**Tópicos Especiales de Ingeniería de Software**

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

**Arquitectura de Software**: La arquitectura de software de un sistema consta de un conjunto de estructuras que permiten analizar: el sistema, los componentes de software que lo integran, (las relaciones entre estos, y las características de ambos.

Ejemplos de estructuras incluyen:

Estructuras de tipo módulo: que representan elementos de código o unidades de datos con un comportamiento estático (como por ejemplo: clases, capas o modelos de datos).

Estructuras de tipo componente-y-conector: que representan elementos con un comportamiento en tiempo de ejecución (como: servicios, servidores, clientes o peers).

Estructuras de asignación: elementos que se relacionan con estructuras que no son de software (como: CPU, sistemas de archivos o redes).

Los términos frameworks verticales, horizontales y de soporte se utilizan comúnmente en el contexto del desarrollo de software y tecnología para describir diferentes tipos de arquitecturas y enfoques. Aquí te explico qué significa cada uno:

**Frameworks Verticales**

Un framework vertical está diseñado para abordar las necesidades específicas de una industria o área particular. Son soluciones muy especializadas que están centradas en un dominio específico y proporcionan herramientas y funcionalidades adaptadas a esa industria.

- Ejemplo: Un framework vertical para la atención médica podría incluir herramientas para manejar historias clínicas electrónicas, programación de citas, o gestión de seguros médicos.

- Ventajas: Ofrecen soluciones muy optimizadas para un sector, por lo que puedes obtener funcionalidades listas para usar sin tener que desarrollar componentes clave desde cero.

- Desventajas: Son menos flexibles fuera de su dominio y pueden requerir adaptaciones o integraciones complejas para otros sectores.

**Frameworks Horizontales**

Un framework horizontal es más general y está diseñado para ser utilizado en una variedad de industrias o aplicaciones. No está centrado en un dominio específico, sino que proporciona un conjunto de herramientas y bibliotecas que pueden ser aplicables a diferentes sectores.

- Ejemplo: Un framework como \*\*React o Angular para el desarrollo web se puede usar para crear aplicaciones de cualquier tipo (comerciales, educativas, de entretenimiento, etc.), no limitándose a un solo sector.

- Ventajas: Son más flexibles y reutilizables en múltiples contextos, permitiendo que un mismo marco de trabajo pueda ser adaptado para muchas necesidades diferentes.

- Desventajas: Al ser más generalistas, no proporcionan soluciones listas para un dominio específico, lo que puede significar más trabajo de personalización.

**Frameworks de Soporte**

Un framework de soporte se refiere a un conjunto de herramientas y bibliotecas que proporcionan funcionalidades adicionales que ayudan a los desarrolladores en tareas comunes o de bajo nivel, como la gestión de bases de datos, manejo de errores, o integración de sistemas.

- Ejemplo: Django ORM o Spring Boot (para Java) son frameworks de soporte porque manejan tareas como la gestión de bases de datos, autenticación, y seguridad sin necesidad de que los desarrolladores implementen esas funcionalidades desde cero.

- Ventajas: Ayudan a simplificar el trabajo de los desarrolladores y a reducir el tiempo de desarrollo, al cubrir tareas comunes o repetitivas.

- Desventajas: Pueden ser demasiado abstractos y ocultar la complejidad de ciertas operaciones, lo que puede dificultar la personalización en situaciones muy específicas.

En resumen:

-Verticales:Enfocados en un dominio específico (industria o sector).

- Horizontales: Más generales y aplicables a múltiples industrias.

- De soporte: Se centran en proporcionar herramientas y servicios adicionales que ayudan en tareas de desarrollo más técnicas.

Opinionated: los frameworks opinionated tienden a tomar decisiones por nosotros. Incluyen una estructura de directorios base donde deben ir ciertos archivos o capas de nuestra aplicación. Incluyen una arquitectura base pre-establecida. Incluyen librerías pre-instaladas y pre-configuradas para resolver problemas comunes (ej: sistemas de rutas, motores de templates, conexión a base de datos, manejo de datos en sesión, etc).

Unopinionated: los frameworks unopinionated son frameworks muy minimalistas, que incluyen unos cuantos elementos y librerías básicas para el desarrollo de aplicaciones. Por lo general, no definen una estructura de directorios base. El desarrollador puede implementar la arquitectura que desee, y si el desarrollador necesita componentes adicionales, deberá instalarlos e integrarlos manualmente.

Texto, Carta

El contenido generado por IA puede ser incorrecto.