



## Sistema de gestión de solicitudes de descargas para la red de la Universidad Agraria de La Habana\*

**Lisué Capó Marrero**

**Daily García Dorta**

Carrera: Ingeniería Informática

Universidad Agraria de La Habana (UNAH) Cuba.

**Resumen:** Cualquier estudiante o profesional, en su desempeño, debe realizar tareas vinculadas a investigaciones auxiliándose de Internet, pero la descarga de información en ocasiones se dificulta, escenario que intentan corregir los sistemas gestores de descargas. La presente investigación enmarcó el ciclo completo de desarrollo, guiado por la metodología ágil *Extreme Programming* (XP), de una aplicación web basada en el CMS Drupal, que incluye funcionalidades propias de un sistema gestor de descargas. Este sistema está concebido como una herramienta informática para contribuir a una mayor eficiencia en los procesos que se llevan a cabo para la descarga de información de Internet por docentes y estudiantes de la Universidad Agraria de La Habana (UNAH).

**Palabras clave:** CMS Drupal; descarga de información; Internet; sistemas gestores de descargas.

---

\* Trabajo tutorado por el Ing. Yasmani Ceballos Izquierdo.  
Recibido: 4 enero 2014 / Aceptado: 28 febrero 2014.

## Download management system for the petitions of the network of the Agrarian University of Havana

**Abstract:** Students or professionals, in their daily occupations should carry out several investigations tasks being aided of Internet, but the downloading information in occasions it's difficult, scenario that try to correct the download managers. The present investigation frames a complete development cycle guided by the *Extreme Programming* (XP) methodology of a web system-based on the Content Management System Drupal that includes functionalities of a download manager system. This system is conceived as a computer tool to contribute to a bigger efficiency in the processes that are carried out for downloading information of Internet by educational and students of the Agrarian University of Havana.

**Key words:** CMS Drupal; downloading information; Internet; download managers.

## Introducción

La expansión de las tecnologías de la información y las comunicaciones (TIC), conjuntamente con Internet, representan un motor de impulso al desarrollo científico de cualquier nación, siendo soporte y fundamento de la sociedad del conocimiento. Esta nueva situación ha propiciado un cambio acelerado hacia una sociedad que basa su riqueza en la gestión de la información y el conocimiento disponibles en Internet.

En Cuba el uso social de la red de redes, a través de los Joven Club de Computación, salas de navegación, universidades e instituciones de investigación y otras opciones, ha propiciado un correcto aprovechamiento de las TIC, permitiendo utilizar esta poderosa herramienta para lograr una sociedad cada vez más preparada y con acceso al conocimiento. Siguiendo esta estrategia, las universidades cubanas se han sumado a este llamado como instituciones totalmente centradas en el conocimiento y su gestión.

La UNAH, una de las principales instituciones de la educación superior cubana, es pilar en la formación de profesionales en varias carreras, disponiendo de una infraestructura tecnológica que, a través del acceso a las tecnologías y a Internet, apoya la actividad científica, la extensión universitaria, la superación y formación integral de profesionales con alta calidad y pertinencia. Con tales fines, la descarga de información docente e investigativa, de forma eficiente, se ha convertido en algo de suma importancia para los usuarios de la red de la UNAH. Esta tarea tan común, generalmente no sucede de forma correcta y segura, pues intervienen en el proceso factores o condiciones variables como: 1) ancho de banda de la red, 2) tamaño del archivo, y 3) concurrencia de usuarios utilizando el servicio en horario docente. Por los motivos expuestos, las descargas están frecuentemente afectadas pues se reinician periódicamente, se descargan ficheros incompletos, o en ocasiones no ocurren.

En la actualidad existen diferentes tipos de aplicaciones informáticas que sirven para la administración o aceleración de descargas, entre ellas los sistemas gestores de descargas (*Download Manager*, en inglés) Básicamente consisten en: *"una utilidad que reemplaza la función de descarga nativa del navegador web. Los administradores de descargas ofrecen numerosas características, incluyendo la habilidad de continuar una descarga después de varios fallos de conexión. Los enlaces pueden ser obtenidos*

*durante el día y planificados para ejecutarse automáticamente por la noche cuando la congestión de la red es menor” (PC Magazine, 2011).*

Sin embargo, no se consideró la utilización de un sistema gestor de descarga como una solución factible pues su uso masivo e incontrolado por los usuarios solo provocaría un empeoramiento crítico de las condiciones de navegación. En contradicción con lo anterior, un mecanismo único, controlado y eficaz para atender todas las solicitudes podría llevar al mejor aprovechamiento y/o correcto funcionamiento de este servicio de vital importancia para el óptimo desempeño profesional del centro universitario.

Es por tal motivo que se hace necesario desarrollar una aplicación web mediante la cual los usuarios de la red UNAH pudieran realizar sus solicitudes de descargas para que estas fueran gestionadas de forma controlada.

### **Metodología de desarrollo de software**

Para garantizar un proceso de desarrollo satisfactorio del sistema se efectuó un estudio de las metodologías, herramientas y tecnologías que existen en la actualidad, para identificar las que serán empleadas en la solución.

Cada proyecto de software, sin importar la envergadura del mismo, tiene como objetivo principal entregar al cliente un producto final con la calidad requerida y que cumpla con todas las especificaciones establecidas. Por tal motivo, surgen las metodologías de desarrollo de software que utilizan una serie de procedimientos, técnicas, herramientas, y soporte documental para realizar un producto de calidad.

Para el desarrollo de las funcionalidades de la aplicación web se seleccionó un método ágil por estar especialmente orientadas para proyectos cambiantes, de corta duración y con equipos pequeños. Las metodologías ágiles constituyen una solución a medida para esos entornos, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto (Penadés *et al.*, 2003). Después de un análisis de las principales metodologías ágiles se eligió la utilización de XP por promover el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y favoreciendo un buen clima de trabajo. De esta manera, 1) los

roles, 2) el proceso de planificación, 3) la disponibilidad del cliente, 4) programación por pares, 5) ubicación del equipo de desarrollo en el mismo local, 6) no trabajar horas extras y 7) el modelo de 40 horas de trabajo semanales que plantea XP, se ajustan perfectamente al ambiente real del presente proyecto.

### Análisis y diseño de la solución

Una vez seleccionada la metodología, se identificaron y describieron las posibles funcionalidades a través de la exposición de los principales artefactos que esta propone en las fases de Exploración y Planificación, concretando la cantidad de entregas, prioridad y tiempos de ejecución. Como parte de los artefactos generados se definieron 3 iteraciones que incluyen un total de 18 historias de usuario (HU) (4 críticas, 7 altas y 7 medias o bajas) que describen los aspectos principales a tener en cuenta para el desarrollo de la solución, contemplando una visión futura de las funcionalidades que debe cumplir el sistema. Además, se determinó, mediante el plan de duración de iteraciones, el orden de desarrollo de las HU dentro de cada iteración así como la estimación completa de la duración de cada iteración (3 semanas cada una).

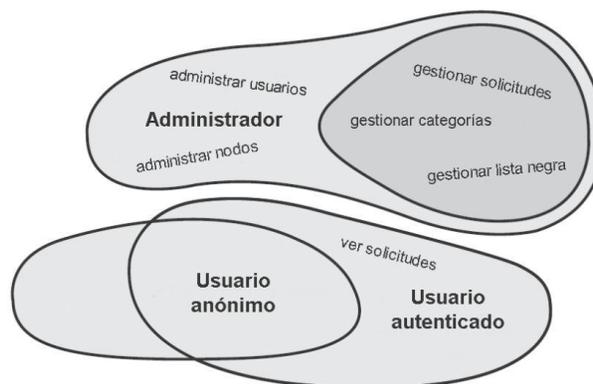


Figura 1. Simples roles con sus permisos mostrados como un diagrama de Venn. Imagen modificada de Hogbin *et al.* (2009).

Asociado a las HU se construyó el plan de entregas que definió un cronograma que especifica las entregas que deben hacerse y conjuntamente se elaboraron tres tarjetas CRC (Clase-Responsabilidad-Colaboración) que representan las funcionalidades a implementar, definiendo las responsabilidades y los colaboradores. Además, se elaboró el primer prototipo de interfaz de usuario no funcional de la solución, en esta etapa,

muy cercano a una arquitectura de información inicial, pero que más adelante fue convergiendo hacia el diseño visual definitivo.

Por otra parte, el estudio de la audiencia a la que va dirigida la solución delimitó que existen tres tipos de usuarios que interactúan con el sistema con diferentes niveles de acceso (Figura 1).

## Implementación de la solución

Para el desarrollo de las funcionalidades de la aplicación web se utilizó el CMS (sistema de gestión de contenidos) Drupal 7.20, proporcionando una arquitectura flexible con funcionalidades para la autenticación, manejo de roles y permisos, control de la presentación mediante plantillas fácilmente personalizables (temas), y módulos para aumentar sus funcionalidades, lo cual permite reducir considerablemente el tiempo de desarrollo. Las plantillas se componen predominantemente de código XHTML, CSS, y en menor tamaño de código PHP (Tomlinson *et al.*, 2010), sin descartar la utilización de código Javascript que en las últimas versiones de Drupal se ha optimizado mucho mediante la librería JQuery (Chaffer *et al.*, 2007).

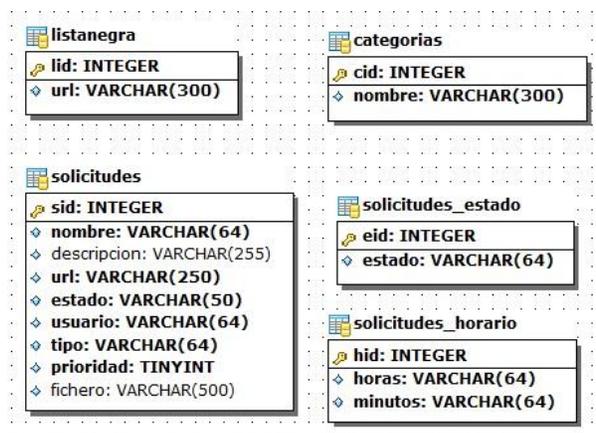


Figura 2. Imagen de la estructura parcial de la base de datos de la solución que muestra las tablas añadidas a la base de datos de Drupal.

El primer paso para comenzar la codificación del sistema fue el diseño y creación de la estructura de las tablas que se agregaron a la base de datos de Drupal, proceso incluido en la etapa inicial de implementación de los módulos para la solución. La instalación por defecto del CMS Drupal 7.20 incluye 74 tablas para el manejo y

almacenamiento de datos (por ejemplo de: contenidos, bloques, nodos, usuarios, entre otros). Este número de tablas aumentó a partir de la inclusión de los módulos desarrollados (que incorporan funcionalidades nuevas) debido a la necesidad de almacenar nuevos datos (Figura 2).

El proceso de creación de los atributos identificadores y selección de su tipo de dato correspondiente, no se realizó arbitrariamente, sino que para las nuevas tablas se aplicó el patrón "Llaves subrogadas", el cual es de gran utilidad pues facilita la interacción con la base de datos en un futuro. Este patrón plantea crear una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. Normalmente se usan enteros en columnas *identity* o GUID (*Global Unique Identifier*) lo cual permite que las tablas sean más fáciles de consultar a partir del identificador (Rankins *et al.*, 2000). Sin utilizar "Llaves subrogadas", hubiese sido necesario un diseño muy cuidadoso, ya que un error pudiera conllevar a inconsistencia en la base de datos y, consecuentemente, a producir información errónea en consultas, de ahí la importancia de utilizar una pauta.

Drupal es un sistema con una arquitectura modular donde las funcionalidades o nuevos tipos de contenido tales como: foros, noticias, galerías están incluidas en los módulos y estos se clasifican en 2 grandes tipos (Gil, 2011): **módulos del núcleo**, incluidos en la instalación por defecto de Drupal y necesarios para el correcto funcionamiento del mismo; y **módulos contribuidos**, que son los desarrollados por la comunidad y proporcionan una serie de funcionalidades que se pueden agregar al sitio según las necesidades del desarrollador.

Durante el desarrollo de la aplicación web se investigaron los módulos del núcleo para elegir los que aportan funcionalidades adicionales que contribuyen a la solución, aunque no incluyen las funcionalidades básicas requeridas; los módulos seleccionados fueron Locale, PHP filter, Poll, Testing. Además de los módulos del núcleo, a la solución se integraron los siguientes módulos contribuidos: *Visitors*, *Private Message*, *Nice Menus*, *Wysiwyg*, *Site map* y *LDAP*. El núcleo de Drupal 7.20, por defecto, incorpora 44 módulos que están disponibles tras la instalación (Gil, 2011), pero ninguno de estos módulos u otros contribuidos aportan las funcionalidades necesarias para la solución, por lo que fue necesario implementar nuevos módulos adicionales.

A continuación se describen los nuevos módulos desarrollados y agregados a la instalación básica de Drupal:

**GestionarSolicitud:** este módulo incluye las funcionalidades para: gestionar una solicitud (adicionar una solicitud de descarga de información y eliminar una solicitud de descarga), además de importar, mediante un archivo XML, varias solicitudes a la vez. El administrador podrá establecer el horario en que comenzarán a realizarse las descargas automáticamente y detenerlas, igualmente establecer prioridades a las solicitudes de descargas y de aprobar o rechazar una solicitud de descarga. Además, ambos podrán consultar las solicitudes completadas, las solicitudes recientes, las estadísticas de las descargas y realizar todas las validaciones correspondientes.

**GestionarCategoria:** este módulo permite que el administrador pueda adicionar, eliminar y consultar una categoría.

**GestionarLista\_Negra:** este módulo permite que el administrador pueda adicionar, importar, eliminar, y consultar las URLs que se encuentren en la lista negra y validar que las mismas estén correctamente escritas.

La programación de los módulos se realizó en parejas y siguiendo el proceso descrito por Tomlinson *et al.* (2010). En una primera etapa de la implementación se crearon los archivos con la información de los mismos (.info) y archivos (.install) para crear la estructura de las tablas asociadas (figura 2). Para cada nuevo módulo, en un tercer archivo (.module) se implementaron las funcionalidades críticas. Además, es necesario destacar que, debido a la manera particular que Drupal establece la escritura de módulos, en la implementación de los módulos de la solución están reflejados patrones de diseño creacionales (*Singleton*), patrones de diseño estructurales (*Bridge*), y patrones de diseño de comportamiento (*Chain of Responsibility* y *Command*).

Además, XP promueve la programación basada en estándares en la práctica (*coding standars*) de manera tal que el código sea fácilmente entendible por todo el equipo, y facilite la recodificación, lo que fortalece la comprensión de los programadores y hace que el código sea más legible (Jeffiries *et al.*, 2000).

En cuanto a la integración de los módulos con la arquitectura de Drupal, se aplicó una integración continua tal y como propone XP. Es decir, se fueron integrando pequeños

fragmentos de código, para evitar que al finalizar el proyecto se tuviera que invertir grandes esfuerzos en la integración final. En todo buen proyecto de XP tendría que existir una versión al día integrada, de manera que los cambios siempre se realicen en esta última versión (Beck, 1999).

Referente al diseño visual, un tema en Drupal define la apariencia visual del sitio, además de ser lo más visible, y discutiblemente su más influyente elemento (Shreves, 2011). Por las especificaciones y necesidades del cliente, se creó el tema *Gestor Unah*, a partir de modificaciones a uno ya existente llamado *tarski*. Para ello, se definió el nombre (en este caso *Gestor Unah*) y se colocó en un nuevo directorio dentro (*sites/all/themes*) usando el nombre definido (por ejemplo *sites/all/themes/gestor*). A continuación, se creó un archivo con la información básica como el nombre, versión, regiones, CSS, entre otras, y se almacenó con la extensión *.info* para así adicionar el nuevo tema a los registros de Drupal (Figura 3).

Establecer prioridades de la cola de solicitudes **A**

#	Solicitud	Operación	Prioridad	Usuario
1	<b>image 001</b> <a href="http://localhost/New20folder/image001.jpg">http://localhost/New20folder/image001.jpg</a>		2	admin
2	<b>tesis de geología de Cuba</b> <a href="http://localhost/New%20folder/1945_Hermes_Tesis.pdf">http://localhost/New%20folder/1945_Hermes_Tesis.pdf</a>		1	gestor
3	<b>video</b> <a href="http://localhost/New%20folder/14.mpg">http://localhost/New%20folder/14.mpg</a>		1	gestor
4	<b>libro de historia</b> <a href="http://localhost/New%20folder/Luis%20Silva%20-...">http://localhost/New%20folder/Luis%20Silva%20-...</a>		1	admin

Rechazar todo

Cola de solicitudes **B**

Estado del gestor: **Detenido**

#	Solicitud	Operación	Prioridad	Usuario
1	<b>Drupal 7</b> <a href="http://localhost/New%20folder/Beginning_DrupaL_7.pdf">http://localhost/New%20folder/Beginning_DrupaL_7.pdf</a>		3	admin
2	<b>Drupal 7</b> <a href="http://localhost/New%20folder/Beginning_DrupaL_7.pdf">http://localhost/New%20folder/Beginning_DrupaL_7.pdf</a>		2	admin
3	<b>Drupal 7 Module Development</b> <a href="http://ftp.sld.cu">http://ftp.sld.cu</a>		1	admin

Iniciar Detener Rechazar todo

Figura 3. Interfaz ilustrativa del tema para el sistema gestor de descargas implementado. a) Interfaz para establecer prioridades en las solicitudes. b) Ejemplo de la cola con prioridad con tres solicitudes y el gestor detenido.

Para modificar y/o personalizar la apariencia visual del tema se utilizaron las hojas de estilo en cascada (CSS), sobrescribiendo clases e identificadores existentes o creando

nuevos. Para el tema desarrollado se modificaron 8 archivos CSS, y se utilizó como herramienta de apoyo para facilitar el trabajo la extensión Firebug 1.11.2 del navegador Mozilla Firefox. Además, el tema *Gestor Unah* incluye los siguientes archivos:

- block.tpl.php
- book-navigation.tpl.php
- comment-wrapper.tpl.php
- comment.tpl.php
- forum-icon.tpl.php
- forum-list.tpl.php
- maintenance-page.tpl.php
- node.tpl.php
- node--book.tpl.php
- page.tpl.php
- page--admin.tpl.php
- page-content.tpl.php
- poll-bar.tpl.php
- poll-bar--block.tpl.php
- profile-listing.tpl.php
- template.php
- theme-settings.php
- favicon3.png

### **Desarrollo conducido por pruebas**

El desarrollo de software en la metodología XP es orientado a pruebas (*Test driven development*, TDD por sus siglas en inglés). TDD, como tal, se engloba dentro del ciclo de vida de la metodología XP, en las fases de iteración, producción y mantenimiento. Pero tal es su importancia, que muchos desarrolladores ya la consideran una metodología independiente, un punto y aparte después de XP. Tal estrategia o metodología es perfectamente aplicable para evaluar la calidad del código de módulos desarrollados para Drupal (Butcher *et al.*, 2010).

Una estrategia de prueba requiere crear código que programáticamente ejercite cada simple línea de la solución para asegurar que esta funcione como se supone. Probar el código programáticamente es una práctica bien establecida en la industria del desarrollo de software (Travis, 2011). En Drupal 7, esta capacidad es otra de las ventajas incorporadas al núcleo de Drupal, al integrarse una herramienta de prueba (módulo *Testing*) que permite definir y automatizar pruebas para el sitio y los módulos desarrollados. Así, utilizando el marco de trabajo que propone *Testing*, los desarrolladores pueden realizar pruebas unitarias y funcionales para validar funcionalmente su código (Travis, 2011). El módulo *Testing* está anexado como parte de Drupal 7, pero es necesario instalarlo como cualquier otro módulo.

La implementación de los casos de pruebas se fundamenta en el paradigma de la POO con cada prueba expresada como un objeto creado a partir de la clase *DrupalUnitTestCase* o *DrupalWebTestCase* que provee Drupal (Figura 4).

► PRUEBAS UNITARIAS PARA VALIDAR URL

Verifica la de la funcionalidad validar URL  
3 pases, 1 fallo, y 0 excepciones

MENSAJE	GRUPO	NOMBRE DE ARCHIVO	LÍNEA	FUNCIÓN	ESTADO
Valida una URL verdadera	Other	GestionarLista_Negra.test	96	GestionarLista_NegraURLTestCase->testValidarURL()	✓
Valida una ftp verdadera	Other	GestionarLista_Negra.test	99	GestionarLista_NegraURLTestCase->testValidarURL()	✗
Identifica una URL falsa	Other	GestionarLista_Negra.test	102	GestionarLista_NegraURLTestCase->testValidarURL()	✓
Verifica contra espacio en blanco	Other	GestionarLista_Negra.test	105	GestionarLista_NegraURLTestCase->testValidarURL()	✓

Figura 4. Ejemplo de pruebas unitarias iniciales para el método `GestionarLista_Negra_validarURL` del módulo `GestionarLista_Negra`. Un fallo indica que es necesario corregir el código. Este ejemplo sugiere cómo funciona TDD.

Al concluir la fase de producción la ejecución de 180 pruebas automatizadas con *Testing* tardó siete minutos con veinte segundos. Intentar realizar manualmente estas pruebas sin automatización hubiera requerido muchas horas y con el potencial del error humano durante las pruebas. Además de las pruebas anteriores, se llevaron a cabo pruebas de aceptación por el cliente, y pruebas de carga y estrés, empleando la herramienta JMeter 2.4 antes de poner en funcionamiento real el sistema.

## Conclusiones

El sistema informático implementado permite optimizar el proceso de gestión de solicitudes de descargas de información de manera que se puede aplicar en la UNAH o en otras instituciones que tengan los problemas mencionados en la descarga de información.

La inclusión en la red de la universidad de un sistema que aporta funcionalidades para gestionar solicitudes de descargas de información, facilita a los profesores, estudiantes y trabajadores de la universidad obtener la información que necesitan para su preparación docente e investigativa, además, el uso adecuado de esta herramienta en un horario no docente propicia un mejor aprovechamiento de la red del centro reduciendo los cuellos de botella (*bottlenecks*) y permitiéndole un mayor control al administrador de la red sobre la información que se descarga en la universidad.

## Referencias bibliográficas

- BECK, K. 1999: *Extreme Programming Explained: Embrace Change*. Editorial Addison-Wesley, Boston. ISBN 0201616416.
- BUTCHER, M. Y DUNLAP, G. 2010: *Drupal 7 Module Development*. Editorial Packt Publishing, Birmingham.
- GIL, F. 2011: *Experto en Drupal 7. Curso de creación y gestión de portales web con Drupal 7. Nivel Inicial*. Publidisa. ISBN 978-84-939410-5-5.
- HOGBIN, J. Y KÄFER, K. 2009: *Front End Drupal Designing, Theming, Scripting*. Prentice Hall, Boston. ISBN 978-0-13-713669-8.
- CHAFFER, J. Y SWEDBERG, K. 2007: *jQuery Reference Guide. A Comprehensive Exploration of the Popular Javascript Library*. Packt Publishing, Birmingham.
- JEFFRIES, R.; ANDERSON, A. Y HENDRICKSON, C. 2000: *Extreme Programming Installed*. Addison Wesley, Boston. ISBN: 0-201-70842-6.
- PC Magazine Encyclopedia. 2011. [En línea]. Consultado: 22 de mayo de 2013. Disponible en: [http://www.pcmag.com/encyclopedia\\_term/0,,t=&i=41944,00.asp](http://www.pcmag.com/encyclopedia_term/0,,t=&i=41944,00.asp).

PENADÉS, M. Y LETELIER, P. 2003: Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Universidad Politécnica de Valencia.

RANKINS, R.; BERTUCCI, P. Y JENSEN, P. 2000: *Microsoft SQL Server 2000 Unleashed*. Sams Publishing, EUA.

SHREVES, R. 2011: *Drupal 7. Themes*. Packt Publishing, Birmingham. ISBN 978-1-849512-76-3.

TOMLINSON, T. Y VANDYK, J. 2010: *Pro Drupal 7 Development*. Apress, New York. ISBN 978-1-4302-2839-4.

TRAVIS, B. 2011: *Pro Drupal 7 for Windows Developers*. Apress, New York.