
CleanCoders

CleanCoders
Documento de estándares
Versión 1.0

	Versión: 1.0
ISO/IEC/IEEE 29148	Fecha: 07/05/2025
Documento de Estándares	

Historial de Revisiones

Fecha	Versión	Descripción	Autor
09/05/2025	1.0	Definición de los estándares	CleanCoders
10/06/2025	1.0	Versiones de las herramientas utilizadas.	CleanCoders
11/06/2025	1.0	Revisión final previo a la presentación del proyecto.	CleanCoders

1 Introducción

Este documento proporciona una visión general completa de las tecnologías, marcos y herramientas utilizadas en la plataforma de gestión de yoga terapéutico TheraPose.

1.1. Tecnologías Backend

FastAPI Framework

TheraPose utiliza FastAPI como su marco web principal, sus características clave son:

Componente	Versión	Propósito
FastAPI	0.109.2	Marco web principal y servidor API
Uvicorn	0.27.1	Servidor ASGI para ejecutar aplicaciones FastAPI
Python-multipart	0.0.9	Manejo de datos de formularios y cargas de archivos
request	2.31.0	Cliente HTTP para llamadas API externas

	Versión: 1.0
ISO/IEC/IEEE 29148	Fecha: 07/05/2025
Documento de Estándares	

1.2. Tecnologías Frontend

Se emplea Jinja2(3.1.3) como motor de plantillas webs para Python, ofrece:

- Herencia en inclusión de plantillas.
- Creación de macros dentro de las plantillas.
- Integración natural con FastAPI.

Se emplea JavaScript debido a su alta popularidad en el desarrollo web:

- Manejo de eventos de interacción
- Manejo de tareas asíncronas sin bloquear el hilo principal.
- JavaScript y FastAPI se integran eficazmente, ya que se comunican a través HTTP Requests.

Se emplea Bootstrap por su simplicidad y eficiencia en el desarrollo web, ofrece:

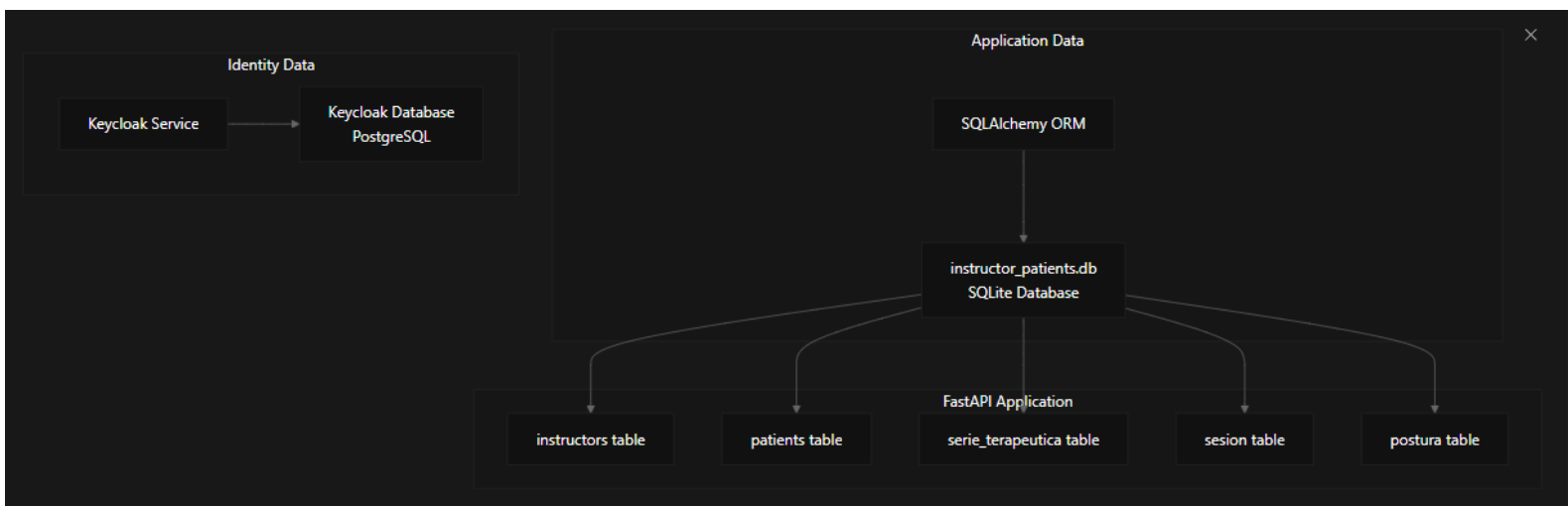
- Sistema de cuadrícula adaptable.
- Componentes prediseñados de UI.
- Amplios Plugins de JavaScript

1.3. Base de datos

1.3.1. Stack de la Base de datos para la aplicación principal

Componente	Versión	Rol
SQLite	3.42 +	Base de datos para la aplicación principal.
SQLAlchemy	2.0.27	ORM y kit de herramientas de base de datos
PostgreSQL	Latest	Base de datos de identidad keycloak

El sistema utiliza un enfoque de base de datos dual:



	Versión: 1.0
ISO/IEC/IEEE 29148	Fecha: 07/05/2025
Documento de Estándares	

1.4. Autenticación y Seguridad

1.4.1. Integración Keycloak

El sistema utiliza Keycloak como su principal solución de gestión de identidad y acceso:

Componente	Versión	Propósito
Keycloak	26.1.3	Proveedor de identidad y servidor de autenticación.
Python-keycloak	3.7.0	Biblioteca de Python para la integración de Keycloak
PostgreSQL	Latest	Base de datos para el backend de Keycloak.

1.5. DevOps e infraestructura

1.5.1. Contenerización

La aplicación utiliza Docker para la contenedorización y orquestación:

Tecnología	Versión	Propósito
Docker	24.0 +	Despliegue de contenedores
Docker Compose	2.22 +	Orquestación multi-contenedor
Imagen de Keycloak personalizada	bryanhert/keycloak-yoga: 26.1.3	Instancia Keycloak preconfigurada

1.6. Elementos gráficos y multimedia

La aplicación TheraPose organiza sus recursos multimedia en dos ubicaciones principales:

- Un directorio raíz llamado `images/`
Únicamente utilizado para el `README.md` de github.
- Y el directorio de recursos estáticos `static/`
Contiene todas las imágenes utilizadas en este proyecto.

La aplicación aprovecha formatos de imagen modernos y eficientes gracias a la herramienta gratuita de optimización de imágenes para sitios web “imgto.xyz” [1].

	Versión: 1.0
ISO/IEC/IEEE 29148	Fecha: 07/05/2025
Documento de Estándares	

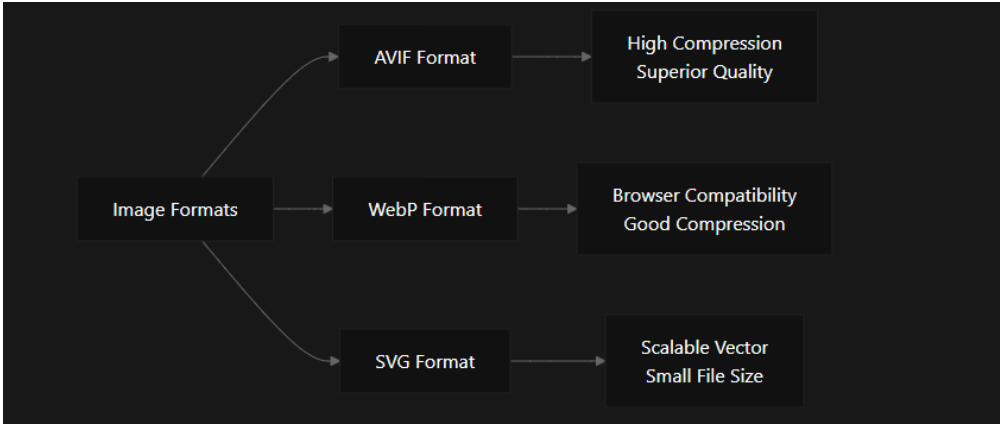


Ilustración 1. Formatos de imágenes utilizados en TheraPose_v1.0

1.7. Servicios Externos e integraciones

1.7.1. Servicios de Terceros

Servicio	Propósito	Método de integración
Youtube	Incrustación de contenido de video	Incrustación directa de URL
Formspree	Manejo de formularios de contacto	Envío de formulario HTML
GitHub	Repositorio de código fuente	Control de versiones y CI/CD

1.7.2. Dependencias Externas

El sistema se integra con servicios a través de diversos mecanismos:

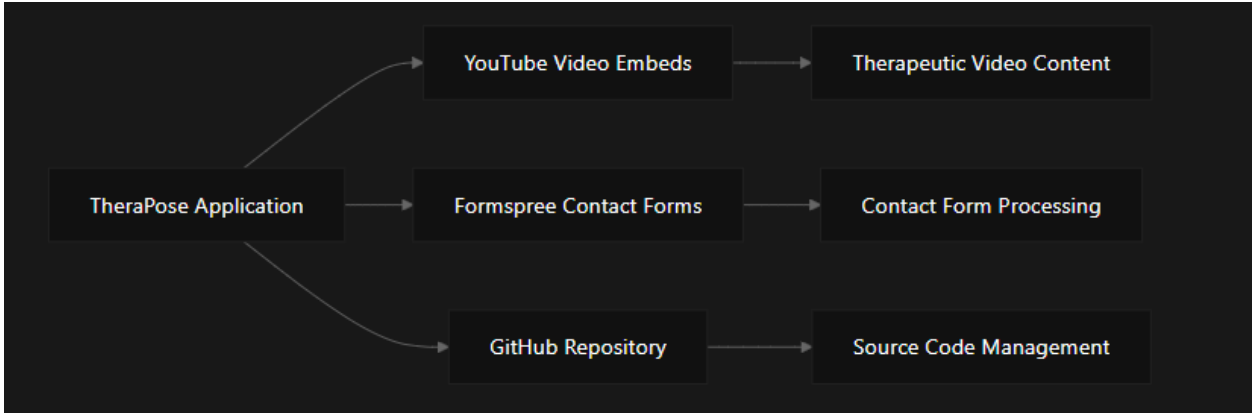


Ilustración 2. Dependencias externas y su integración.

	Versión: 1.0
ISO/IEC/IEEE 29148	Fecha: 07/05/2025
Documento de Estándares	

1.8. Dependencias de Desarrollo

1.8.1. Gestión de paquetes de Python

La lista completa de dependencias de Python se gestiona a través de requirements.txt:

```
fastapi==0.109.2
uvicorn==0.27.1
python-keycloak==3.7.0
python-dotenv==1.0.1
jinja2==3.1.3
python-multipart==0.0.9
requests==2.31.0
sqlalchemy==2.0.27
```

Bibliografía

[1] C. Fayock, “imgto.xyz: Free Image Optimization Toolkit,” imgto.xyz. [Online]. Available: <https://imgto.xyz/>. Accessed: Jun. 10, 2025.