# **Documentación de la Aplicación de Análisis de Baloncesto**

## **1. Funcionalidades Implementadas**

La aplicación web desarrollada con Dash y Flask proporciona un conjunto completo de herramientas para el análisis de rendimiento deportivo y seguimiento médico de jugadores de baloncesto. Las funcionalidades clave implementadas incluyen:

* **Autenticación y Seguridad**
  + Sistema de login/logout con Flask-Login.
  + Control de sesión permanente y expiración automática.
  + Restricción de acceso a dashboards para usuarios autenticados.
* **Performance Dashboard**
  + Filtros por temporada, equipo, rango de fechas y resultado (victoria/derrota).
  + Cálculo y visualización de:
  + Rating ofensivo y defensivo.
  + Cuatro factores (%eFG, %Rebote Ofensivo, %Pérdidas, FT Rate).
  + Evolución temporal de resultados.
  + Donuts de distribución de posesiones propias y rivales.
  + Gráfico radar de comparativa avanzada entre equipo y rivales.
  + Generación de informe PDF completo personalizado con los datos y gráficas.
* **Dashboard Médico**
  + Filtros por temporada, tipo de lesión, equipo, jugador y rango de fechas.
  + Gráficos de:
  + Cantidad de lesiones por tipo.
  + Duración de lesiones.
  + Tabla comparativa con información detallada de lesiones (jugador, tipo, cantidad, días de baja).
* **Rendimiento y Optimización**
  + Sistema de caché con Flask-Caching para funciones de carga y procesamiento de datos.
  + Almacenamiento de datos en base de datos SQLite y archivos CSV.
  + Separación modular por callbacks, layouts, componentes, y utilidades.

## **2. Desafíos Encontrados y Soluciones**

|  |  |
| --- | --- |
| **Desafío** | **Solución** |
| Generar gráficas en formato imagen para PDF con Plotly | Se optó por usar orca en lugar de kaleido por problemas de compatibilidad. |
| Insertar logos e imágenes de forma centrada en el PDF | Se calculó la posición horizontal de forma dinámica para centrar cada imagen. |
| Optimizar tiempos de carga | Se usó @cache.memoize(timeout=300) en funciones pesadas como consultas a SQLite o gráficas complejas. |
| Filtrado complejo de datos médicos por múltiples condiciones | Se aplicó una lógica clara con listas y comprobaciones tipo isinstance para asegurar compatibilidad. |
| Control de sesión y redirecciones seguras | Se implementó validación en cada render de página para evitar acceso sin login. |

## **3. Decisiones de Diseño**

* **Framework**: Dash por su integración sencilla con componentes HTML/CSS y su extensión natural con Python.
* **Login**: Se implementó con Flask en lugar de usar soluciones externas para tener mayor control.
* **Caché**: Se eligió filesystem para asegurar persistencia entre ejecuciones locales sin complejidad.
* **Estilo visual**: Bootstrap y colores definidos por roles (azules para el equipo, rojos para rivales) para una experiencia intuitiva.
* **Estructura del proyecto**: Separación por carpetas lógicas (layouts, callbacks, components, utils) para facilitar el mantenimiento.
* **Exportación a PDF**: Uso de FPDF con personalización completa para el informe: encabezado, pie, tablas y gráficas centradas.

Este documento resume el funcionamiento técnico y las decisiones clave que sustentan el desarrollo de la app de análisis de baloncesto. Se ha priorizado la modularidad, la claridad visual y la eficiencia en la experiencia de usuario.

## **4. Muestra gráfica de funcionamiento**

* **Página de login:** En el que se introduce usuario y contraseña para acreditarse.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **Página de inicio**: Página que nos da la bienvenida, nos explica de que va la aplicación y nos da la posibilidad de entrar a las dos pestañas que tenemos: Performance y Médico. También se puede observar el menú superior en el que se puede pinchar para volver a inicio, cada una de las pestañas y cerrar sesión.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* Página de performance: Página que nos muestra el desempeño de un equipo de baloncesto durante un periodo de tiempo. Tiene filtros de temporada, de equipo, de fechas y de resultado, pudiendo mostrar las métricas de rendimiento y evolución del equipo en ese período o el del equipo solo en las victorias o derrotas. Por defecto están indicadas toda la temporada y todos los partidos tanto victorias como derrotas.

Las métricas utilizadas son el Rating Ofensivo (puntos anotados cada 100 posesiones), Rating Defensivo (puntos recibidos por 100 posesiones del rival), tiro de campo efectivo, porcentaje de rebote ofensivo, porcentaje de pérdidas y ratio de tiros libres que están consideradas como las variables que mejor pueden representar el desempeño.

Cómo gráficas se muestra la evolución temporal del equipo en la temporada o fechas indicadas, un gráfico donut que indique la distribución de posesiones del equipo, un radar para hacer una comparativa de estadísticas avanzadas entre las métricas del equipo y las que les producen los rivales y otro donut, en este caso que tienen los rivales cuando se enfrentan al equipo seleccionado.

* Página médica: