

CONCEPTOS GENERALES Y REFERENCIAS

**Los sistemas de información Web
(SIW):**

Definición:

**Un Sistema de Información Web
(SIW) se refiere a un conjunto de
tecnologías y herramientas que
permiten la recopilación,
almacenamiento, procesamiento
y distribución de información a
través de la web. Estos sistemas
están diseñados para ser
accesibles desde cualquier
dispositivo conectado a Internet,**

facilitando la interacción entre usuarios y sistemas a través de interfaces web. Un SIW puede involucrar bases de datos, servidores web, aplicaciones cliente y servidor, y diversas herramientas de comunicación.

Características:

Accesibilidad Remota: Están diseñados para ser accesibles a través de la web, desde cualquier lugar con conexión a Internet.

Interactividad: Permiten la interacción de usuarios a través

de formularios, interfaces dinámicas, y la generación de contenido en tiempo real.

Escalabilidad: Pueden adaptarse a un número creciente de usuarios y manejar grandes volúmenes de datos.

Multi-plataforma: Funcionan en diferentes sistemas operativos y dispositivos, desde computadoras de escritorio hasta smartphones y tablets.

Integración de datos: Facilitan la integración con otros sistemas a través de APIs y servicios web.

Actualización en tiempo real:
Permiten actualizaciones inmediatas de la información, sin necesidad de recargar la página.

Ejemplos de aplicaciones prácticas:

E-commerce: Plataformas como Amazon y MercadoLibre que permiten la compra y venta de productos a través de la web.

Redes sociales: Facebook, Twitter y LinkedIn, que gestionan interacciones entre usuarios y la distribución de contenido.

Aplicaciones de gestión empresarial (ERP): SAP, Oracle ERP, que permiten a las empresas gestionar su inventario, ventas, finanzas, etc., a través de interfaces web.

Plataformas de educación en línea: Moodle, Coursera, que ofrecen cursos y formación a distancia.

"Software Engineering" de Ian Sommerville.

Apartado 6.2. Modelado conceptual:

6.2.1. Modelos Independientes de Computación (CIM):

Los Modelos Independientes de Computación (CIM, por sus siglas en inglés Computation Independent Models) son una parte fundamental en la ingeniería del software y los sistemas de información, especialmente en metodologías como MDA (Model Driven Architecture) y en el diseño de arquitecturas orientadas a servicios.

Definición:

El CIM se enfoca en la representación del sistema en un nivel de abstracción muy alto, sin preocuparse de los detalles técnicos de implementación o computación. Este tipo de modelo describe el comportamiento y las funciones del sistema desde el punto de vista de los usuarios y las necesidades del negocio, sin entrar en detalles sobre cómo se realizará técnicamente. Es una representación de los requisitos y la visión general del sistema.

Características:

Independencia de la computación: El modelo no depende de tecnologías o plataformas de hardware específicas.

Enfoque en el negocio: El CIM se centra en los objetivos y necesidades del negocio, y describe cómo el sistema debe satisfacerlos.

Abstracción alta: No entra en detalles sobre las implementaciones tecnológicas o

de programación, sino que se limita a representar lo que se necesita hacer.

Utilización para la comunicación: Este tipo de modelo es útil para comunicar entre analistas, clientes y otras partes interesadas sin necesidad de entender los detalles técnicos.

Facilita el diseño: Proporciona una base sólida sobre la cual se desarrollarán modelos más detallados, como el modelo independiente de plataforma (PIM) o el modelo independiente de tecnología (CIM).

Ejemplo: Supongamos que una empresa desea implementar un sistema de gestión de pedidos. Un CIM podría describir cómo los usuarios deben realizar pedidos, qué datos se deben capturar (como el cliente, el producto y la cantidad), y las reglas de negocio que deben aplicarse (como descuentos por volumen). Sin embargo, no especificaría si este sistema debe estar en la nube, en un servidor local o qué lenguaje de programación se utilizará.

El modelo CIM es crucial para asegurarse de que todas las partes interesadas comprendan el sistema de manera coherente antes de que se inicie el proceso de diseño y desarrollo técnico.

Model-Driven Software Engineering in Practice" de Marco Brambilla, Jordi Cabot, y Manuel Wimmer.