



# Desenvolupament en xarxa d'un software lliure com a PFC

Óscar Miras

2 de Juliol de 2010

Jornades de programari lliure de la UPC

# Índice I

- 1 **Introducción y especificación del programa**
- 2 Elección del software libre de partida
- 3 Negociaciones y propuestas
- 4 Diseño e implementación
- 5 Publicación
- 6 ¿Qué me puede aportar?

## Problemas en la gestión tradicional por carpetas.

### Dificultades de la gestión tradicional

- Los criterios de clasificación temática fijados por el usuario pueden cambiar con el tiempo.
- La gestión de muchos archivos es en si difícil.
- Sensible a errores humanos.

¿Podríamos complementar los criterios de ordenación establecidos con otros criterios, respetando la clasificación por carpetas del usuario?



# Clasificación por criterio de orden temporal.


## Definición






Una clasificación por orden temporal consiste en agrupar un conjunto de archivos según sus atributos temporales.

## Características



- Agrupar ficheros que pertenezcan a un periodo de tiempo concreto.
- Aprovecha la idea temporal que tiene el usuario sobre un documento.
- Independiente de la temática de clasificación establecida por el usuario.


 amsterdam2008	 P8060013.JPG	 P8070039.JPG	 P8070063.JPG	 P8090201.JPG
 Brasil	 P8060014.JPG	 P8070040.JPG	 P8070064.JPG	 P8090202.JPG
 BuenosAires	 P8060015.JPG	 P8070041.JPG	 P8070065.JPG	 P8090203.JPG
 carica	 P8060018.JPG	 P8070042.JPG	 P8070066.JPG	 P8090204.JPG
 granada2007	 P8060021.JPG	 P8070044.JPG	 P8070067.JPG	 P8090205.JPG
 granadaOscar	 P8060022.JPG	 P8070045.JPG	 P8070068.JPG	 P8090206.JPG
 Iguazú	 P8060023.JPG	 P8070046.JPG	 P8070069.JPG	 P8090207.JPG
 luis	 P8060024.JPG	 P8070047.JPG	 P8070071.JPG	 P8090208.JPG
 NorteArgentina	 P8060025.JPG	 P8070049.JPG	 P8070072.JPG	 P8090209.JPG
 Ruth	 P8060026.JPG	 P8070050.JPG	 P8070074.JPG	 P8090210.JPG
 t	 P8060027.JPG	 P8070051.JPG	 P8070075.JPG	 P8090211.JPG






 **Buenos**  
feb 11, 2008

 Play    Select 




Add a description


 

 **Iguazú**  
ago 14, 2007

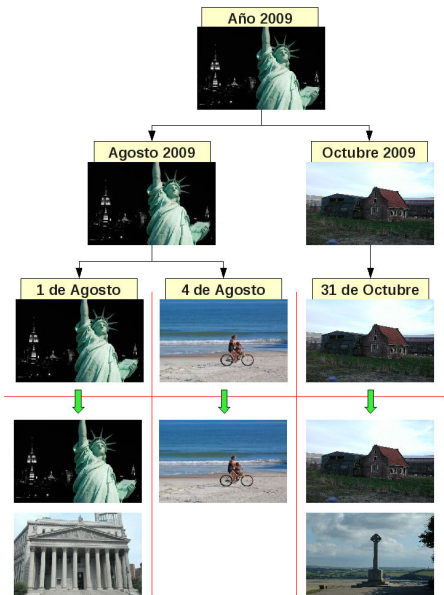
 Play    Select 

Add a description







# Algunas consideraciones.

## Dificultades

- A veces, no podremos elegir un representante.

# Algunas consideraciones.

## Dificultades

- A veces, no podremos elegir un representante.
- Si podemos, la elección de un representante no es trivial.  
¿Cuál escoger?
  - Al azar.
  - El fichero más visitado.
  - Sinergias entre los archivos.
  - Una ponderación de varias características.

# Algunas consideraciones.

## Dificultades

- A veces, no podremos elegir un representante.
- Si podemos, la elección de un representante no es trivial.  
¿Cuál escoger?
  - **Al azar.**
  - El fichero más visitado.
  - Sinergias entre los archivos.
  - Una ponderación de varias características.

# Índice I

- 1 Introducción y especificación del programa
- 2 Elección del software libre de partida**
- 3 Negociaciones y propuestas
- 4 Diseño e implementación
- 5 Publicación
- 6 ¿Qué me puede aportar?

# Ventajas y motivación.

## Ventajas frente al software propietario

- 1 Acceso al código fuente.
- 2 Comunicación directa con los programadores.

## Motivación

- 1 Averiguar si existe algún software que implemente parte de las funcionalidades de nuestro proyecto.
- 2 Aportar nuestro proyecto a la comunidad.

# Características.




## Granularidad de la solución




- ¿Vamos a buscar único software de partida ?
- ¿Vamos a buscar diversos softwares que, modularmente, implementen partes de las funcionalidades que deseamos?




## Consejo.




En cualquier caso, que nuestra decisión abarque parte del mayor número posible de funcionalidades que queremos implementar; y hagamos un estudio de las formas de interconectar diversos módulos.




- No siempre es fácil extraer parte del código de un software.
- No siempre es fácil añadir nuevos módulos a nuestra solución sin haberlo previsto.

Programa	Pros	Contras
 Nautilus		
 Thunar		
 GQView		

Programa	Pros	Contras
 Nautilus	<ul style="list-style-type: none"><li>■ Varias pestañas de navegación.</li><li>■ Soporte de protocolos de red: FTP, WebDav, Samba...</li></ul>	<ul style="list-style-type: none"><li>■ Dependiente del escritorio GNOME.</li><li>■ Trabajo añadido en aprender su funcionamiento.</li></ul>
 Thunar		
 GQView		

Programa	Pros	Contras
 Nautilus	<ul style="list-style-type: none"><li>■ Varias pestañas de navegación.</li><li>■ Soporte de protocolos de red: FTP, WebDav, Samba...</li></ul>	<ul style="list-style-type: none"><li>■ Dependiente del escritorio GNOME.</li><li>■ Trabajo añadido en aprender su funcionamiento.</li></ul>
 Thunar	<ul style="list-style-type: none"><li>■ Ventanas independientes de navegación.</li><li>■ Sigue más la filosofía Unix.</li></ul>	<ul style="list-style-type: none"><li>■ Equipo no interesado en la explotación del criterio temporal.</li></ul>
 GQView		

Programa	Pros	Contras
 Nautilus	<ul style="list-style-type: none"><li>■ Varias pestañas de navegación.</li><li>■ Soporte de protocolos de red: FTP, WebDav, Samba...</li></ul>	<ul style="list-style-type: none"><li>■ Dependiente del escritorio GNOME.</li><li>■ Trabajo añadido en aprender su funcionamiento.</li></ul>
 Thunar	<ul style="list-style-type: none"><li>■ Ventanas independientes de navegación.</li><li>■ Sigue más la filosofía Unix.</li></ul>	<ul style="list-style-type: none"><li>■ Equipo no interesado en la explotación del criterio temporal.</li></ul>
 Geeqie		

Programa	Pros	Contras
 <b>Nautilus</b>	<ul style="list-style-type: none"><li>■ Varias pestañas de navegación.</li><li>■ Soporte de protocolos de red: FTP, WebDav, Samba...</li></ul>	<ul style="list-style-type: none"><li>■ Dependiente del escritorio GNOME.</li><li>■ Trabajo añadido en aprender su funcionamiento.</li></ul>
 <b>Thunar</b>	<ul style="list-style-type: none"><li>■ Ventanas independientes de navegación.</li><li>■ Sigue más la filosofía Unix.</li></ul>	<ul style="list-style-type: none"><li>■ Equipo no interesado en la explotación del criterio temporal.</li></ul>
 <b>Geeqie</b>	<ul style="list-style-type: none"><li>■ Equipo interesado en la explotación del criterio temporal usando metadatos.</li></ul>	<ul style="list-style-type: none"><li>■ No es un navegador de ficheros de propósito general.</li></ul>

enero 2007						
	1	2	3	4	5	6
7	8	9	10	11	12 (3)	13 (7)
14 (1)	domingo 14 		17	18	19	20
21			24	25	26	27
28			31			

## enero 2007

viernes 12



sábado 13



domingo 14



# Geeqie

- Aprovecharemos la clasificación que hace Geeqie usando metadatos.
- La gestión de metadatos y representantes para un número elevado de tipos de ficheros es complicada.

# Índice I

- 1 Introducción y especificación del programa
- 2 Elección del software libre de partida
- 3 Negociaciones y propuestas**
- 4 Diseño e implementación
- 5 Publicación
- 6 ¿Qué me puede aportar?

# Valorar objetivos generales de Geeqie.

## Motivación

Es recomendable averiguar si las líneas de trabajo actuales pueden influir en el desarrollo de nuestro programa. Esto nos permitirá:

- Prever conflictos en el desarrollo.
- Realizar un diseño que prevea los posibles cambios cercanos.
- Valorar si implementar o esperar primero dichos cambios.

Característica a mejorar	Solución	Valoración

Característica a mejorar	Solución	Valoración
<ul style="list-style-type: none"><li>■ Geeqie mapea los atributos temporales de los ficheros en disco.</li><li>■ El acceso es costoso para un número elevado.</li></ul>	<ul style="list-style-type: none"><li>■ <b>SQLite.</b> Gestor de base de datos.</li></ul>	<ul style="list-style-type: none"><li>■ No es objetivo nuestro mejorar la eficiencia del programa.</li><li>■ Crear funciones de tal manera que aprovechen ésta característica cuando se implemente.</li></ul>

Característica a mejorar	Solución	Valoración
<ul style="list-style-type: none"><li>■ Geeqie mapea los atributos temporales de los ficheros en disco.</li><li>■ El acceso es costoso para un número elevado.</li></ul>	<ul style="list-style-type: none"><li>■ <b>SQLite.</b> Gestor de base de datos.</li></ul>	<ul style="list-style-type: none"><li>■ No es objetivo nuestro mejorar la eficiencia del programa.</li><li>■ Crear funciones de tal manera que aprovechen ésta característica cuando se implemente.</li></ul>
<ul style="list-style-type: none"><li>■ Geeqie no distingue entre fotografía y control de interfaz.</li><li>■ La programación de nuevas funcionalidades se vuelve complicada.</li></ul>	<ul style="list-style-type: none"><li>■ <b>Clutter.</b> Biblioteca para la creación de interfaces.</li></ul>	<ul style="list-style-type: none"><li>■ Muy interesante para nuestro cometido.</li><li>■ Demasiado complicado de implementar en Geeqie en un lapso de tiempo corto.</li></ul>

# Decisiones

## Decisión final

- Usaremos el código actual de Geeqie, sin implicarnos en las mejoras que propone el equipo.
- En el futuro, es posible que tengamos que adaptar nuestras funciones; aunque las dejaremos preparadas para que sean fácilmente adaptables.

# Índice I

- 1 Introducción y especificación del programa
- 2 Elección del software libre de partida
- 3 Negociaciones y propuestas
- 4 Diseño e implementación**
- 5 Publicación
- 6 ¿Qué me puede aportar?

# Implementación.

## Ventajas frente a software nuevo

- Muchas funcionalidades implementadas
  - Recorrer directorio para obtener una lista de ficheros ordenada temporalmente.
  - Dibujo de elementos por pantalla.
  - Control de eventos implementado.
- Módulos probados.

# Implementación.

## Dificultades

- Gran dependencia entre los módulos.
- Dificultad de gestionar nuevas estructuras.
- Acceso directo a las estructuras, no encapsulado.
- Código sin documentar.
- Geeqie está bajo desarrollo.

# Índice I

- 1 Introducción y especificación del programa
- 2 Elección del software libre de partida
- 3 Negociaciones y propuestas
- 4 Diseño e implementación
- 5 Publicación**
- 6 ¿Qué me puede aportar?

# Evaluación y reimplementaciones.

## Evaluación y cambios.

- El programa es evaluado por un conjunto de usuarios.
- Se realizan algunas reimplementaciones en el código.

# Publicación.

## Ubicación del programa

- El programa se publica en una rama de desarrollo alternativa dentro de Geeqie.
- Cuando el nuevo escenario implementado sea suficientemente probado y aceptado por el equipo, podrá ser publicado en la rama oficial.

File ▲	Rev.	Age	Author	Last log entry
branches/	1886	3 months	DarkEndymion	
tags/	1909	5 weeks	nadvornik	tagging 1.0 release
trunk/	1915	4 weeks	mow	Add unknown file class to grou

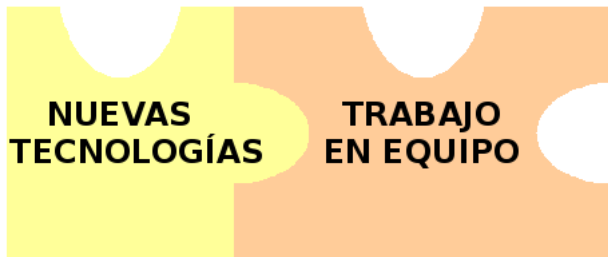
# Índice I

- 1 Introducción y especificación del programa
- 2 Elección del software libre de partida
- 3 Negociaciones y propuestas
- 4 Diseño e implementación
- 5 Publicación
- 6 ¿Qué me puede aportar?**

# Aportes del desarrollo de software libre en red.



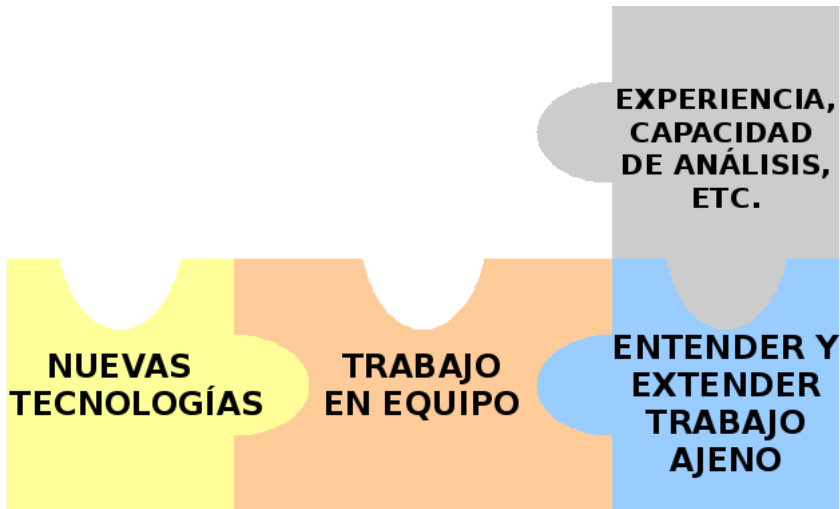
## Aportes del desarrollo de software libre en red.



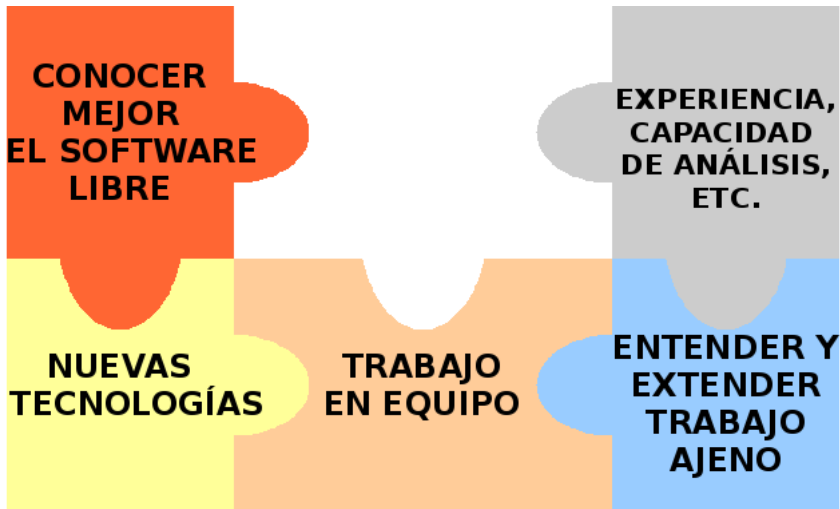
## Aportes del desarrollo de software libre en red.



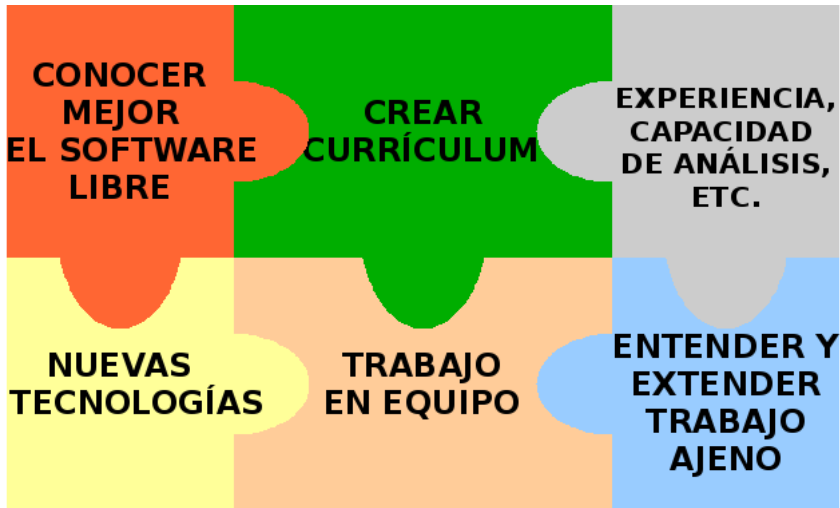
## Aportes del desarrollo de software libre en red.



## Aportes del desarrollo de software libre en red.



## Aportes del desarrollo de software libre en red.



# Aportes del desarrollo de software libre en red.