

# Tema-3-Analisis-y-diseno-de-sist...



**juanfrandm98**



**Sistemas de Informacion Basados en Web**



**3º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**



**Que no te escriban poemas de amor  
cuando terminen la carrera**



*(a nosotros por  
suerte nos pasa)*

**WUOLAH**

Que no te escriban poemas de amor  
cuando terminen la carrera ▶▶▶▶▶▶▶▶▶▶



WUOLAH

(a nosotros por suerte nos pasa)

No si antes decirte  
Lo mucho que te voy a recordar

Pero me voy a graduar.  
Mañana mi diploma y título he de  
pagar

Llegó mi momento de despedirte  
Tras años en los que has estado mi  
lado.

Siempre me has ayudado  
Cuando por exámenes me he  
agobiado

Oh Wuolah wuolah  
Tu que eres tan bonita

## Tema 3. Análisis y diseño de sistemas web

- Conocer el marco legal aplicable a los Sistemas de Información Basados en Web.
- Comprender la necesidad del seguimiento de las recomendaciones y normativas de accesibilidad y usabilidad.

### 3.1. Ingeniería de Requisitos

La Ingeniería de Requisitos (IR) es el conjunto de actividades que llevan a la especificación de las necesidades de usuarios y departamentos interesados en la producción del sistema web, así como las restricciones que recaen sobre el mismo. Normalmente, un sistema web tendrá que interaccionar con otros sistemas o influirá en la manera en que trabajarán los empleados, departamentos...

Leyes no escritas de la IR:

- La especificación de requisitos es un problema de comunicación.
- Los interesados y el equipo de desarrollo no escriben correctamente lo que entienden como requisitos.
- La validación de los requisitos se produce tarde.
- El usuario ve el producto final tarde.
- Los requisitos suelen ser inestables y variables, lo que es muy malo para el desarrollador.

Los requisitos funcionales son aquellos que determinan qué debe hacer el software, los servicios que debe prestar y a qué datos debe reaccionar. Por ejemplo, los elementos de entrada que tendrá el interfaz web, qué funcionalidad tendrá o cómo se vinculan las páginas de información entre sí. Pueden referirse a:

- Requisitos de la organización: diferentes puntos de vista de la organización o entorno donde se implantará la solución.
- Requisitos del dominio de la aplicación: funcionalidad en sí de la aplicación - contenido de la aplicación, flujo de información, estructura de la información.
- Requisitos de navegación: cómo se pasa de entre distintos elementos de información.
- Requisitos de interacción: lo relativo a la interfaz de usuario.

Los requisitos no funcionales son restricciones o condiciones que se imponen al sistema que no tienen que ver con la funcionalidad. También son llamados atributos de calidad. Por ejemplo:

- Requisitos del producto: memoria que se puede usar o plataforma sobre la que debe correr, navegadores soportados, resoluciones de pantalla...
- Requisitos organizacionales: impuestos por el desarrollador, como por ejemplo lenguajes a usar, estándares de desarrollo, metodologías...
- Requisitos externos: normativas legales aplicables, interacción con sistemas externos...

Existen distintos mecanismos para la extracción de requisitos:

- Estudio de viabilidad: indicado para sistemas novedosos en los que las posibilidades de fracaso son más altas. Se trata de un estudio previo que recoge las especificaciones, funcionalidades y presupuesto. Es imprescindible contar con un especialista que conozca el negocio (clientes, canales de distribución, flujo de interacción, interacción cliente-servicio...).

WUOLAH

- Entrevistas: son fundamentales y no deben hacerse sólo al jefe, sino a personas de todos los estamentos, incluso a los clientes. Hay que prepararlas previamente y las mejores preguntas son las abiertas.
- Desarrollo conjunto de aplicaciones: sesiones conjuntas entre usuarios y analistas. Se trata de obtener requisitos conjuntamente aprovechando las dinámicas de grupo.
- Observación directa de la etnografía: observar un sistema funcionando para entender cómo se llevan a cabo las tareas en la actualidad. Se analiza no solo el programa, sino la interacción con las personas que lo usan y las interacciones entre las personas. Hay que dejar constancia de que no es una evaluación del personal para que el trabajo se desarrolle normalmente.

Una vez obtenida la información hay que clasificar, ordenar, priorizar, documentar y especificar los requisitos en un documento de requisitos. No es una tarea trivial y lleva tiempo, pero es importante.

Es fundamental negociar y validar los requisitos con el cliente. Normalmente querrá que se satisfagan todos, pero a un coste limitado. Puede ser necesaria la firma de un contrato con los mismos, resaltando su importancia y concretando exactamente el alcance del proyecto. Hay que evitar malentendidos por ambas partes y puede ser interesante ofrecer información sobre el precio y obligatoriedad o no de cada requisito.

Puede haber dificultades en la extracción de requisitos como:

- Diferenciar políticas de empresa con requisitos.
- Diferenciar entre requisito y opciones de implementación.
- Límites difusos del sistema: el cliente irá añadiendo requisitos según describe el problema.
- Diferencias de dominio entre el cliente y el desarrollador: cada uno es experto en su ámbito, lo que dificulta su comprensión.
- Volatilidad: los requisitos cambian y hay que aprender a gestionarlo.
- Problemas no tecnológicos: cambios de interlocutor, políticas de empresa, de prioridades...

### 3.2. Diseño de Aplicaciones Web

Una vez realizada la especificación de requisitos hay que refinar las abstracciones identificadas en la fase previa para organizar los datos, la navegación, la presentación y la arquitectura de la aplicación.

Las aplicaciones web tienen una serie de peculiaridades:

- Mayor accesibilidad de la información y servicios al poder haber muchos usuarios simultáneos.
- Interfaz orientada al documento.
- Variedad de tecnologías de gestión, acceso y procesamiento de datos.
- Variedad de tecnologías y motores de visualización.
- Arquitectura compleja (sistema distribuido, balanceo de carga...).

Mediante un buen diseño del flujo de trabajo podemos organizar las diferentes etapas de un proyecto minimizando los errores y manteniendo siempre el foco en el aspecto, funcionalidades y experiencia de uso de nuestro sitio web. Usaremos herramientas conocidas de ingeniería del software:

- Diagramas de flujo UML: describe paso a paso las operaciones de los componentes de un sistema.
- Diagramas de secuencia UML: describe las líneas de vida de los objetos y los mensajes intercambiados entre ellos para ejecutar una función.

El diseño de datos se hará a partir de diagramas de clases UML, que describe la estructura de un sistema mostrando las clases, sus atributos, operaciones y relaciones entre objetos. En el caso de bases de datos relacionales, la conversión es directa.

El diseño de navegación consiste en estructurar las rutas de navegación a través de la información y servicios ofrecidos por nuestro sistema de información web. Son fundamentales la estructura del sitio y el comportamiento del usuario al navegar (acciones del usuario y eventos que desencadena).

Para diseñar la estructura del sitio debemos distinguir los siguientes componentes:

- Ítems atómicos: elementos de información con instancias de entidades de datos.
- Ítems compuestos: estructura compuesta de varios ítems atómicos.
- Estructuras contextuales: estructuras de navegación para acceder a los ítems (menús, índices, metaetiquetas...).

IFML (*Interaction Flow Modeling Language*) es el estándar para llevar a cabo el diseño estructural de aplicaciones. Siguiendo el esquema Modelo, Vista, Controlador fundamentalmente se encarga de la parte de la Vista, pero está claro que depende y hace referencia al modelo y al controlador.

En el modelado con IFML abordamos las siguientes perspectivas:

- Especificación de la estructura de la vista (contenedores, relaciones entre ellos...).
- Especificación del contenido de la vista.
- Especificación de eventos (de los que pueden afectar al estado del interfaz).
- Especificación de transición de eventos (efectos de los eventos sobre la interfaz).
- Especificación de los parámetros de conexión (dependencias de entrada/salida entre los componentes de la vista y las acciones).

Los diagramas IFML consisten en uno o más contenedores de vista de alto nivel. Internamente pueden estructurarse como una jerarquía de subcontenedores que pueden contener componentes de vista (contenido o elementos de entrada de información). Estos componentes tienen parámetros de entrada y salida. Los contenedores y componentes se pueden asociar con eventos (indica la interacción con el usuario).

Es necesario también un diseño de adaptación, puesto que las aplicaciones web serán usadas posiblemente desde diversas partes del mundo y bajo distintos dispositivos. Normalmente, la adaptabilidad se refiere a tres características:

- Localización e internacionalización: idioma, moneda, hora, fechas, aspectos culturales... En PHP hay varias posibilidades:
  - Crear varias versiones de las páginas con los textos traducidos (NO).
  - Usar ficheros de texto con las traducciones y cargar el oportuno.
  - Utilizar la librería gettext.
  - No reinventar la rueda y usar un motor de plantillas que soporte la internacionalización: i18n extension for Twig.
- Personalización y adaptación: ajustes para cada uno de los usuarios mediante adaptación de contenidos, acciones de navegación automática, adaptación de la estructura de navegación y adaptación del layout.
- Accesibilidad: ajuste para personas con discapacidades.