Sistemas Web Desconectados

Defossé Nahuel, van Haaster Diego Marcos

23 de febrero de 2009

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

Resumen

Capítulo 1

Introducción

Capítulo 2

Motivación

Hoy en día Internet supone un excelente medio para obtener información sobre diversos temas. Para que esta información sea realmente útil es imprescindible que el acceso a ella sea simple e intuitivo, de forma que cualquier persona pueda encontrar y utilizar lo que desea con tan sólo unos conocimientos básicos.

2.1. Un poco de historia sobre la WWW

Esto es posible gracias a la world wide web (Web), la cual surge alrededor del año 1990 en el CERN[1], ante la necesidad de distribuir e intercambiar información acerca de investigaciones de una manera más efectiva. "Tim"John Berners-Lee fue quien definió los componentes básicos que constituyen la Web y dan soporte al almacenamiento, transmisión e interpretación de los documentos de hipertexto.

El término hipertexto no resultaba nuevo en aquel momento: fue acuñado por Ted Nelson en 1965, para designar a los documentos que pueden contener enlaces (referencias) a otras partes del documento o a otros documentos. De esta forma, el documento no necesita ser leído secuencialmente. El hipertexto dio un gran salto con el desarrollo de Internet, posibilitando que un documento este físicamente distribuido en distintas máquinas conectadas entre sí.

Berners-Lee desarrolló lo que por sus siglas en inglés se denominan: el lenguaje HTML[2] o lenguaje de etiquetas de hipertexto, el protocolo HTTP[3] y el sistema de localización de objetos en la Web URL[4], en conjunto con las herramientas necesarias para que la Web funcionase: el primer navegador y editor Web, el primer servidor Web y las primeras páginas Web, que al mismo tiempo describían el proyecto.

La Web funciona siguiendo el denominado modelo cliente-servidor, habitual en las aplicaciones que funcionan en una red: existe un servidor, que es quien presta el servicio, y un cliente, que es quien lo recibe. El cliente web o navegador es un programa con el que el usuario interacciona para solicitar a un servidor web el envío de documentos codificados en lenguaje HTML. Estos documentos

[1] Organización Europea para la Investigación Nuclear
[2] HyperText Markup Language
[3] HyperText Transfer Protocol
[4] Universal Resource Locator http://es.wikipeida.com/urlURL en la Wikipeida

se transfieren mediante el protocolo HTTP. El navegador debe interpretar el HTML para mostrar la información al usuario en el formato adecuado. El servidor web, o simplemente servidor, es un programa que está permanentemente escuchando las peticiones de conexión de los clientes mediante el protocolo HTTP. Cuando llega una petición, el servidor busca el documento que ha sido solicitado en su sistema de archivos y, si lo encuentra, lo envía al cliente; en caso contrario, devuelve un error.

Con el paso de los años se fueron incorporando innovaciones en el servidor. El siguiente paso lógico lo constituyeron los documentos o páginas dinámicas. Éstas se generan al ser solicitadas con información específica del momento o del usuario. Las aplicaciones CGI (Common Gateway Interface) surgieron como una de las primeras maneras prácticas de crear el contenido de una forma dinámica. En una aplicación CGI, el servidor pasa las solicitudes del cliente a un programa externo. Este programa puede estar escrito en cualquier lenguaje que el servidor soporte e interactuar con bases de datos u otros recursos que el servidor posea. Su salida se envía al cliente, en lugar del tradicional archivo estático.

El uso extendido de CGI evolucionó paulatinamente hacia el diseño e implementación de frameworks. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo. Típicamente, un framework puede incluir soporte de base de datos, programas, librerías, entre otros, para ayudar a desarrollar y vincular los diferentes componentes de un proyecto.

Paralelamente a las innovaciones que surgían en el servidor, se comenzaron a agregar en los navegadores nuevas funcionalidades, entre las que se destacan los intérpretes para lenguajes de scripting. JavaScript es uno de estos lenguajes; paulatinamente tomó relevancia y se convirtió en un estándar. Se ejecuta en el navegador al mismo tiempo que se descarga junto con el código HTML y permite una modificación o interacción con el código de la página a través del manejo del DOM (Document Object Model). Netscape incorporo por primera vez a DOM, con el fin de acceder, añadir y cambiar dinámicamente contenido estructurado en una página; esto se denomina HTML dinámico (DHTML).

Los avances, tanto en el servidor como en el navegador, convergieron en el nacimiento de un nuevo concepto: las .ªplicaciones web". En una aplicación web las páginas son generadas dinámicamente en un formato estándar soportado por los navegadores. Agregando lenguajes interpretados en el lado del cliente, se añaden elementos dinámicos a la interfaz de usuario. Generalmente, cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

Las aplicaciones web son populares debido a lo práctico que resulta el navegador web como cliente de acceso a las mismas. También resulta fácil actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. En la actualidad, existe una gran oferta de frameworks web para facilitar el desarrollo de aplicaciones web.

Una ventaja significativa de las aplicaciones web es que funcionan independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para los múltiples sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes.

Las aplicaciones web tienen ciertas limitaciones en las funcionalidades que

ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio, como dibujar en la pantalla o arrastrar y soltar, que no están soportadas por las tecnologías web estándar. Los desarrolladores web, generalmente, utilizan lenguajes interpretados o script en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez. Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías en el lado del servidor. El alcance universal de la Web la ha hecho un terreno muy atractivo para la implementación de sistemas de información. Los sistemas operativos actuales de propósito general cuentan con un navegador web, con posibilidades de acceso a bases de datos y almacenamiento de código y recursos.

La web, en el ámbito del software, es un medio singular por su ubicuidad y sus estándares abiertos. El conjunto de normas que rigen la forma en que se generan y transmiten los documentos a través de la web son regulados por la W3C (Consorcio World Wide Web). La mayor parte de la web está soportada sobre sistemas operativos y software de servidor que se rigen bajo licencias OpenSource1 (Apache, BIND, Linux, OpenBSD, FreeBSD). Los lenguajes con los que son desarrolladas las aplicaciones web son generalmente OpenSource, como e PHP, Python, Ruby, Perl y Java. Los frameworks web escritos sobre estos lenguajes utilizan alguna licencia OpenSource para su distribución; incluso frameworks basados en lenguajes propietarios son liberados bajo licencias OpenSource.

Capítulo 3

Propuesta

Las principales empresas de software se están abocando a desarrollar aplicaciones que funcionan tanto conectadas a la Web como desconectadas de ella, brindando soluciones que reúnan lo mejor de las dos opciones, como lectores de noticias, sistemas de gestión de proyectos y hasta paquetes de oficina (procesadores de texto, hojas de cálculo, etc.), por mencionar algunas.

A mediados del 2007 Google libera un complemento para los navegadores web llamado *Google Gears*, que provee un servidor web de contenido estático, una base de datos y un mecanismo para ejecutar tareas en segundo plano, llamado Worker Pool. Mediante este complemento se permite almacenar localmente los documentos y otros elementos, como las imágenes y el código JavaScript, presentando como novedad la capacidad de almacenar datos de la aplicación web en el cliente, en una base de datos de similares características a las que se encuentran normalmente en los servidores web. Varias aplicaciones de Google comienzan a hacer uso de este complemento, siendo notable su utilización en la aplicación de oficina basada en web, Google Docs (http://docs.google.com), la cual brinda un procesador de texto y una planilla de cálculo que pueden ser usadas incluso cuando el cliente está desconectado de la web, con la condición previa de que haya entrado una vez al sitio y habilitado el modo desconectado.

En la actualidad existen alrededor de 150 frameworks de aplicaciones web (desarrollados en los lenguajes PHP, Python, Ruby, Perl, Lua, ASP, Java, ColdFusion, Groovy y Common LISP), que siguen en mayor o menor grado el patrón Modelo-Vista-Controlador (MVC), este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Capítulo 4

Resultados

WIP[1]

[1]Work in progress :)

Capítulo 5

Plan de trabajo

El framework sobre el cual se instrumentará esta funcionalidad, será elegido tras un análisis de su modelo MVC y la facilidad de desarrollo.

En cuanto a la transferencia de lógica y de datos, se intentará realizar un transferencia de la lógica de negocio de la aplicación web al cliente, de manera de evitar redundancia. Es decir, una vez definido el modelo de datos, éste se adaptará para que pueda ser manejado por el cliente de manera automática.

Esta transferencia del modelo estará sujeto a criterios de seguridad, privacidad y consistencia de de datos que serán añadidos como atributos adicionales al modelo existente en la aplicación web.

También será importante brindar mecanismos para la sincronización de la aplicación desconectada con la aplicación web.

Se espera poder brindar un sistema de migraciones, de manera que la aplicación desconectada pueda adaptarse a los cambios en el modelo de datos en el servidor.

5.1. Primera etapa

5.1.1. Indagación bibliográfica

5.1.2. Selección de framework web

- Definición de las aplicaciones web, su estructura y su mecánica de trabajo, identificando las partes que generan transformación de la información en contraste con las aplicaciones tradicionales.
- Identificación de condiciones y requisitos necesarios para que una aplicación web se ejecute en forma desconectada a la red que la provee.
- Establecimiento de las capacidades y las opciones que existen actualmente para dar soporte a una aplicación desconectada.
- Identificación de los componentes de un framework web.
- Identificación de los frameworks MVC. Generalidades sobre las opciones más populares.
- Generalidades sobre el modelo de desarrollo de los framework web.

- Evaluación de un conjuto de frameworks MVC.

5.1.3. Diseño e implementación

Generación de contenido a partir de un framework

- Generación de contenido .[es]táticoütilizando los componentes de un framework MVC.
- Generación de contenido dinámico de ejecución retardada en el cliente.
- Almacenamiento de datos en el cliente y diferentes mecanismos de transporte.
- Organización de datos y meta información sobre el modelo de dominio.
- Delegación de responsabilidad al cliente. Ejecución concurrente.
- Captura de eventos de la interfase en modo desconectado.
- Consideraciones sobre degradación de servicio.

Desconexión

- Sincronización basada en marcas de tiempo, privilegios y prioridades.
- Delegación del control de requisiciones web a rutina totalmente manejada por el cliente.
- Rutina de puente de contenido dinámico/estático.
- Rutinas de intercepción de flujo de control o controlador del lado del cliente.

Interrelación con la aplicación

- Esquema de autorización y privilegios sobre los datos que son enviados al cliente.
- Generación de esquemas de reflexión de las entidades del modelo de negocios de la aplicación online y sus contrapartes offline.
- Agregación de meta información para la generación de una vista de datos para lograr Seguridad.
- Provisión de garantías de integridad de datos.
- Consideraciones sobre seguridad.
- Consideraciones sobre políticas de sincronización.
- Consideraciones sobre la evolución del modelo de datos: migraciones.

5.1.4. Ejemplo de aplicación

5.1.5. Conclusiones y posibles extensiones

Capítulo 6

Conclusiones

$$\int_{a=0}^{b=3} \{ \sum_{ahora}^{9/3/2009} trabajo\ de\ diego\ + \sum_{9/3/2009}^{fines\ de\ abril} aportes\ de\ nahuel \} \quad (6.1)$$

Bibliografía

[1] Tratado de la W3C sobre las aplicaciones web desconectadas. http://www.w3.org/TR/offline-webapps/

[2] Llevando las aplicaciones basadas en web al ámbito offline. http://www.linux.com/feature/60827/

[3] Entrevista con Kevin Lynch sobre el futuro de las aplicaciones web desconectadas. http://www.technologyreview.com/read$_{a}rticle.aspxch = specialsections\&sc\bar{e}merging08\&id = 20245$

[4] Well, I'm Back Robert O'Callahan. Aplicaciones web desconectadas. http://weblogs.mozillazine.org/roc/archives/2007/02/offline$_{w}eb_{a}pp.html$

[5] Documentación para desarroladores de Google Gears. http://code.google.com/apis/gears/

[6] Biografía de Tim Barners-Lee. Wikipedia http://es.wikipedia.org/wiki/Tim$_{B}erners - Lee$

[7] Definición de Web, Wikipedia. http://es.wikipedia.org/wiki/World$_{W}ide_{W}eb$

[8] Definición de Protocolo HTTP, Wikipedia. http://es.wikipedia.org/wiki/HTTP

[9] Definición de JavaScript, Wikipedia. http://es.wikipedia.org/wiki/JavaScript

[10] Definición de CGI, Wikipedia. http://es.wikipedia.org/wiki/Cgi

[11] Definición de DHTML, Wikipedia. http://es.wikipedia.org/wiki/DHTML

[12] Definición de Framework Web, Wikipedia. http://en.wikipedia.org/wiki/Web$_{a}pplication_{f}ramework$

[13] Listado de frameworks web, Wikipedia. http://en.wikipedia.org/wiki/List$_{o}f_{w}eb_{a}pplication_{f}rameworks$