

- Informe
  - Introduccion
    - Breve descripcion del proyecto.
    - Objetivos y metas.
    - Sistema especifico a simular y variables a tener en cuenta.
  - Detalles de la implementacion.
  - Resultados y experimentos.
    - Resumen de los Resultados de la Simulación.
    - Interpretación de los Resultados.
    - Hipótesis Derivada de los Resultados.
    - Experimentos realizados para verificar las hipotesis.

# Informe

---

## Introduccion

---

### Breve descripcion del proyecto.

Este proyecto se enfoca en mostrar, mediante una simulación de eventos discretos y el análisis de una tabla de estadísticas de la Major League Baseball, cómo podemos utilizar únicamente los datos recopilados de la primera mitad de la temporada de 2021 para prever con relativa precisión los resultados de la segunda mitad del campeonato. En resumen, nuestro objetivo es intentar anticipar el clasificador final de la tabla de posiciones basándonos exclusivamente en los datos disponibles hasta ese momento.

### Objetivos y metas.

Nuestro propósito es aproximarnos tanto como sea posible al resultado factual de la temporada. De esta manera, podremos evaluar la precisión de nuestras simulaciones, ya que contamos con los datos reales del problema que servirán como base para las mismas.

### Sistema especifico a simular y variables a tener en cuenta.

Nos proponemos realizar una simulación de la segunda mitad de la temporada de la Major League Baseball del año 2021, empleando como punto de partida los resultados obtenidos en la primera mitad. Para ello, incorporaremos variables adicionales, tales como los posibles lesiones de jugadores, que podrían influir en el desenlace de cada partido.

## Detalles de la implementacion.

---

El código comienza importando las bibliotecas necesarias: pandas para el manejo de datos, random para la generación de números aleatorios y numpy para las operaciones matemáticas.

A continuacion explicaremos el codigo por funciones:

- `load_data(s_date, e_date)` carga los datos de los partidos de un archivo CSV, filtra los partidos que ocurrieron entre las fechas dadas y organiza los resultados en un diccionario, particularmente nos interesan los partidos de la primera mitad de la temporada del 2021.
- `get_history(team1, team2, results)` devuelve el número de partidos jugados y ganados entre dos equipos.
- `simulate_injured_players(p=0.5)` simula si hay jugadores lesionados en un equipo, usando una distribucion binomial con  $p = 0.5$  siendo  $p$  la probabilidad de éxito.
- `simulate_game(team1, team2, results, game_simulations)` simula un partido entre dos equipos basándose en los resultados históricos y en la posibilidad de que haya jugadores lesionados, para hacer esto se calcula una tasa que llamamos "win\_rate", esta nos servira para usar esta tasa en una simulacion de Monte Carlo y estimar el ganador de un partido.
- `simulate_season(statistics, game_simulations)` simula una temporada completa, jugando cada equipo contra todos los demás, por cada par de equipos se simula el juego y se guardan los resultados.
- `create_results_table(total_wins)` convierte los resultados de la simulación en un DataFrame de pandas y lo ordena por el número total de victorias.
- `get_sim_results(num_simulations=30, game_simulations=100)` ejecuta la simulación de la temporada varias veces y devuelve los resultados en forma de tabla.
- `get_real_results()` obtiene los resultados reales de la temporada.

- `calculate_position_distances(df_real, df_simulated)` calcula la diferencia entre las posiciones reales y simuladas de cada equipo.
- `print_results(num_simulations, game_simulations)` ejecuta todo el proceso varias veces y devuelve la distancia media entre las posiciones reales y simuladas de los equipos.

El código termina llamando a `print_results(150, 400)`, lo que significa que se ejecutan 150 simulaciones de la temporada, cada una con 400 simulaciones de partidos, y se imprime la distancia media entre las posiciones reales y simuladas.

## Resultados y experimentos.

---

### Resumen de los Resultados de la Simulación.

La simulación realizada ha mostrado una notable coincidencia con los resultados finales de la temporada observada. A continuación, se presenta una representación gráfica que ilustra la comparación entre los datos reales y los obtenidos a través de la simulación.

 Tabla de Resultados

### Interpretación de los Resultados.

Para profundizar en la comparación entre los datos reales y los simulados, aplicamos la función `calculate_position_distances(df_real, df_simulated)`. Esta herramienta permite identificar y cuantificar las diferencias entre ambos conjuntos de datos, ofreciendo una visión clara sobre la precisión de nuestra simulación.

### Hipótesis Derivada de los Resultados.

Basándonos en los hallazgos obtenidos, podemos formular la hipótesis de que, mediante el uso de simulaciones, es factible predecir con un alto grado de precisión el resultado final de la tabla de posiciones de una temporada de la Major League Baseball, incluso antes de su conclusión. Esto subraya la utilidad de las simulaciones como herramientas predictivas en el ámbito deportivo, especialmente cuando se dispone de datos relevantes de la primera mitad de la temporada.

# Experimentos realizados para verificar las hipótesis.

Realizamos múltiples ejecuciones de la simulación con el objetivo de confirmar que los resultados obtenidos no sean el producto de circunstancias fortuitas o aleatorias, sino que reflejen una tendencia consistente y válida. Este enfoque metodológico nos permite establecer una mayor confianza en la precisión y fiabilidad de nuestros hallazgos, al minimizar el riesgo de atribuir el éxito de la predicción a factores externos o a la casualidad.