

## Patrimonio Cultural



Desarrollo de Herramientas  
de Administración y Acceso

## Objetivos del curso

- Adquirir visión generalizada de las tecnologías de desarrollo utilizadas en Sistemas de gestión del Patrimonio Cultural.
- Política de acceso público.
- Interfaces de acceso a bases de datos.
- Especificaciones de Administración.
- El SIPCA como ejemplo de Sistema de Gestión de Patrimonio Cultural.

2

## Política de Acceso Público

- La **información** almacenada ha de ser **accesible**.
- **Conservación** del patrimonio cultural.
- **Difusión** del patrimonio cultural.
- Las nuevas tecnologías de información como herramienta al servicio del Patrimonio Cultural.
- **Interfaces de acceso** a bases de datos.

3

## Política de Acceso Público

- **Ley de Protección de Datos**
  - Ley Orgánica 15/1999, de 13 de Diciembre, de Protección de Datos de Carácter Personal.
  - En todo momento se garantizará y protegerá, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

4

## Interfaces de acceso a base de datos

- **Consideraciones previas:**
  - Expresar claramente nuestras necesidades. (Que sea rápida, segura, normalizada, etc)
  - El programador o empresa que desarrolla la aplicación no tiene por qué saber nada sobre Patrimonio Cultural.
  - Elección de software, entornos y lenguajes de programación.
  - En definitiva, saber qué se nos pretende vender.

5

## Interfaces de acceso a base de datos

- **Interfaces "de escritorio"**
  - La principal característica de este tipo de interfaces es que serán programas que tendrán que ser **instalados** en nuestros PC (**FileMaker**).
  - Desarrollo de alto coste
  - Relegadas a un segundo plano tras la llegada de las **Interfaces WEB**.

6

## Interfaces de acceso a base de datos

### ■ Interfaces web

- Construidas utilizando páginas WEB.
- Documentos de texto a los que se les añaden etiquetas que nos permiten visualizar el texto en distintas formas y establecer enlaces entre unas páginas y otras.
- Se idea el formato HTML (HyperText Markup Language) como estándar para representar estos documentos. (<http://www.w3c.org>)



7

## Interfaces de acceso a base de datos

### ■ Interfaces web

- Navegadores web como frontal **universal** para la presentación de información.
- Dicha información podrá encontrarse almacenada en **cualquier parte** del mundo y ser accesible.
- Acceso a **datos actualizados en tiempo real**.
- **Costes** de desarrollo **mas bajos**.

8

## Interfaces de acceso a base de datos

### ■ Data Warehousing

- Surgen como respuesta a la problemática de extraer información sintética a partir de datos atómicos almacenados en bases de datos.
- La información almacenada en base de datos ya no se trata como entidades independientes, sino que surge el concepto de **grupo**.
- Por ejemplo, podremos realizar operaciones sobre grupos de registros de bienes culturales resultantes de una búsqueda. Estas operaciones afectarán al grupo.

9

## Interfaces de acceso a base de datos

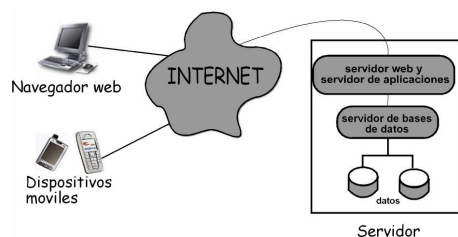
### ■ Interfaces web

- El navegador realiza una petición, consulta de datos.
- Esta petición es pasada al servidor web o al servidor de aplicaciones.
- El servidor de aplicaciones procesa dicha petición y la pasa al servidor de base de datos que será el encargado de realizar la consulta a la base de datos.
- El servidor de base de datos devuelve el resultado de ejecutar dicha petición al servidor de aplicaciones.

10

## Interfaces de acceso a base de datos

### ■ Interfaces WEB



11

## Aplicaciones WEB

### ■ Aplicaciones web estáticas

- Es necesario modificar el código de la página para que el contenido cambie.
- Útiles cuando la información mostrada cambie pocas veces.
- Se pueden desarrollar herramientas que generen páginas web estáticas.
- Fáciles de desarrollar.

12

## Aplicaciones WEB

### ■ Aplicaciones web dinámicas

- El contenido de la página web cambiará en función del usuario que la visite y de la tarea que quiera realizar.
- Podremos filtrar, ordenar y presentar la información de la forma más adecuada en función de las necesidades de cada momento.
- Nos permiten mostrar ofrecer la información mas actual.
- Desarrollo mas complejo.

13

## Software libre vs propietario

### ■ Software propietario (o "no libre")

- Algún individuo o compañía retiene los derechos de autor sobre dicho software.
- No se tiene acceso al código fuente de dicho software.
- Se niega el derecho a copiarlo, modificarlo o estudiarlo.

14

## Software libre vs propietario

### ■ Software libre

- Se trata de un concepto de "libertad" no de precio. No tiene por qué ser gratis.
- Libertad de usar el programa.
- Libertad de estudiar cómo funciona. Acceso al código fuente
- Libertad de distribuir copias.
- Libertad de mejorar el programa y hacer públicas las mejoras a los demás.
- <http://www.gnu.org/philosophy/free-sw.es.html>

15

## Software libre vs propietario

### ■ Software comercial

- Se trata de software libre.
- Las compañías que lo producen cobran dinero por el producto.
- Ejemplos de software comercial:
  - Varias distribuciones de Linux: Suse, Red hat, etc.

16

## Software libre vs propietario

### ■ Conclusión

- Ventajas del uso del software libre frente al uso de software propietario.  
<http://www.opensource.org>  
<http://sourceforge.net/>
- Evolución hacia la implantación de software libre en detrimento del software comercial.  
(Linux frente a Windows)

17

## Aplicaciones WEB

### ■ Servidor web

- Programa que implementa el protocolo HTTP (Protocolo de transferencia de Hipertexto).
- Recibe las peticiones HTTP llevadas a cabo por el Navegador.
- Protocolo diseñado para transferir páginas web o páginas HTML
- Páginas HTML: textos complejos con enlaces, imágenes, botones, etc.

18

## Aplicaciones WEB

### ■ Servidor web

- Ejemplos de servidores web importantes:
  - **Apache**. Servidor de código abierto más usado en la actualidad. Gratuito  
<http://www.apache.org>
  - **IIS** (Internet Information Server). Propiedad de Microsoft Corporation.  
Incorporado en las últimas versiones de Windows. De pago.

19

## Aplicaciones WEB

### ■ Servidor de aplicaciones

- En función del lenguaje de programación empleado.
- Independiente tanto de la interfaz de usuario como de la base de datos.
- Ejecuta la lógica de negocio de la aplicación.
- Intérprete del lenguaje de programación utilizado.

20

## Aplicaciones WEB

### ■ Servidor de aplicaciones

- Ejemplos de servidores de aplicaciones:
  - Tomcat. Servidor de aplicaciones gratuito.  
[jakarta.apache.org/tomcat/](http://jakarta.apache.org/tomcat/)
  - iPlanet.
  - WebLogic ([www.bea.com](http://www.bea.com))
  - Resin ([www.caucho.com](http://www.caucho.com)) Gratis si es de uso personal. De pago si el uso es comercial.

21

## Aplicaciones WEB

### ■ Lenguajes de programación

- Existen multitud de lenguajes de programación, tanto de cliente como de servidor.
- Nos interesan aquellos utilizados en la generación de páginas web dinámicas.
- La generación dinámica de contenidos requiere que el servidor realice algún tipo de procesamiento adicional sobre la petición HTTP iniciada por el cliente, con el fin de generar una respuesta personalizada y adaptada al usuario.

22

## Aplicaciones WEB

### ■ Lenguajes de programación

- Java (Sun Microsystems). [www.sun.com](http://www.sun.com)
- .Net (Microsoft). [www.microsoft.es](http://www.microsoft.es)
- Php, XML, etc.
- Java es un lenguaje robusto ya que realiza verificaciones en busca de errores.
- Lenguaje seguro.
- Portable. Un programa escrito en Java puede ser ejecutado en diferentes entornos.

23

## Aplicaciones WEB

### ■ Servidores de bases de datos

- **Oracle**: Es el mayor y más usado Sistema Manejador de Base de Datos Relacional (RDBMS) en el mundo. Alto coste
- **SQL Server**: Potente servidor de base de datos. Generalmente utilizado en aplicaciones que utilizan tecnología Microsoft (Alto coste). Utilizado en Domus (Software de gestión de museos utilizado por la DGA).
- **MySQL**: Alta velocidad y gratuito ([www.mysql.com](http://www.mysql.com))
- **PostgreSQL**: Más potente que MySQL pero más lento

24

## Aplicaciones WEB

### ■ Conclusiones

- Siempre o casi siempre dispondremos de software gratuito o software libre, lo cual disminuirá los costes de desarrollo.
- Gran abanico de combinaciones entre Sistemas Operativos, Servidores de Aplicaciones, Servidores de Base de Datos y Lenguajes de Programación.
- Decidir siempre en función de nuestras necesidades no en función del presupuesto.

25

## Especificaciones de Administración

- Ejemplo pliego de prescripciones técnicas para el desarrollo de software de gestión documental para la Administración.

### 3. Entornos de desarrollo y explotación

		Imprescindible	
1	Base de datos	INFORMIX/POSTGRES QL	SI
2	Servidor web:	Apache, preferentemente.	SI
3	Servidor de aplicaciones:	Tomcat 1.4.x.	SI
4	Sistema Operativo	GNU LINUX	SI
<small>Si la solución propuesta difiere sustancialmente de las prescripciones...</small>			SI

26

## Especificaciones de Administración

### 4. Funcionalidades

		Imprescindible	
3	4.1 Controles y seguridad		Imprescindible
<b>USUARIOS DEL SISTEMA</b>			
11	Permite que el administrador restrinja el acceso a la información a determinados usuarios o grupos de usuarios.		SI
<small>Permite que el administrador asigne al perfil del usuario ciertas atribuciones para una variedad de operaciones, funciones y acciones relacionadas.</small>			
<b>COPIAS DE SEGURIDAD</b>			
31	Está dotado de procedimientos automáticos de copia de seguridad y restauración que permitan realizar copias periódicas de seguridad de datos y los atributos administrativos del repositorio del SISTEMA o de algunos de ellos.		SI

27

## Especificaciones de Administración

### 4.2 Funciones administrativas

		Imprescindible	
39	Permite que los administradores, de forma controlada y sin ningún esfuerzo innecesario, recuperen, visualicen y reconfiguren parámetros del SISTEMA y opciones escogidas en el momento de la configuración -por ejemplo, los elementos que se indexarán- así como la reasignación de usuarios y funciones a los perfiles de usuarios.		SI
40	Incluye instrumentos de copia de seguridad y recursos que permitan restaurar el SISTEMA a partir de tales copias y de la pista de auditoría, sin menoscabo de la integridad del SISTEMA.		
41	Incluye instrumentos de recuperación y restauración en previsión de posibles fallos del SISTEMA o de errores en la actualización. Asimismo notifica a los administradores el resultado de la operación.		

28

## Especificaciones de Administración

### 4.5 Autoridades y tesauros

		Imprescindible
98	Formato de autoridades aplicando las normas internacionales: segunda edición de la norma ISAAR(CPF)	SI
99	Recursos de elaboración de tesauros y relaciones normalizadas de descriptores, tanto asociativas como jerárquicas.	SI
100	Importación y exportación de autoridades de acuerdo con la norma ISAAR(CPF) y con el formato EAC.	SI

29

## Especificaciones de Administración

### REQUISITOS DE CARACTER NORMATIVO Y LEGISLATIVO

		Imprescindible	
204	Se atiene a la legislación aplicable en materia de gestión de documentos de archivo.		
205	No incluye ninguna característica incompatible con la legislación en materia de protección de datos o de otro tipo.		
<b>CONSERVACIÓN A LARGO PLAZO Y OBSOLESCENCIA TECNOLÓGICA</b>			
206	Permite la conversión en masa de los datos, sus relaciones, las pistas de auditoría y los propios documentos multimedia o imágenes asociadas a otros soportes o SISTEMAS acordes con las normas sobre el formato o los formatos en uso correspondientes.		SI
<small>La especificación utiliza normas más modernas de representación general y...</small>			

30

## Especificaciones de Administración

### ■ Ofertas - Condiciones Iniciales.

- Desarrollo de aplicación de gestión de centro documental.
- Distintos perfiles de usuarios (Redactores de tesauros, Administradores, Gestores de datos, etc).
- Accesibilidad vía web (Internet/Intranet).
- Implementación de estándares.
- Migración de datos.
- PC Pentium 4, ADSL 1 Mb.
- 5 Usuarios simultáneos.

31

## El SIPCA como ejemplo

### ■ Características principales del SIPCA

- Sistema de Información del Patrimonio Cultural. Desarrollado por el CIPCA (Centro de Información del Patrimonio Cultural Aragonés).
- Permite el trabajo en **entorno distribuido**.
- Los componentes de ese entorno distribuido serán un **nodo central** encargado de recoger todos los datos de los **nodos secundarios** los cuales tendrán modelos de datos compatibles pero no iguales que el **nodo central**.

32

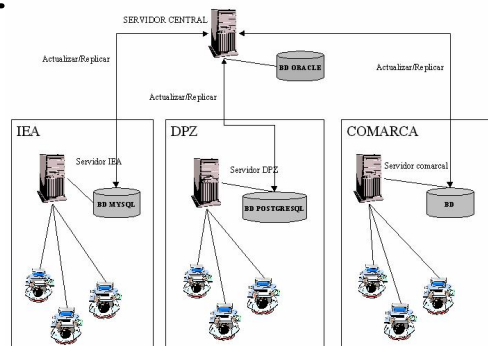
## El SIPCA como ejemplo

### ■ Características principales del SIPCA

- Compatible con **cualquier Sistema Operativo**.
- Servidor Web **Apache**.
- Servidor de aplicaciones **Tomcat**.
- Compatible con distintos servidores de bases de datos (**Oracle, MySQL**, etc).
- Tecnologías aplicadas en el sistema: **Servicios WEB, XML** y lenguaje de programación **Java**.

33

## El SIPCA como ejemplo



## El SIPCA como ejemplo

### ■ Servicios Web

- Técnica que permite a los servicios utilizados por distintos sistemas operativos y que se ejecutan en distintas plataformas "hablar entre ellos".
- Disponibilidad abierta al público. Serán altamente **accesibles** debido a la facilidad y uniformidad de acceso al basarse en estándares.
- Al ser accesibles a todo el mundo, suponen una clara ventaja comercial pero ¿qué ocurre con la seguridad?

35

## El SIPCA como ejemplo

### ■ Servicios Web - Seguridad

- Las técnicas de seguridad tradicionales como cortafuegos (que actúan filtrando puertos y direcciones IP's) no servirán en este caso ya que la información viaja "en claro" a través de internet.
- Necesitamos cortafuegos que actúen **a nivel de aplicación** no a nivel de red. Estos serán los **cortafuegos de aplicaciones XML**.
- Son capaces de comprender los servicios web, las peticiones de servicio y el contenido de los mensajes XML intercambiados.

36

## El SIPCA como ejemplo

### ■ Servicios Web - XML

- XML (Extensible Markup Language) como lenguaje estándar de intercambio de información.
- Sintaxis creada a partir de un **estándar internacional** de procesamiento de texto existente, el Lenguaje Generalizado Estándar de Marcas (Standard Generalized Markup Language, ISO 8879:1986(E)).
- <http://www.w3.org/XML>

37

## El SIPCA como ejemplo

### ■ Servicios Web – XML

- Se trata de un lenguaje de etiquetas con una **estructura jerárquica**. Un fichero de texto en el que introduciremos datos de forma estructurada.
- Es un **lenguaje dinámico**, permite que cada desarrollador defina estas etiquetas como quiera.
- La **extensibilidad y flexibilidad** de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.

38

## El SIPCA como ejemplo

### ■ Servicios Web - XML

- **Un ejemplo de XML**

```
<pais nombre="España">
  <comunidadautonoma nombre="Aragón">
    <provincia>Huesca</provincia>
    <provincia>...</provincia>
  </comunidadautonoma >
  <comunidadautonoma nombre="Andalucía">
    <provincia>Jaén</provincia>
    <provincia>...</provincia>
  </comunidadautonoma >
</pais>
```

39

## El SIPCA como ejemplo

### ■ Servicios Web - XML

- Para que otros sistemas comprendan la estructura de nuestro XML, tendremos que definir dicha estructura en lo que se conoce como **DTD (Document Type Definitions)**.
- El propósito del DTD es **definir la estructura** de los bloques de nuestro XML de manera que pueda ser interpretado por cualquier aplicación.
- Puede definirse dentro del documento XML o fuera, en un archivo externo.

40

## El SIPCA como ejemplo

### ■ Comunicación

- Dos tipos de nodos, un **nodo central** y **nodos distribuidos** que pueden trabajar de forma independiente.
- Nodo central **encargado de administrar todos los datos** asociados al Patrimonio Cultural Aragonés.
- Nodos distribuidos podrán tener acceso a toda la información o a información determinada.

41

## El SIPCA como ejemplo

### ■ Procesos

- Todas las modificaciones, inserciones realizadas dentro del nodo central han de ser enviadas a los nodos distribuidos.
- Estas modificaciones no necesariamente han de ser enviadas en tiempo real.
- **Proceso de sincronización**: Proceso mediante el cual los nodos, tanto central como distribuidos, mantienen la información actualizada.
- Podrá ser **automático** o **manual**.

42

## El SIPCA como ejemplo

### ■ Proceso de sincronización automático

- Se pueden fijar intervalos de tiempo determinados para el proceso de sincronización. Al final del proceso todos los nodos tendrán la información actualizada.



43

## El SIPCA como ejemplo

### ■ Proceso de sincronización manual

- En cualquier momento, cualquier nodo podrá actualizar los datos que ha modificado sobre el servidor central.
- También recogerá los datos que han cambiado del servidor central desde el momento en que se realizó por última vez este proceso.

44

## El SIPCA como ejemplo

### ■ Operaciones

- Se trata de un proceso **bidireccional** en el que se distinguen dos tipos de operaciones.
- **Actualización y replicación.**
- **Actualización:** Envío al nodo únicamente de los datos que han cambiado desde una determinada fecha.
- **Replicación:** Envío de toda la información al nodo

45

## El SIPCA como ejemplo

### ■ Elementos

- Para que la comunicación entre nodos se realice de forma correcta definimos dos elementos principales:
  - **Política de resolución de conflictos:** Permitirá evitar inconsistencias entre datos de nodos secundarios y el nodo central.
  - **Modelo de Intercambio de Datos:** Tenemos en cuenta que el modelo de datos de los nodos distribuidos puede variar. Se define un modelo central que entienda todos los nodos.

46

## El SIPCA como ejemplo

### ■ Control de usuarios

- Todo sistema que quiera asegurar una cierta fiabilidad en su uso, debe proporcionar un control de usuarios.
- Debido a la distinta naturaleza de los nodos y de los datos de cada nodo, se definen usuarios con distintos privilegios (Consulta y modificación).

47

## El SIPCA como ejemplo

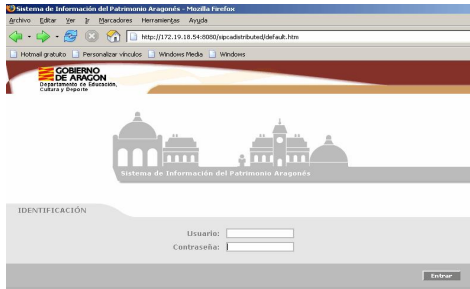
### ■ Registro de metadatos

- Es necesario que el sistema ofrezca un mecanismo que asegure que va a existir un control sobre los cambios que se realicen en la información.
- Esto implica recoger información sobre los datos o elementos a los que afecta la modificación.
- Para facilitar esto, se facilita un modelo de datos que recoge estos metadatos en cada uno de los nodos del sistema.

48



## El SIPCA como ejemplo



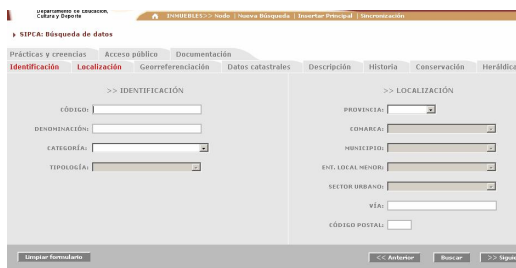
49

## El SIPCA como ejemplo



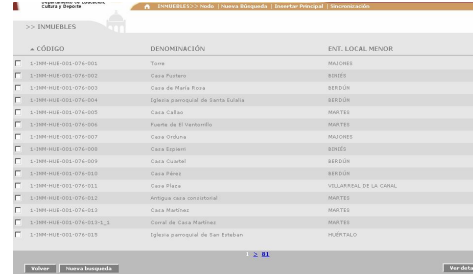
50

## El SIPCA como ejemplo



51

## El SIPCA como ejemplo



52

## El SIPCA como ejemplo



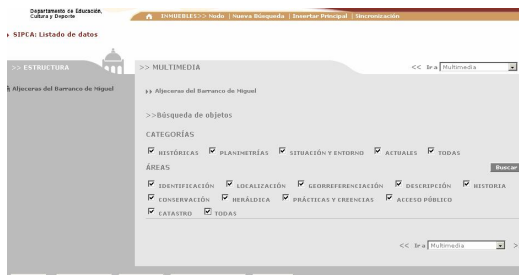
53

## El SIPCA como ejemplo



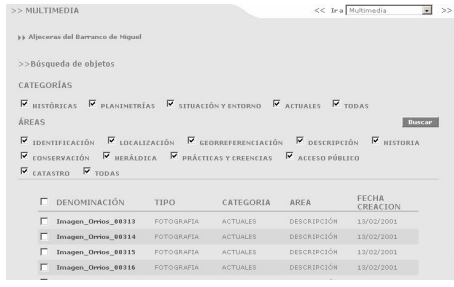
54

# El SIPCA como ejemplo



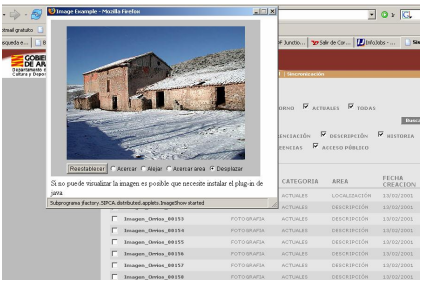
55

# El SIPCA como ejemplo



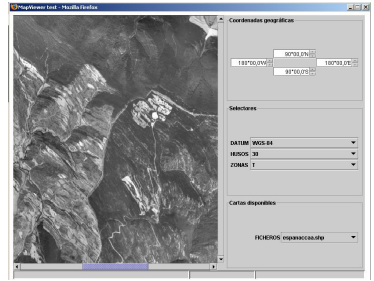
56

# El SIPCA como ejemplo



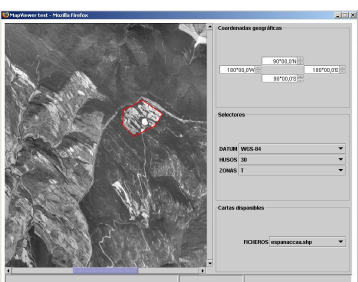
57

# El SIPCA como ejemplo



58

# El SIPCA como ejemplo



59

# El SIPCA como ejemplo



60