Estructura del Proyecto de Visualizador Médico 3D

Archivos Principales

```
prostate_analyzer/
├── main.py                   # Punto de entrada principal
├── config.py                 # Configuraciones y constantes
├── requirements.txt          # Dependencias del proyecto
├── app/
│   ├── __init__.py
│   ├── main_window.py        # Ventana principal de la aplicación
│   ├── models/
│   │   ├── __init__.py
│   │   ├── case_model.py     # Modelo para casos de estudio
│   │   └── prediction_model.py # Interfaz para modelo de predicción
│   ├── views/
│   │   ├── __init__.py
│   │   ├── viewer_widget.py  # Widget principal de visualización
│   │   ├── mpr_widget.py     # Widget para visualización multiplanar (MPR)
│   │   ├── volume_widget.py  # Widget para renderizado 3D
│   │   ├── case_panel.py     # Panel para gestión de casos
│   │   └── report_dialog.py  # Diálogo para generación de reportes
│   ├── controllers/
│   │   ├── __init__.py
│   │   ├── case_manager.py   # Gestión de casos y archivos
│   │   └── prediction_controller.py # Control de predicción
│   └── utils/
│       ├── __init__.py
│       ├── image_loader.py   # Utilidades para cargar imágenes
│       ├── vtk_utils.py      # Funciones auxiliares para VTK
│       └── report_generator.py # Generación de reportes
└── resources/
    ├── styles/
    │   └── dark_theme.qss    # Hoja de estilos para tema oscuro
    ├── icons/
    │   └── ... (varios iconos)
    └── models/
        └── ... (modelos pre-entrenados)
```

Dependencias Principales

- **PyQt5**: Para la interfaz gráfica
- **VTK**: Para visualización 3D y MPR
- **MONAI**: Para procesamiento de imágenes médicas
- **SimpleITK**: Para manejo de formatos de imágenes médicas
- **Torch**: Para la ejecución del modelo de predicción
- **NumPy**: Para procesamiento numérico
- **Matplotlib**: Para visualizaciones 2D adicionales
- **ReportLab**: Para generación de reportes PDF

Flujo de implementación recomendado

1. **Fase 1**: Implementación de la interfaz básica y la visualización 2D
    - Crear la ventana principal con tema oscuro
    - Implementar la carga de archivos NIFTI/DICOM
    - Crear widgets de visualización 2D para cortes axial, sagital y coronal
2. **Fase 2**: Integración de visualización 3D con VTK
    - Implementar widget de renderizado 3D
    - Añadir controles para manejo de volumen 3D
    - Integrar sincronización entre vistas 2D y 3D
3. **Fase 3**: Implementación del modelo de predicción
    - Integrar el modelo pre-entrenado
    - Añadir preprocesamiento de imágenes
    - Implementar visualización de resultados de segmentación
4. **Fase 4**: Generación de reportes y finalización
    - Crear sistema de reportes con capturas automáticas
    - Implementar exportación de resultados
    - Optimización de rendimiento y pruebas finales
5. **Fase 5**: Empaquetado y distribución
    - Configurar PyInstaller para crear ejecutables
    - Pruebas en diferentes sistemas operativos
    - Creación de instalador