuestro Web Scrapping, el uso de la librería threading para realizar el Web Scrapping de forma más rápida, el uso de la librería de MySQL para conectarse a la base de datos y manipular los datos, entre otras. Además de que logramos aprender la importancia que tiene la información en la actualidad, ya que gracias a la información que se puede obtener de la internet, se pueden realizar diversas tareas que pueden realizar múltiples aportes a la sociedad y a la industria, como por ejemplo, el uso de la información para realizar predicciones para poder tomar mejores decisiones, y así poder tener mejores resultados.