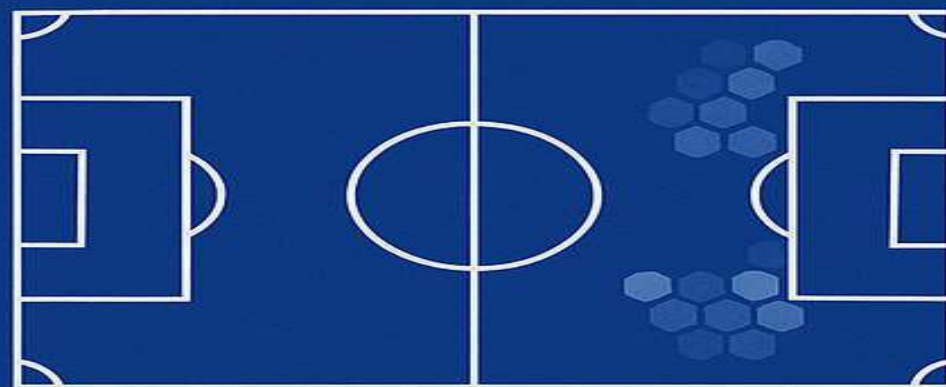
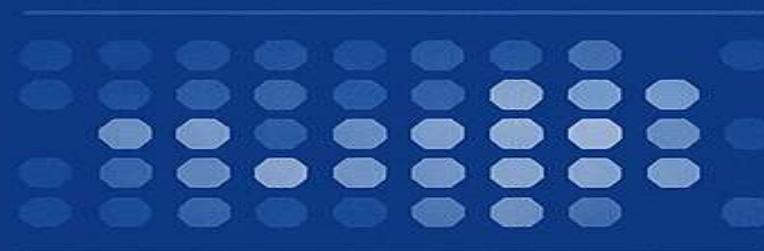
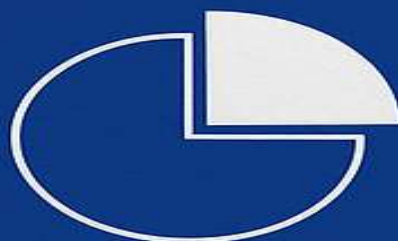


ANÁLISIS DEL JUEGO DEL CHELSEA



Carlos García Orts

Tutor: Jorge López

Máster de Formación Permanente en Big Data Deportivo
2024-2025

- 1.Introducción y objetivos del proyecto
- 2.Arquitectura y metodología de la solución
- 3.Demostración del dashboard interactivo
 1. Análisis post-partido
 2. Visión global
 3. Análisis pre-partido
 4. Forma de los jugadores
- 4.Conclusiones y futuras mejoras

1. Introducción

En el fútbol moderno, el problema ya no es la falta de datos, sino el exceso. Los cuerpos técnicos están inundados de estadísticas complejas y necesitan herramientas que traduzcan ese 'ruido' en señales claras.

Este proyecto nace para solucionar ese problema: transformar miles de datos de partidos en inteligencia táctica accionable y fácil de entender.

- **Objetivo Principal:** Desarrollar una herramienta analítica en Power BI capaz de transformar datos brutos de partidos en insights tácticos accionables para un cuerpo técnico.
- **Objetivos Específicos:**
 - Identificar el **Estilo de Juego** a través de KPIs.
 - Analizar el rendimiento en diferentes **Contextos** (Local/Visitante, Competición).
 - Presentar análisis **PRE y POST Partido** para un ciclo de competición.
 - Demostrar una **metodología** de trabajo profesional.

2. La solución

¿Qué compone la solución?



Un script en Python que extrae automáticamente los datos de los partidos.

Un modelo de datos robusto en Power BI que limpia, relaciona y prepara la información.



Lo más importante, un dashboard interactivo con 4 vistas para responder a las necesidades del entrenador.



El corazón de la solución es un modelo de datos en estrella. La tabla de dimensión DimPartidos filtra todas las tablas de hechos (StatsTiros, StatsPases, etc.), asegurando que todo el dashboard reaccione de forma coherente a la selección de un partido."

¿Cómo se ha realizado?

```

# -- FUNCIÓN DE RECUPERACIÓN --
def recuperar_archivos_partido(ruta_carpetas_partido):
    print("\n-- Recuperando ficheros en: {ruta_carpetas_partido} --")
    nombre_carpetas = os.path.basename(ruta_carpetas_partido)
    try:
        equipo_local, equipo_visitante = nombre_carpetas.split("_vs_")
    except ValueError:
        print("\t - X ERROR: El nombre de la carpeta '{nombre_carpetas}' no tiene el formato '{equipo1}_vs_{equipo2}'")
        return

    for nombre_fichero_antiguo in os.listdir(ruta_carpetas_partido):
        nombre_final = ""
        # Se aplica la recuperación de ficheros
        if nombre_fichero_antiguo == "tabla_1.csv": nombre_final = "{equipo_local}. Resumen.csv"
        elif nombre_fichero_antiguo == "tabla_1.xls": nombre_final = "{equipo_local}. Puntos.xls"
        elif nombre_fichero_antiguo == "tabla_3.csv": nombre_final = "{equipo_local}. TiposIngresos.csv"
        elif nombre_fichero_antiguo == "tabla_4.xls": nombre_final = "{equipo_local}. ArchivosDefensivos.xls"
        elif nombre_fichero_antiguo == "tabla_5.csv": nombre_final = "{equipo_local}. Puntajes.csv"
        elif nombre_fichero_antiguo == "tabla_6.xls": nombre_final = "{equipo_local}. EstadísticasDefensivas.xls"
        elif nombre_fichero_antiguo == "tabla_7.csv": nombre_final = "{equipo_local}. Porteros.csv"
        elif nombre_fichero_antiguo == "tabla_8.xls": nombre_final = "{equipo_visitante}. Resumen.xls"
        elif nombre_fichero_antiguo == "tabla_9.csv": nombre_final = "{equipo_visitante}. Puntos.csv"
        elif nombre_fichero_antiguo == "tabla_10.xls": nombre_final = "{equipo_visitante}. TiposPuntos.xls"
        elif nombre_fichero_antiguo == "tabla_11.csv": nombre_final = "{equipo_visitante}. EstadísticasDefensivas.csv"
        elif nombre_fichero_antiguo == "tabla_12.csv": nombre_final = "{equipo_visitante}. Puntajes.csv"
        elif nombre_fichero_antiguo == "tabla_13.xls": nombre_final = "{equipo_visitante}. EstadísticasDefensivas.xls"
        elif nombre_fichero_antiguo == "tabla_14.csv": nombre_final = "{equipo_visitante}. Porteros.csv"
        elif nombre_fichero_antiguo == "tabla_15.xls": nombre_final = "Tirada_nuevo_equipo1.xls"
        elif nombre_fichero_antiguo == "tabla_16.csv": nombre_final = "{equipo_local}. Tipos.csv"
        elif nombre_fichero_antiguo == "tabla_17.csv": nombre_final = "{equipo_visitante}. Tipos.csv"

    if nombre_final:
        os.rename(os.path.join(ruta_carpetas_partido, nombre_fichero_antiguo), os.path.join(ruta_carpetas_partido, nombre_final))
        print("\t - OK Se ha copiado '{nombre_fichero_antiguo}' a '{nombre_final}'")

```

```

-- EJECUCION PRINCIPAL ---
if __name__ == '__main__':
    chrome_options = Options()
    chrome_options.add_argument("--headless=new")
    service = Service(ChromeDriverManager().install())
    driver = webdriver.Chrome(service=service, options=chrome_options)

    if os.path.exists(CARPETA_RAIZ):
        shutil.rmtree(CARPETA_RAIZ)
    os.makedirs(CARPETA_RAIZ)

    print("-----")
    print(" Fase 1: Iniciando descarga de partidos...")
    print("-----")
    for url in URLS_PARTIDOS:
        descargar_y_llenar(url, CARPETA_RAIZ, driver)
    driver.quit()
    print("\n🏆 Proceso de descarga y llenado completado.")

    print("\n-----")
    print(" Fase 2: Iniciando renombrado de ficheros...")
    print("-----")
    carpetas_de_partidos = [os.path.join(CARPETA_RAIZ, d) for d in os.listdir(CARPETA_RAIZ) if os.path.isdir(os.path.join(CARPETA_RAIZ, d))]
    for carpeta in carpetas_de_partidos:
        renombrar_archivos_partido(carpeta)
    print("\n🏆 Proceso de renombrado completado.")

    print("\n\n🟢🟢🟢 ¡TODO EL PROCESO HA FINALIZADO CON ÉXITO! 🟢🟢🟢")

```

- Llamada a librerías necesarias para el scrapping y modelado de datos.
- Dos funciones principales:
 - descargar_y_limpiar
 - renombrar_archivos_partido
- Ejecución principal

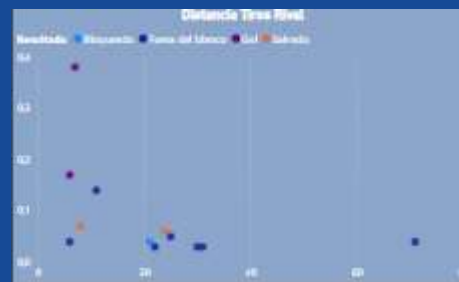
3. Dashboard

Informe Post-Partido



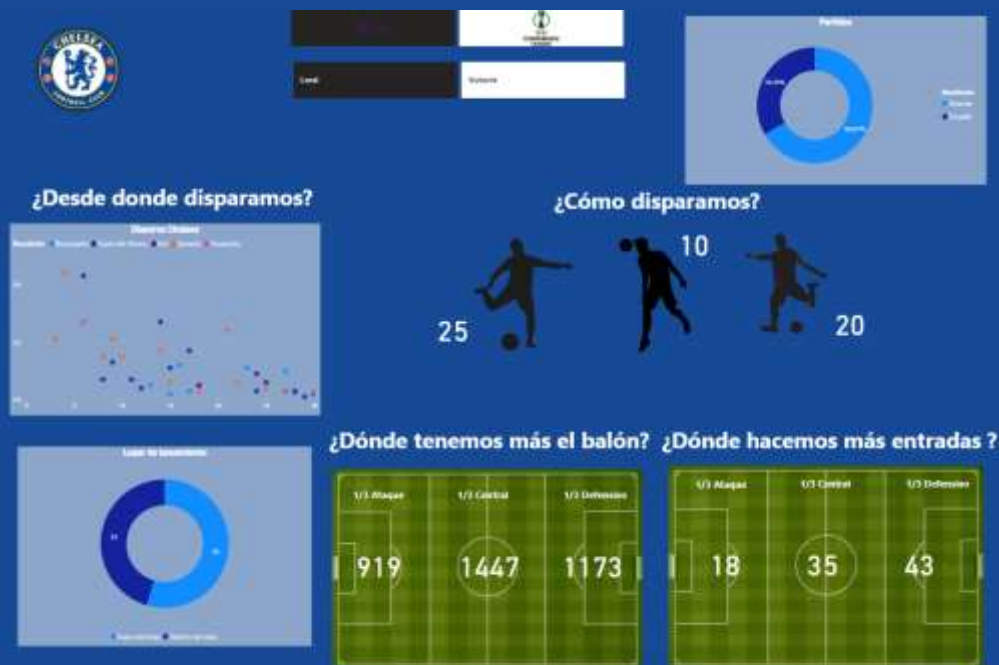
La primera página permite un análisis 360° de un partido ya jugado, comparando el rendimiento de ambos equipos en métricas ofensivas y defensivas clave.

Una de las claves del análisis es el gráfico de xG vs. Distancia. Por ejemplo, en el partido contra el Ipswich Town, vemos que ellos teniendo menos de la mitad de tiros que nosotros nos empataron el partido, sus ocasiones tuvieron un xG (calidad) muy inferior y aún así consiguieron empatarnos el partido.



3. Dashboard

Informe Visión Global



La segunda página agrega los datos de todos los partidos para analizar tendencias y el ADN del equipo. Los filtros permiten estudiar el rendimiento por competición o sede.

Estos dos insights son claves, nos hacen saber donde tenemos más el balón en nuestro caso se puede comprobar que somos un equipo ofensivo por la cantidad de toques que damos en campo contrario.

También corroboramos la presión alta campo ya que tenemos muchas acciones defensivas entre el carril la zona central y la atacante

¿Dónde tenemos más el balón? ¿Dónde hacemos más entradas?



3. Dashboard

Informe Pre-Partido



Finalmente, la página más estratégica. Esta herramienta simula la preparación de un partido, permitiendo al analista estudiar a fondo al próximo rival basándose en su rendimiento en enfrentamientos anteriores.

También deducimos que es un equipo que te ahoga en el centro del campo y que muy posiblemente prepare un partido con un bloque medio.

Es un equipo que le gusta dominar el centro del campo como podemos comprobar con el gráfico de toques de balón

Por ejemplo, para preparar la ida contra el Legia Warsaw. El dashboard identifica inmediatamente a sus jugadores más peligrosos en ataque y a sus pilares defensivos.

4. Conclusiones

- **Éxito del Proyecto:** Se ha desarrollado con éxito una herramienta de BI funcional que cumple todos los objetivos propuestos, transformando datos en inteligencia táctica.
- **Principal Aportación:** La herramienta no solo reporta qué pasó, sino que facilita la comprensión del porqué, conectando datos con el modelo de juego.
- **Insight Clave:** El análisis reveló que, aunque el Chelsea es dominante en posesión, su principal área de mejora es la conversión de tiros desde fuera del área, que tienen un xG promedio muy bajo.



4. Mejoras

- **Más Datos:** Integrar datos de tracking (GPS) para analizar distancias recorridas y velocidades.
- **Mejores Datos:** Tener la capacidad de poder identificar donde está cada jugador para poder hacer mapas de calor, ubicaciones exactas de disparos, donde se encuentra cada jugador en la presión...
- **Machine Learning:** Implementar modelos predictivos para estimar el resultado de un partido o el riesgo de lesión de un jugador.
- **Automatización:** Crear un flujo en la nube (ej: Azure Functions) que ejecute el script de Python y actualice el dashboard diariamente de forma 100% automática.

5. Referencias

Menciona brevemente las fuentes y herramientas.

- **Fuente de Datos:** fbref.com
- **Tecnologías:** Python (Selenium, Pandas, os, io, bs4, time, shutil), Microsoft Power BI (Power Query, DAX).

