



FRAMEWORKS



SOLID





SOLID

- **Descripción General:** Representa cinco principios de diseño de software orientado a objetos con el objetivo de crear código extensible, flexible, y mantenible. Facilita la escalabilidad y robustez del software permitiendo la adición ágil de nuevas funciones.
- **Área de Desarrollo:** Aborda aspectos fundamentales del diseño de software orientado a objetos, aplicable independientemente del lenguaje de programación.



- **Principios de Funcionamiento:**

- SRP: Principio de Responsabilidad Única.
- OCP: Principio de Abierto/Cerrado.
- LSP: Principio de Sustitución de Liskov.
- ISP: Principio de Segregación de Interfaces.
- DIP: Principio de Inversión de Dependencias.

- **Tipo de Framework:** No es un framework, sino un conjunto de principios de diseño para software orientado a objetos.

- **Desarrollo y Tecnologías:** No cubre el proceso de desarrollo completo, pero proporciona una guía para el diseño modular y mantenible del código.



Patrones de diseño y arquitectónicos que implementa Solid y en qué parte del mismo son implementados.

SOLID no implementa patrones de diseño o arquitecturas específicas, sino que establece principios fundamentales para crear diseños de software más sólidos.

Situaciones donde se recomienda su uso

En proyectos de software donde se oriente a objetos. Ya que esto hace que el código sea resistente a futuros cambios. Donde sus principios ayudan en los proyectos de largo plazo, también para sistemas complejos y entornos donde se prevén cambios frecuentes. Facilitando la prueba y el refactorizado del código.





Flask

FLASK





FLASK

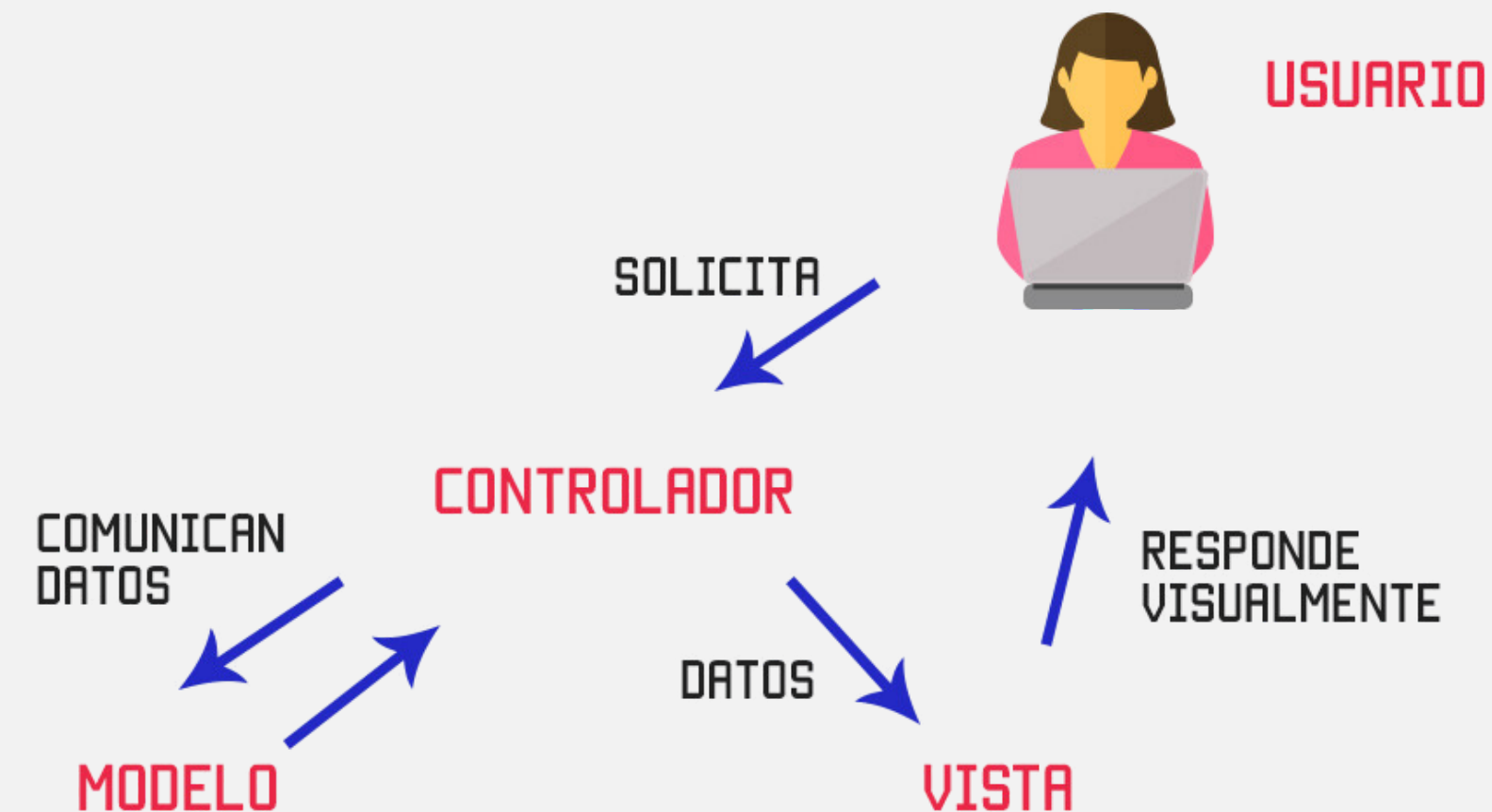
- **Descripción General:** Framework web en Python que ofrece flexibilidad y facilidad de uso manteniendo las operaciones de bajo nivel. Es ideal para prototipos rápidos y aplicaciones simples.
- **Área de Desarrollo:** Utilizado en una amplia gama de proyectos, especialmente para prototipos rápidos y aplicaciones web simples.
- **Principios de Funcionamiento:**
 - Basado en el principio WSGI.
 - Usa Jinja2 para el templating web.
- **Tipo de Framework:** Web framework en Python.





Patrones de diseño y arquitectónicos que implementa Solid y en qué parte del mismo son implementados.

Flask, también implementa el patrón de diseño Modelo-Vista-Controlador. Donde este patrón separa la lógica de la aplicación siendo el modelo -> , la vista -> la interfaz del usuario y el controlador -> manejo de solicitudes y respuestas.





Proceso de Desarrollo con Flask

- **Cobertura de Desarrollo:** Flask es un microframework para Python que es suficientemente flexible y ligero para el desarrollo de aplicaciones web tanto simples como complejas. Su diseño minimalista puede ser fácilmente extendido con numerosas extensiones disponibles.
- **Patrones de Diseño y Arquitectura:** Flask implementa el patrón Modelo-Vista-Controlador (MVC), facilitando la separación entre la lógica de aplicación, la interfaz de usuario, y el manejo de solicitudes y respuestas. Además, utiliza Jinja2 para la gestión de plantillas web, ampliando su versatilidad para el desarrollo de aplicaciones web personalizadas.





Situaciones Recomendadas para su Uso

1. Prototipado:

- Ideal para el prototipado rápido de aplicaciones web, permitiendo a los desarrolladores validar ideas y conceptos de manera eficiente.

2. Proyectos Pequeños o Medianos:

- Adecuado para proyectos que no requieren la complejidad de frameworks más robustos, ofreciendo una gran flexibilidad sin sobrecargar el desarrollo.

3. Servicios Web y APIs RESTful:

- Su facilidad de manejo de solicitudes HTTP y la compatibilidad con extensiones para desarrollo de APIs lo convierten en una opción sólida para servicios web y APIs RESTful.





COMPARACIÓN



Semejanzas	Diferencias
Tanto SOLID como Flask tienen como objetivo final mejorar la calidad, mantenibilidad y flexibilidad del código.	SOLID es un conjunto de principios de diseño, mientras que Flask es un framework web completo con una estructura y componentes predefinidos.
Con los principios de SOLID, se crean clases y componentes más coherentes y desacoplados, lo que facilita la reutilización de código en diferentes partes de la aplicación o en otros proyectos. De manera similar, Flask alienta la reutilización de código a través de extensiones y blueprints (módulos de aplicación), que permiten encapsular y reutilizar funcionalidades específicas.	SOLID aborda el diseño de software a un nivel más abstracto y genérico, aplicable a cualquier sistema orientado a objetos. Por otro lado, Flask se enfoca específicamente en el desarrollo de aplicaciones web.