SPORTS DATA CAMPUS

MÁSTER EN PYTHON AVANZADO APLICADO AL DEPORTE

MÓDULO 9.Dash con Plotly, para crear aplicaciones de centralización de datos deportivos

SPORTS DATA CAMPUS

UCAM UNIVERSIDAD CATÓLICA DE MURCIA

ENIIT INNOVA IT BUSINESS SCHOOL

BIG DATA International Campus

Tabla de contenidos

TAREA: Dashboard Deportivo con Dash

Objetivo:

Desarrollar una aplicación web deportiva utilizando Dash que demuestre la capacidad de implementar visualizaciones interactivas, autenticación de usuarios y generación de reportes.

Instrucciones:

1. **Sistema de Autenticación (20 puntos)**
   - Implementar login con usuario: admin, contraseña: admin
   - Utilizar Flask-Login para gestión de sesiones
   - Implementar logout (cerrar sesión)
   - Proteger rutas que requieran autenticación
2. **Navegación y Layout (20 puntos)**
   - Crear un menú de navegación con tres páginas:
     - Home (página de bienvenida)
     - Dashboard de Performance
     - Dashboard de área no competitiva (elegir una: GPS, Médico, Antropometría, etc.)
   - Implementar CSS personalizado para el diseño
   - Utilizar Dash Bootstrap Components para estructura responsiva
   - El diseño debe ser consistente en todas las páginas

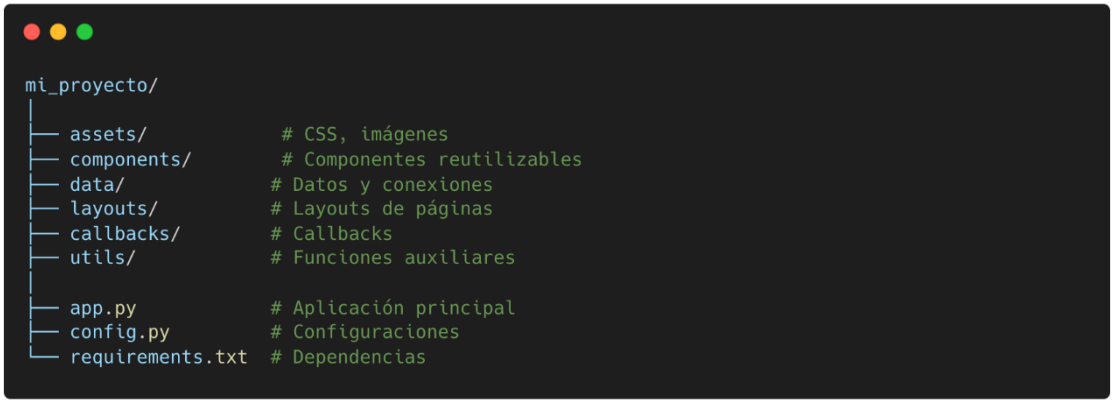
3. **Dashboards y Visualizaciones (25 puntos)**
   - Dashboard de Performance:
     - Al menos dos tipos diferentes de gráficos interactivos
     - Filtros de fecha y selección de jugadores/equipos
     - Botón para exportación a PDF con template personalizado
   - Dashboard de Área No Competitiva:
     - Al menos dos visualizaciones diferentes
     - Una tabla interactiva (usar DataTable o AG Grid)
     - Sistema de filtrado específico del área

Nota: las areas o departamento pueden ser: nutrición (antropometrias), medicina (lesiones), física (gps),administracion (contratos), etc.

4. **Interactividad y Datos (25 puntos)**
   - Implementar callbacks para actualización de gráficos
   - Conexión a al menos dos fuentes de datos diferentes
   - Manejo de errores y estados de carga
   - Implementar cache para optimizar consultas pesadas
   -
5. **Publicación y Documentación (10 puntos)**
   - Estructura de proyecto organizada
   - Archivo requirements.txt
   - README.md con instrucciones
   - Código comentado y siguiendo buenas prácticas

Formato de Entrega

1. **Estructura de Directorios**



```
mi_proyecto/
│
├── assets/            # CSS, imágenes
├── components/        # Componentes reutilizables
├── data/              # Datos y conexiones
├── layouts/           # Layouts de páginas
├── callbacks/         # Callbacks
├── utils/             # Funciones auxiliares
│
├── app.py             # Aplicación principal
├── config.py          # Configuraciones
└── requirements.txt   # Dependencias
```

2. **Documentación PDF que incluya:**
    - Capturas de pantalla de cada página
    - Explicación de funcionalidades implementadas
    - Desafíos encontrados y soluciones
    - Decisiones de diseño tomadas

Criterios de Evaluación

- Funcionalidad completa de autenticación y navegación
- Calidad y efectividad de las visualizaciones
- Implementación correcta de callbacks y manejo de datos
- Diseño atractivo y responsivo
- Calidad del código y documentación

Importante

- Usuario: **admin**
- Contraseña: **admin**
- El código debe estar bien organizado y comentado
- La aplicación debe ser responsiva y funcionar en diferentes dispositivos
- Al menos un dashboard debe incluir exportación a PDF
- Manejar adecuadamente errores y estados de carga

Nota: Asegúrese de documentar cada paso del proceso de desarrollo y justificar las decisiones tomadas en cuanto a diseño de la interfaz y manejo de datos. La claridad y efectividad en la comunicación de los datos son cruciales para esta tarea.