



# **Trabajadores. Creación de un sitio modelo para el desarrollo de portales web utilizando el CMS Plone.**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Autores:** Ailyn Alfonso González  
Joel Armada Herrera

**Tutores:** Dunia Suárez Ferreiro  
Karel Antonio Verdecia Ortiz

## Agradecimientos

---

Agradecemos a todos y en especial a nuestra tutora Dunia Suárez Ferreiro.

## Dedicatoria

---

A nuestras familias.

*"En software, muy raramente partimos de requisitos con sentido. Incluso teniéndolos, la única medida del éxito que importa es si nuestra solución resuelve la cambiante idea que el cliente tiene de lo que es su problema"*

*-- Jeff Atwood*

## Resumen

---

La Universidad de las Ciencias Informáticas, específicamente la Facultad 10, protagoniza el proceso de informatización de la prensa cubana que está teniendo lugar en estos momentos en nuestro país. El principal objetivo de tan importante tarea es lograr mejorar el posicionamiento de nuestros medios digitales en Internet, llevando de esta forma la realidad cubana a cada rincón del mundo.

Para llevar adelante este proceso se seleccionó el CMS (*Content Management System*) Plone. El primer sitio publicado como resultado de este proyecto es el periódico “Trabajadores”. A partir de problemas detectados durante su desarrollo producto de la extrema lentitud con la que se avanzaba debido a la poca documentación sobre la tecnología y falta de capacitación de los integrantes del proyecto, además de inconformidades planteadas por parte de los periodistas del medio, surge la idea de crear un sitio modelo, partiendo del periódico “Trabajadores”, que constituya una plantilla a seguir en el futuro desarrollo de portales web informativos en el proyecto de Informatización de la Prensa.

## Palabras claves

---

CMS, Sistema manejador de contenidos, foro, encuesta, boletines, portal, sitio, Trabajadores, periódico, modelo, plantilla, Informatización de la Prensa, prensa cubana, digitalización, Zope, Plone, Python, Scrum.

# Índice

---

Agradecimientos .....	II
Dedicatoria .....	III
Resumen.....	V
Palabras claves.....	VI
Introducción .....	1
Capítulo 1. Fundamentación Teórica.....	4
Sitios web informativos.....	4
Usabilidad y Accesibilidad. ....	4
Servicios brindados por sitios web informativos. ....	7
Sistema Manejador de Contenidos. ....	8
Plone y Zope.....	9
Metodologías de desarrollo. ....	12
Capítulo 2. Características del Sistema.....	14
Antecedentes del periódico Trabajadores.....	14
Estructura del sitio.....	15
Productos. ....	17
Características del sitio modelo. ....	18
Definición de estándar de organización del código del sitio. ....	18
Product Backlog. ....	19
Capítulo 3. Implementación y Prueba.....	22
Planificación del proyecto.....	22
Sprints Backlogs.....	22
Pruebas .....	29
Prueba #1 .....	29

Prueba #2 .....	32
Prueba #3 .....	35
Prueba #4 .....	39
Prueba #5 .....	42
Prueba #6 .....	42
Entrada: .....	42
Conclusiones .....	44
Recomendaciones .....	45
Bibliografía .....	46
Referencias.....	48
Anexos .....	49
Anexo I. Metodología Scrum .....	49
Los roles en <i>Scrum</i> .....	50
El flujo de <i>Scrum</i> .....	50
Artefactos de Scrum. ....	53
Anexo II. Códigos de estado HTTP .....	55
1xx: Respuestas informativas .....	55
2xx: Peticiones correctas .....	56
3xx: Redirecciones .....	57
4xx Errores del cliente .....	58
5xx Errores de servidor.....	61
Glosario de Términos.....	62



# Introducción

---

La puesta en Internet de sitios informativos fue desde el primer momento un desafío para cualquier medio periodístico a nivel internacional. Con esta opción la información llegaría a un mayor número de usuarios y podría accederse a ella desde cualquier lugar del mundo. Muchos diarios fueron puestos en la red desde el principio, pero sin haber hecho antes un estudio profundo del largo camino que tendrían que recorrer. Actualmente, en lo que se refiere a este aspecto, se han logrado grandes avances y se han definido un gran número de condiciones para que un periódico digital pueda lograr un buen posicionamiento en la llamada “Autopista de la información”.

La prensa cubana hace su aparición en Internet el 1ro de abril de 1996 con el periódico Granma Internacional siguiéndole, a partir del 2000, la mayoría de los medios de prensa cubanos, existentes en el país, logrando así un mayor dinamismo del mensaje cubano para el mundo. El posicionamiento de estos nuevos portales web ha ido cambiando junto con los grandes avances y las nuevas técnicas inteligentes que utilizan los buscadores y otros servicios para mostrar la información que minuto a minuto se publica en la red. Muchos de los periódicos digitales de Cuba quedan fuera del posicionamiento mundial debido a que no cumplen con los parámetros tecnológicos que los buscadores toman en cuenta para mostrar sus resultados.

Actualmente en Cuba se está llevando a cabo un proceso de digitalización de la prensa con el objetivo de corregir los errores que fueron cometidos en un inicio y que actualmente frenan el logro de un buen posicionamiento en Internet por parte de los medios cubanos de prensa. El Comité Central del PCC en coordinación con la Universidad de las Ciencias Informáticas, tomó la decisión de sacar adelante este proyecto con la utilización de la tecnología de software libre Plone/Zope.

En este momento el desarrollo de un portal web utilizando Plone y Zope se hace muy complicado ya que la curva de aprendizaje de esta tecnología es muy alta y existe muy poca documentación sobre el tema, además de que la existente no está actualizada.

Se hace extremadamente necesario agilizar este proceso ya que la Facultad 10, de la Universidad de las Ciencias Informáticas, tiene la misión de digitalizar la prensa cubana y hay una gran cantidad de sitios que deben ser desarrollados.

El primer sitio de la prensa cubana en estar totalmente migrado a Plone/Zope es el sitio Trabajadores. La dirección del periódico Trabajadores ha mostrado cierta inconformidad con el funcionamiento del sitio y recientemente propuso una serie de cambios que desea se le realicen al portal. La dirección del PCC ha planteado la necesidad de corregir todos los problemas que se están presentando en este medio puesto que ha sido el primero en migrar y es necesario que sirva de ejemplo para que los demás periódicos migren también a Plone.

De la situación problemática antes expuesta se deriva el siguiente **problema científico**: ¿Cómo contribuir, mediante un sitio web modelo, a agilizar e incentivar el desarrollo de portales web en los medios de prensa cubanos utilizando Plone y Zope?

Para lograr un sitio modelo se requiere hacer un estudio profundo de todos los servicios y necesidades comunes que presentan actualmente los medios de prensa cubanos. Los portales web de la prensa cubana son el **objeto de estudio** central de este trabajo, tomando como **campo de acción** el portal web del periódico Trabajadores.

El presente trabajo se plantea, como **objetivo general**, crear un nuevo sitio para el periódico Trabajadores que sea un modelo a seguir para el desarrollo de futuros portales web informativos utilizando Plone/Zope.

Surgen entonces las siguientes **preguntas científicas**:

- ¿Cuáles son las bases teóricas necesarias para el trabajo con Zope/Plone?
- ¿Cuáles son los principales servicios que debe brindar un sitio web informativo?
- ¿Cómo crear un sitio Web informativo modelo mediante Trabajadores?

Las **principales tareas investigativas** que se proponen para dar respuesta a las preguntas antes expuestas se listan a continuación:

- Investigar las bases teóricas para el trabajo con Zope/Plone.
- Realizar un estudio sobre los principales servicios que son brindados actualmente por los sitios web informativos.
- Desarrollar un sitio modelo, que agilice la creación de nuevos portales web para la prensa cubana.

Para el desarrollo de este trabajo se emplearon los siguientes **métodos científicos de investigación**:

- El **analítico-sintético** para recopilar toda la información relacionada con las bases teóricas de las tecnologías Plone/Zope.
- **Entrevistas** realizadas a los periodistas de los diferentes medios de prensa para conocer los principales servicios que debe brindar un sitio web informativo.  
También se realizaron entrevistas a los periodistas, redactores y editores de la edición digital del periódico Trabajadores, con el objetivo de llegar a un entendimiento común del funcionamiento que debe tener el sistema a implementar.
- La **modelación**, método teórico utilizado para crear abstracciones que ayuden a explicar realmente como funcionaría el periódico Trabajadores cuando se encuentre ya publicada su versión final. De esta forma se podría determinar qué características y qué servicios serían necesarios incluir en el desarrollo del proyecto, además de predecir la respuesta de nuestro sistema a variaciones de algunos de sus parámetros sin tener que ejecutar estas actividades en la realidad.
- El **sistémico**, utilizado para estudiar la estructura de un sitio informativo, sus componentes y las relaciones que se establecen entre cada uno de ellos, comprendiendo así el funcionamiento y la dinámica que debe verse representada en el producto final. De esta forma quedará definido el comportamiento de cada uno de los componentes del sistema y la dependencia existente entre ellos, convirtiéndose dichas relaciones en las leyes que regirán el desarrollo del proyecto.

# Capítulo 1. Fundamentación Teórica

---

En este capítulo se resumen las principales características de los sitios web informativos, de los servicios más comúnmente brindados por estos y de su comportamiento en la prensa digital cubana. Además se aborda el tema de los CMS (Content Management System), en particular de la tecnología Zope/Plone que será utilizada para el desarrollo del portal del periódico Trabajadores. Se presenta Scrum como la metodología ágil a utilizar durante el desarrollo del proyecto.

## Sitios web informativos.

El principal objetivo de este tipo de sitios web informativos es brindar información a los usuarios de forma tal que la navegación a través de estos sea lo más amigable posible. Producto de la gran aceptación y los grandes resultados que muestra a nivel internacional este tipo de formato para sitios web, actualmente resulta difícil encontrar un medio de comunicación, ya sea de radio, prensa o televisión, que no tenga una versión digital en Internet ya que constituye una vía fácil de llegar a gran número de personas en corto tiempo.

Un ejemplo claro de la utilización de los sitios informativos es el caso de la prensa escrita donde prácticamente todos los periódicos de cierta importancia tienen representación digital. De esta forma la información es actualizada minuto a minuto, siendo así poseedores siempre de las últimas noticias.

Con los sitios web informativos se logra una mejor comunicación con los lectores ya que puedes conocer su opinión mediante encuestas o foros, algunos de los servicios más comunes brindados en sitios de este tipo, ayudando así a los trabajadores del medio en cuestión a saber cómo hacer llegar mejor la información. Surge entonces la preocupación por acaparar la atención del mayor número de usuarios posible jugando en este sentido un papel importante el grado de usabilidad y accesibilidad del portal.

## Usabilidad y Accesibilidad.

No podemos definir usabilidad con accesibilidad, pero de igual forma tampoco se concibe separar un concepto del otro ya que ambos se complementan, están íntimamente relacionados.

Ambos aspectos van en busca del mismo objetivo, optimizar la efectividad, eficiencia y satisfacción de los usuarios. Mientras más accesible sea un portal más usable será.

La usabilidad se enfoca más al público objetivo del sitio web. De esta forma un sitio orientado a desarrolladores profesionales puede hacer uso de funciones avanzadas de búsqueda y emplear un lenguaje más técnico que otro orientado a un público más general.

Según Nielsen un concepto simple y eficaz de usabilidad es el siguiente:

*“Usabilidad es un atributo de calidad que mide lo fáciles de usar que son las interfaces web.”*(GARCÍA 2004)

Para desarrollar un sitio con buena usabilidad debemos centrarnos en las necesidades del usuario y tener en cuenta los siguientes aspectos:

- Pensar en qué modo el usuario desearía encontrar la información.
- Tener en cuenta las limitaciones de los recursos del usuario a nivel de ancho de banda, capacidad de procesamiento y resolución.
- Diseñar el grado de profundidad que daremos a nuestros contenidos, y si el usuario podrá avanzar en complejidad a medida que lo desee.
- Definir un árbol de navegación atractivo como referencia de navegabilidad en nuestra página web, no abrumando con excesivos enlaces, y mostrándolos a medida que el usuario se interesa por ellos.
- Definir una iconografía y un diseño visual con importante carga semántica que transmita justo lo que nos proponemos, para así poder ayudarnos del impacto visual en lugar de que éste produzca confusión.(BARRANCO 2004)

Los aspectos antes enumerados son solo pequeñas consideraciones sobre este tema, existen una gran cantidad de reglas y técnicas para dotar a los sitios de una alta usabilidad.

Por otro lado la accesibilidad está orientada a ampliar el número de usuarios que, con éxito, logran acceder al sitio web.

Yusef Hassan y F. Jesús Martín plantean que:

*“Accesibilidad Web es la posibilidad de que un producto o servicio web pueda ser accedido y usado por el mayor número posible de personas, indiferentemente de las limitaciones propias del individuo o de las derivadas del contexto de uso.”* (GARCÍA 2004)

La gran importancia que le es concedida actualmente a la Web ha creado la necesidad de que esta sea accesible, proporcionando así un acceso equitativo e igualdad de oportunidades para cualquier tipo de usuario. Un portal web accesible ayuda a integrar más a la sociedad a personas con alguna discapacidad.

La accesibilidad web depende de diferentes componentes que trabajan en conjunto para obtener mejoras que podrían hacer el sitio más accesible. Entre estos componentes se incluyen:

- **Contenido:** Es la información que será mostrada en el sitio. Esto incluye imágenes, videos, textos; además de la estructura y presentación del sitio.
- **Navegadores Web, reproductores, multimedias:** Herramientas utilizadas por los usuarios para visualizar contenidos web.
- **Tecnología de asistencia:** Lectores de pantalla, teclados alternativos, intercambiadores, software de escaneo, entre otros.
- **Conocimiento del usuario:** Experiencia en el uso de la Web.
- **Desarrolladores:** El equipo de trabajo encargado de desarrollar la aplicación.
- **Herramientas de autor:** Software empleado para el desarrollo de aplicaciones Web.
- **Herramientas de evaluación:** Herramientas que evalúan los índices de accesibilidad Web, y otras que validan el HTML y el CSS del sitio.



Figura 1. Relación entre componentes

## **Servicios brindados por sitios web informativos.**

Una de las ventajas que trae Internet para la prensa digital es la de brindar en sus sitios servicios de valor agregado que no pueden ser ofrecidos en la prensa escrita. Estos servicios añaden valor a la información y hacen más dinámica la interacción con los usuarios, atrayendo así a mayor número de ellos.

Los servicios de valor agregado son *“aquellos servicios que se basan en el uso de las tecnologías y en el grado de interactividad que ofrece la Red para proporcionar al lector utilidades y fuentes de información que no están al alcance de los diarios convencionales por las limitaciones tecnológicas intrínsecas de sus versiones impresas.”*(REYES)

Con la utilización de los servicios de valor agregado la prensa digital logra enriquecer la capacidad informativa y comunicativa de sus publicaciones aumentando el prestigio de la entidad y contribuyendo a transformar los periódicos en sistemas integrados de información y comunicación.

En el caso de la prensa digital cubana, a pesar del gran esfuerzo que se está realizando para sacarla adelante, aún es muy limitada la oferta de los servicios de valor agregado. Por tanto, uno de los principales objetivos del proceso de informatización de la prensa es lograr una homogeneidad en estos servicios, para lograr obtener una publicación digital de cada periódico.

Los servicios más comunes en los medios de la prensa digital cubana son:(REYES)

- **Foros de discusión:** Aplicaciones web que permiten intercambiar ideas y puntos de vistas sobre diversos temas. Pueden ser moderados por un administrador que se encarga de decidir los temas de discusión que serán publicados y de mantener el orden en el debate.
- **Encuestas de opinión:** Conjunto de preguntas dirigidas a conocer estadísticamente la opinión de los usuarios del sitio sobre un tema determinado.
- **Boletines informativos:** Publicación centrada en un tema de interés para sus suscriptores, distribuida de forma periódica, por lo general mediante el uso del correo electrónico. Su principal objetivo es informar a los usuarios de las más recientes actualizaciones.
- **RSS (Really Simple Syndication):** Formato XML que se utiliza para difundir contenidos de un sitio web. Los usuarios deben suscribirse a este servicio y de esta forma obtendrán un resumen de los contenidos que ha sido actualizados en dicho portal.
- **Sistemas de búsqueda de información:** Es un sistema que realiza búsquedas en los archivos almacenados en un servidor web. Las búsquedas pueden ser por temas o palabras claves, al final

de esta el sistema devolverá un conjunto de enlaces a los contenidos que se relacionan con la búsqueda realizada.

- **Hemeroteca:** Por hemeroteca se entiende el conjunto total de las ediciones publicadas en un medio de prensa, organizada y procesada, desde donde se accede a un número o varios de ellos con el propósito de ser consultados.

## Sistema Manejador de Contenidos.

Construir y mantener un sitio web puede ser un trabajo muy engorroso si no se cuenta con las herramientas necesarias. Hasta hace muy poco tiempo solamente se utilizaban editores mediante los cuales se podían desarrollar páginas web. Estos editores estaban orientados solamente a la creación de sitios, una vez que un sitio estaba creado se hacía muy difícil su mantenimiento. Debido a esto ha nacido el concepto de Sistema de Gestión de Contenido o CMS (*Content Management Systems*), que no son más que herramientas que permiten crear, actualizar, dar mantenimiento y ampliar un sitio web de forma fácil y eficiente con la ayuda de múltiples usuarios.

James Robertson propone una división en cuatro categorías de las funcionalidades de los sistemas de gestión de contenidos (GARCIA and ALFONSO 2004), las cuáles explicaremos a continuación:

- **Creación de contenido:** Un CMS facilita que los usuarios sin experiencia en el campo de la programación web puedan concentrarse solamente en el contenido que se mostrará en el sitio. Por lo general esto se lleva a cabo mediante editores WYSIWYG (What You See What You Get), mediante los cuales los usuarios pueden ver el resultado final mientras editan el documento. Además se aportan herramientas para definir la estructura, el formato, el aspecto visual y la posibilidad de adicionar nuevas funcionalidades al sitio.
- **Gestión de contenido:** Los contenidos creados permanecen en una base de datos donde también se almacenan otros datos necesarios para el funcionamiento del sitio. La estructura del sitio puede ser manejada mediante herramientas que normalmente proporcionan los CMS, teniendo de esta forma una visión jerárquica del sitio haciendo más fácil las modificaciones, aprovechando esto se puede asignar responsables a cada área del sitio, facilitando la actualización de este. También se brindan herramientas que permiten la creación de usuarios y grupos de usuarios con diferentes permisos, lo cual garantiza que se pueda implementar un correcto flujo de trabajo.



- **Publicación:** Los contenidos que ya han sido aprobados se publican de forma manual o automática cuando llega su fecha de publicación y cuando caducan son archivados en la base de datos en vista a ser utilizados en futuras referencias. Este proceso se rige por el flujo de trabajo definido previamente para el sitio en cuestión. Es importante que exista una separación entre el contenido y su aspecto visual, de esta forma se garantiza que se pueda cambiar el diseño del sitio sin afectar los documentos ya creados.
- **Presentación:** Algunos CMS traen soporte para normas internacionales de accesibilidad, gestionando así automáticamente la accesibilidad del sitio. De igual forma proporcionan compatibilidad con diferentes navegadores y plataformas, además de brindar la posibilidad de adaptar el idioma al gusto del visitante.

## **Plone y Zope.**

Zope (*Z Object Publishing Environment*), es una plataforma utilizada como herramienta para el desarrollo rápido de aplicaciones web ricas en contenido y funcionalidad. Tiene una arquitectura muy completa y compleja. Es considerado por sus creadores como un servidor de aplicaciones donde se da mayor importancia a la gestión de contenidos, portales y servicios.

La licencia de Zope, ZLP, es calificada por la *Free Software Foundation* como software libre. Esta licencia ha permitido que un gran número de desarrolladores puedan utilizar Zope sin costo alguno lo cual también posibilita el surgimiento y crecimiento de comunidades de desarrollo que tienen como objetivo incrementar y mejorar sus funcionalidades.

La arquitectura de Zope esta basada en diferentes mecanismos de intercambio de datos con el servidor Zope y un conjunto de herramientas de apoyo a ese servidor Zope. La siguiente imagen muestra el esquema de la arquitectura de Zope según sus desarrolladores. (Zope Architecture)

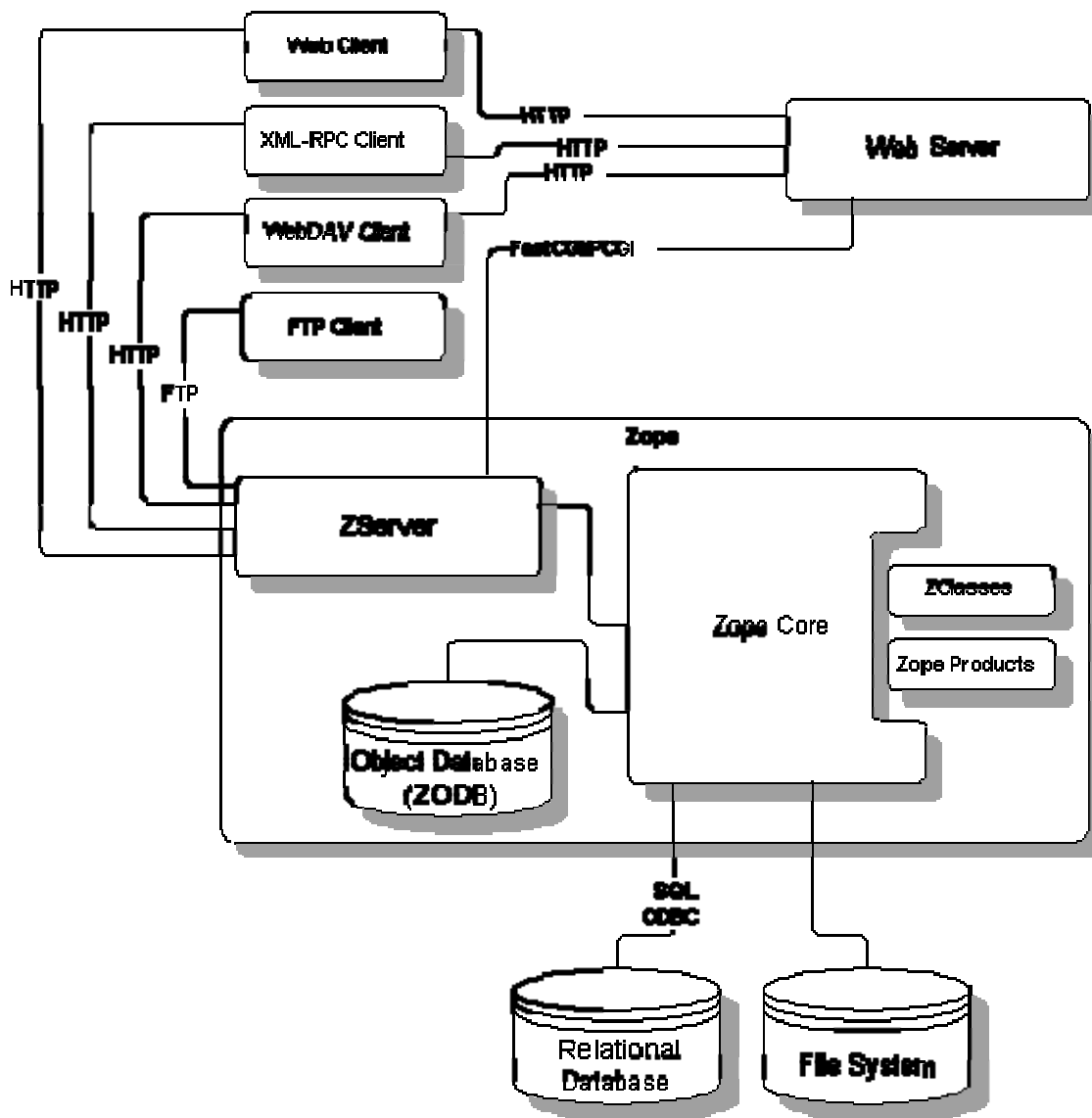


Figura 2. Arquitectura de Zope

Una de las principales características de Zope es que todos sus elementos son objetos, producto de esto para su implementación se eligió el lenguaje de programación Python y su base de datos, ZODB (*Zope Object Database*), es orientada a objetos.

ZODB, al permitir el almacenamiento de forma persistente de los objetos, proporciona grandes facilidades a la hora de modificar y buscar datos dentro de todos los objetos que forman el sitio web. Además al ser

orientada a objeto es mucho más adecuada para la elaboración de jerarquías necesarias para las aplicaciones de gestión de contenidos.

Sobre la plataforma Zope se ha desarrollado el CMS Plone. Este sistema manejador de contenido es desarrollado basado en código abierto y con él pueden crearse fácilmente sitios web corporativos, sitios de noticias, servidores de intranet o extranet, sistemas de publicación y repositorios de documentos. Es una herramienta muy potente para la publicación y gestión de contenidos.

Plone brinda grandes ventajas a sus usuarios:

- Gracias a sus “carpetas inteligentes” y “flujos de trabajo” permite adaptarse a múltiples funciones.
- Proporciona también un entorno gráfico tipo web que permite interactuar con el servidor a través de un navegador web desde cualquier lugar.
- Los usuarios pueden editar las páginas del sitio web en tiempo real.
- Presenta herramientas que ayudan a mejorar el trabajo en comunidad.
- Cientos de desarrolladores forman parte del Plone Development Team alrededor del mundo, y varias empresas se especializan en desarrollar y dar soporte a Plone.
- Existen una gran cantidad de productos desarrollados para agregar sobre Plone con el objetivo de adicionar nuevas funcionalidades y tipos de contenidos al CMS.
- En Plone se puede gestionar gracias a la base de datos de Zope el historial de las acciones que han sido realizadas en el sitio, esto permite deshacer los últimos cambios realizados en caso de necesidad.
- Plone puede interactuar con diferentes bases de datos relacionales, sean estas de código abierto o de uso comercial.
- La licencia utilizada por Plone es GPL (General Public License), la cual autoriza a utilizar Plone y a beneficiarse de futuras mejoras sin costo alguno.
- Plone cumple rigurosamente con los estándares para usabilidad y accesibilidad. Además de acatar las pautas de accesibilidad del W3C (World Wide Web Consortium) para utilizar estándares basados en mejoras prácticas como XHTML y CSS.
- La interfaz de Plone ha sido traducido a más de 20 idiomas. Plone detecta la configuración de su navegador, activando el idioma correspondiente.

Debido a las ventajas antes planteadas Plone es considerado un CMS ideal para la gestión de sitios donde sea necesaria la actualización y modificación constante de contenidos.

La prensa cubana esta llevando adelante el proceso de digitalización de sus sitios web con la utilización de este sistema manejador de contenidos ya que además de los grandes beneficios que brinda, su extensibilidad permite la creación de nuevos productos que cubran las necesidades actuales de los periódicos digitales.

## **Metodologías de desarrollo.**

Las metodologías de desarrollo del software son una serie de procedimientos, técnicas y ayudas a la documentación para el proceso de desarrollo de productos de software.

La utilización de metodologías ayuda a los desarrolladores indicando los pasos a seguir para lograr el producto final deseado, describiendo además que papel debe desempeñar cada miembro del equipo en el desarrollo de las actividades del proyecto. Además cada metodología define la información que se debe tener en cuenta antes de comenzar cada actividad y los entregables que se deben producir como resultado de dichas actividades.

Hasta hace muy poco tiempo el proceso de desarrollo del software brindaba vital importancia al control del proceso mediante rigurosas definiciones de roles, actividades y artefactos incluyendo modelado y documentación detallada. Este esquema tradicional ha demostrado ser necesario para el desarrollo de grandes proyectos. Sin embargo, en muchos de los proyectos actuales, más pequeños en cuanto a recursos y tiempo, en los que se sigue exigiendo una alta calidad, estas metodologías tradicionales no siempre son las más adecuadas. Es ante esta situación que surgen las metodologías ágiles, orientadas a proyectos pequeños sin renunciar a las prácticas esenciales para asegurar la calidad del producto.

De acuerdo a las características del proyecto a realizar, con el cual se pretende agilizar el proceso de informatización de la prensa cubana y que además cuenta con un pequeño equipo de desarrollo, se decidió emplear para dirigir el proceso una metodología ágil. Existen diferentes metodologías ágiles, entre ellas podemos citar:

- XP o Programación Extrema.
- Scrum.
- RUP (Rational Unified Process) Ágil.
- Crystal
- Evo (Evolutionary Project Management)
- ASD (Adaptive Software Developmen)

Al realizar un estudio y análisis de las metodologías antes mencionadas y teniendo en cuenta las necesidades del proyecto se decidió, dadas sus características, que la más adecuada para dirigir el proceso de desarrollo del producto era *Scrum* (Ver Anexo I.), ya que prioriza el trabajo en función del valor que este tenga para el negocio, maximizando de esta forma la utilidad de lo que se construye. Esta metodología esta diseñada para adaptarse a los cambios de los requerimientos, lo cual es muy útil ya que en el caso del proceso de informatización de la prensa y más específicamente en el desarrollo del portal web para el periódico Trabajadores constantemente surgen nuevos requerimientos y necesidades, los cuales, a pesar de no estar incluidos en la fase inicial del proyecto, deben ser también tomados en cuenta para el desarrollo del producto final. De esta forma los requerimientos y las prioridades se revisan y ajustan durante el desarrollo del software en intervalos muy cortos y regulares, adaptando así en tiempo real el producto que se está construyendo a las necesidades del cliente.

*Scrum* se basa en los principios de inspección continua, adaptación, auto-gestión e innovación. En cada una de sus iteraciones se obtienen entregables que hacen visibles los avances alcanzados, logrando así estimular y comprometer al cliente con el proyecto ya que nota el crecimiento del producto.

Concretando lo expuesto en este capítulo, el portal del periódico Trabajadores se desarrollará utilizando el CMS Plone, seleccionado por el Centro de Información de la Prensa para llevar adelante el proceso de digitalización de la prensa cubana. Como metodología a utilizar para el desarrollo se utilizará *Scrum*, metodología ágil que se adecua a las características de trabajo del proyecto.

## Capítulo 2. Características del Sistema

---

### **Antecedentes del periódico Trabajadores.**

Antes del 6 de junio del 2007 (día en el que sale al aire el nuevo sitio del periódico Trabajadores, desarrollado utilizando el CMS Plone) el sitio web del periódico Trabajadores era estático y no seguía los estándares establecidos para la utilización correcta de HTML y CSS. La actualización de la información de este portal por parte de los periodistas se hacía muy engorrosa ya que tenían que subir los archivos a un servidor mediante el protocolo FTP (*File Transfer Protocol*). El diseño visual del sitio no cumplía con las normas de accesibilidad y usabilidad lo cual traía como consecuencia que se hiciera muy difícil navegar el mismo.

Debido a estos problemas el sitio no alcanzaba un buen posicionamiento en los buscadores de Internet. Al buscador de Google, por ejemplo, se le hacía imposible indexar el portal debido a la estructura de este basada en *frames*.

Ante esta situación se determinó que el sitio web del periódico Trabajadores sería uno de los primeros en migrar a Plone. Es entonces que como resultado de la primera fase de desarrollo de este proyecto se publica en junio del 2007 la nueva versión del portal, con una nueva interfaz y desarrollado en su totalidad sobre el CMS Plone.

Este nuevo sitio cumplía ya con estándares de HTML y CSS. En su diseño fueron tomados en cuenta normas de accesibilidad y usabilidad, logrando una interfaz más amigable a los usuarios. Los problemas de actualización del sitio fueron eliminados aplicando el flujo de trabajo básico del CMS Plone. A partir de este momento fue mejorando el posicionamiento del portal en Internet aunque aún no se logran los resultados esperados.

A finales del año 2007 y principios del 2008 el nuevo sitio comienza a presentar problemas de extrema lentitud y se hace necesario realizar una investigación en búsqueda de las posibles causas. Los errores detectados, unido a inconformidades manifestadas por parte de los periodistas del medio y a la necesidad de hacer de este sitio un sitio modelo mediante el cual se pudiera agilizar el proceso de desarrollo de portales web en el proyecto de Informatización de la Prensa, dan lugar a una segunda etapa de desarrollo.

## Estructura del sitio.

La versión digital web del periódico Trabajadores brinda a sus usuarios materiales noticiosos, servicios, suplementos y otras informaciones. Estas se encuentran distribuidas de la siguiente forma en el portal:

- **Secciones principales:** En estas secciones son agrupadas las noticias según su temática. Relacionadas con el tópico de la sección también se brindan otras informaciones e incluso se pueden encontrar nuevas secciones.
  - **Cuba:** En esta sección encontramos noticias relacionadas con el ámbito nacional, además de otras secciones como:
    - **Cuba por dentro:** Contiene materiales relacionados con la cultura, la historia y lugares de Cuba.
    - **Cuba en imágenes:** Se brindan galerías de imágenes de nuestro país.
  - **Mundo:** En esta sección encontramos noticias relacionadas con el acontecer internacional.
  - **Deportes:** Esta sección agrupa noticias relacionadas con el mundo del deporte, además contiene la sección “Protagonistas” que publica entrevistas realizadas a glorias y personalidades del deporte cubano.
  - **Cultura:** Contiene noticias relacionadas con el ámbito cultural.
- **Suplementos:** Los suplementos brindan información a los usuarios acerca de un tema. Actualmente existen 3 suplementos:
  - Salud.
  - Mundo Laboral.
  - Memoria Histórica.
- **Servicios de valor agregado:** Como todo portal informativo el sitio del periódico Trabajadores brinda servicios con el objetivo de incrementar la interacción con sus usuarios, además de brindarle mayores funcionalidades y facilidades. Entre los servicios brindados por el portal encontramos:
  - **Boletines informativos:** Permiten el envío de noticias directamente al buzón de correo electrónico de los usuarios suscritos.

- **Encuestas de opinión:** A través de estas los usuarios expresan su opinión acerca de un tema determinado y a través de los resultados conocen la opinión general de las personas que acceden al sitio.
- **RSS (Really Simple Syndication):** Se utiliza para difundir contenido del sitio a los usuarios que estén suscritos a este servicio.
- **Foros de discusión:** A través de este foro los usuarios proponen temáticas para la discusión y expresan su opinión sobre diferentes temas propuestos por otros usuarios. El foro es moderado por los administradores del sitio para garantizar el correcto uso de la herramienta.
- **Buscador:** Facilita a los usuarios la búsqueda de información en el sitio dándole diferentes opciones de búsqueda para así ofrecer mejores resultados.
- **Consultas:** El sitio brinda consultas sobre 3 temáticas: Legislación laboral, Psicología y Sexualidad. De esta forma se permite a los usuarios consultar acerca de las temáticas antes relacionadas a especialistas en el tema que se dedican a brindar consejos y respuestas a través de este servicio en el sitio web.
- **Otras secciones:**
  - **Columnistas:** En esta sección el usuario puede encontrar las columnas publicadas en el sitio según el autor de estas del cual también se puede encontrar una breve descripción biográfica.
  - **Blogs de nuestros periodistas:** Donde se publican las direcciones web de los blogs más destacados de los periodistas del periódico Trabajadores.
  - **Coberturas:** En esta sección los usuarios pueden encontrar coberturas periodísticas a eventos acontecidos tanto a nivel nacional como internacional.
  - **Infografías:** Presenta, a través de imagen y sonido, diferentes trabajos relacionados con temas de posible interés a los usuarios.
  - **Multimedia:** En esta sección los usuarios pueden acceder a multimedias publicadas en el sitio teniendo la posibilidad de visualizarlas on-line o descargarlas si lo desean.



## Productos.

Una de las ventajas que ofrece Plone es la facilidad con la que se puede adaptar a las necesidades de cada usuario mediante la creación de nuevos productos. En torno a este tema existe una creciente comunidad de desarrollo que se dedica a mejorar y crear nuevas funcionalidades y tipos de contenidos para el CMS Plone.

En el caso del sitio web del periódico Trabajadores fueron empleados varios productos con el objetivo de cubrir todas las funcionalidades y servicios que demandaba el portal.

Entre los productos utilizados en el portal del periódico Trabajadores destacan:

- **Noticia de prensa:** Plone trae por defecto el producto NewsItem para el trabajo con materiales periodísticos, pero este no cumplía con los requerimientos básicos que debería tener un contenido de este tipo para el trabajo en la prensa cubana, por lo que se desarrolló el producto “Noticia de Prensa” tratando de cubrir al máximo las demandas de los periodistas de los medios informativos cubanos.
- **PloneGazette:** Este producto fue aplicado al sitio para garantizar el servicio de envío de boletines noticiosos a los usuarios de la versión digital del periódico.
- **CMFQuestions:** Con el objetivo de conocer la opinión de los usuarios sobre diferentes temas de actualidad, la edición digital del periódico demandó la posibilidad de realizar encuestas a través del sitio. Para dar cumplimiento a esta solicitud se empleó el producto CMFQuestions que permite la fácil creación y monitoreo de encuestas de opinión.
- **Prensa:** Este producto fue desarrollado para facilitar a los editores la ubicación de los materiales noticiosos en las diferentes secciones del sitio.
- **PloneBoard:** Utilizado para garantizar el servicio de consulta. De esta forma se intercambian preguntas y respuestas entre los usuarios y los encargados de las diferentes consultas.
- **ATMultimedia:** Para garantizar la publicación de archivos de audio y video se empleó este producto mediante el cual los usuarios pueden reproducir el video o el audio desde el propio sitio web o descargarlo si lo desean.

## Características del sitio modelo.

Debido a la necesidad de agilizar el proceso de digitalización de la prensa cubana, para lo cual fue seleccionado el CMS Plone, surge la idea de desarrollar un sitio que sirva como modelo para el rápido desarrollo de portales web destinados a los periódicos cubanos.

La creación de este sitio sería de gran ayuda debido a la gran dificultad que supone el desarrollo de portales utilizando Plone, esto unido a que existe muy poca documentación sobre la tecnología y a que no se cuenta con el suficiente personal cualificado para llevar adelante el proceso de informatización de la prensa utilizando el CMS Plone.

El sitio modelo debe ser accesible, usable y seguir estándares de programación web. Para su desarrollo se debe crear una nomenclatura común para definir el nombre de vistas, *portlets*, macros, clases, formularios y hojas de estilo facilitando de esta forma la comprensión y modificación del código.

### Definición de estándar de organización del código del sitio.

Para lograr una mejor organización del código fuente del sitio es necesario definir una estructura de carpetas estándar para almacenar estos archivos. Plone trae definida una estructura de este tipo la cual será aplicada en nuestro sitio:

- **content:** Contiene las vistas que se desarrollarán para cada uno de los objetos del portal.
- **forms:** Almacena los formularios que se emplearán en las plantillas del sitio.
- **images:** Guarda todas las imágenes que se emplean en el sitio web.
- **login:** Contiene todo el código relacionado con la autenticación en el sitio, tanto plantillas como formularios y scripts.
- **portlets:** En esta carpeta se guardan todos portlets que se utilizan en el sitio.
- **scripts:** Almacena todos los scripts que se emplean en el portal.
- **styles:** Guarda todas las hojas de estilo que se utilizan en el sitio.
- **templates:** Todos los templates que se emplean en el sitio son almacenados en esta carpeta.

Además de almacenar el código fuente en cada una de estas carpetas según su funcionalidad, en el sitio se hace necesario también establecer una nomenclatura común para nombrar cada tipo de archivo, para

esto se establecerá como estándar agregar al nombre del archivo un sufijo identificativo de la función que juega en el portal:

Tipo de archivo	Sufijo
vistas	_view
formularios	_form
imágenes	_image
portlets	_portlet
scripts	_script
hoja de estilo	_style.css
plantillas	_templates

## Product Backlog.

Generalmente el *product backlog* de un proyecto surge a partir de un listado de requisitos con los cuales debe cumplir el producto final. En el caso de esta segunda etapa de desarrollo del sitio del periódico Trabajadores se parte de una versión inicial del portal a partir de la cual se debe lograr un sitio modelo, por tanto el *product backlog* se crea orientándose a dar solución a una serie de deficiencias que fueron encontradas en la versión inicial y de solicitudes de cambio hechas por parte de la dirección del periódico. Todos estos cambios dirigidos a crear un sitio modelo con el cual se logre agilizar el proceso de desarrollo de portales web como parte del proyecto de informatización de la prensa.

ID	Descripción	Tiempo Estimado(h)
<b>Crítico</b>		
1	Permitir que los usuarios puedan votar en las encuestas sin estar autenticados previamente. *	4
2	Permitir que los usuarios puedan suscribirse a los boletines sin tener que autenticarse previamente en	4

	el portal.*	
3	Optimizar buscador del sitio.*	6
4	Modificar mapa del sitio para lograr que solo se muestren las carpetas que están públicas a los usuarios. *	1
5	Optimización del código fuente.*	30
6	Configurar el producto CacheFu.*	3
7	Validar que al quedar vacía una sección del sitio no se deje de mostrar el sitio a los usuarios. *	1
<b>Necesario</b>		
8	Crear vista para los materiales comentarios.*	4
9	Agregar banner "Terrorismo No".*	1
10	Brindar la opción de entrar al sitio en inglés *solamente desde la portada.	1
11	Mostrar los materiales relacionados en la vista de las noticias.*	1
12	Mostrar los materiales relacionados en la noticia principal de portada.*	1
13	Modificar el "cobertura_portlet" que se muestra desde la portada principal del sitio para que se muestre también el logo identificativo de la misma antecediendo al título de cada cobertura.	2
14	Crear una visa para la carpeta "Infografías"	4
15	Crear una visa para la carpeta "Coberturas"	4
16	Crear una visa para la carpeta "Mirada Gráfica"	4
17	Crear una visa para la carpeta "Multimedia"	4
18	Evitar que las fotos que se muestran desde portada en las secciones "De lo último" y "Destacado" se	1

	deformen.	
19	Modificar la vista de las imágenes ampliadas para que en la misma se muestre una descripción de la imagen.	1
20	Eliminar restricción a la hora de mostrar “Otras noticias principales” en las portadas del sitio.	1
21	Dar la opción de que las imágenes dentro del cuerpo del texto de las noticias puedan ser ampliadas.*	1
22	Modificar la sección “Cuba en imágenes” para que muestre desde portada la última galería publicada y la opción de ver el resto de las galerías.*	1
23	Eliminar de la portada la opción “Ver todos los titulares”	1
24	Eliminar el producto ATRatings	1
25	Corregir el uso del tag A y Li	4
26	Poner el atributo rel="nofollow" en los enlaces para los que sea necesario.	2
27	Corregir defecto de HTML en listados	3
28	Poner el tag META de "nofollow" en las vistas que muestran noticias.	3

Dados los antecedentes ya analizados del periódico Trabajadores, comienza una segunda etapa de desarrollo para dar solución a los problemas que fueron surgiendo desde la publicación de la nueva versión del portal en Internet. Esta etapa se basa en la lista de tareas expuesta anteriormente en este capítulo en el que también se puede encontrar una descripción detallada de la estructura del portal y de la propuesta inicial que se da para hacer del portal Web del periódico Trabajadores un sitio modelo.

## Capítulo 3. Implementación y Prueba

---

En este capítulo se detallan los *Sprint Backlogs* que fueron necesarios llevar a cabo para dar cumplimiento a las tareas definidas en el *Product Backlog*. Se describen también los diseños y resultados de las pruebas que fueron necesarias realizar a la nueva versión del portal del periódico Trabajadores a lo largo del desarrollo del mismo en esta segunda etapa.

### Planificación del proyecto.

Las responsabilidades de manejo de proyecto fueron distribuidas de la siguiente forma, siguiendo la filosofía de trabajo de la metodología *Scrum*.

Rol	Miembro
<i>product owner</i>	Manuel Vázquez Acosta
<i>Scrum master</i>	Ailyn Alfonso González
<i>team</i>	Joel Armada Herrera
<i>team</i>	Ailyn Alfonso Gonzalez
<i>team</i>	Karel A. Verdecia Ortiz
<i>team</i>	Javier Chirino Sanchez
<i>team</i>	Raidel Arenas Perez

### Sprints Backlogs.

A continuación se relacionan los *sprints backlogs* correspondientes a cada una de las iteraciones que se llevaron a cabo para dar cumplimiento a todas las tareas propuestas en el *product backlog*.

Sprint	Inicio	Duración
1	05-nov-07	1

Velocidad	0.8
-----------	-----

	L	M	M	J
	05-nov	06-nov	07-nov	08-nov
Tareas pendientes	1	1	1	0
Horas de trabajo pendientes	6	2	2	0
Fecha de cierre	10-nov	09-nov	09-nov	09-nov

Sprint Backlog										
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real				
#1	Permitir que los usuarios puedan votar en las encuestas sin estar autenticados previamente.	código	Terminada	Ailyn Alfonso	4	2				
#2	Permitir que los usuarios puedan suscribirse a los boletines sin tener que autenticarse previamente en el portal.	código	Terminada	Ailyn Alfonso	4	2				
#3	Optimizar buscador del sitio.	código	Terminada	Karel A. Verdecia	6	10	6	2	1	
#4	Modificar mapa del sitio para lograr que solo se muestren las carpetas que están públicas a los usuarios.	código	Terminada	Ailyn Alfonso	1					

Sprint	Inicio	Duración
2	15-ene-08	1

Velocidad	0.8
-----------	-----

	M	M	J	V
	15-ene	16-ene	17-ene	18-ene
Tareas pendientes	2	2	2	1
Horas de trabajo pendientes	17	12	6	2
Fecha de cierre	20-ene	21-ene	22-ene	19-ene

Sprint Backlog										
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real				
#5	Optimización del código fuente.	código	Terminada	Joel Armada y Ailyn Alfonso	10	15	11	8	4	
#6	Configurar el producto CacheFu.	código	Terminada	Karel A. Verdecia, Ailyn Alfonso, Joel Armada	5	6	6	4	2	2
#7	Validar que al quedar vacía una sección del sitio no se deje de mostrar el sitio a los usuarios.	código	Terminada	Joel Armada	4	2				



Sprint	Inicio	Duración
3	21-ene-08	1

Velocidad	0.5
-----------	-----

	L	M	M	J	V
	21-ene	22-ene	23-ene	24-ene	25-ene
Tareas pendientes	3	2	2	2	0
Horas de trabajo pendientes	9	4	2	2	0
Fecha de cierre	24-ene	24-ene	25-ene	25-ene	25-ene

Sprint Backlog											
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real					
#8	Crear vista para los materiales comentarios.	código	Terminada	Ailyn Alfonso	1	2					
#9	Agregar banner "Terrorismo No".	código	Terminada	Joel Armada	1						
#10	Brindar la opción de entrar al sitio en inglés solamente desde la portada.	código	Terminada	Ailyn Alfonso	1						
#21	Las imágenes dentro del cuerpo del texto de las noticias deben poder ser ampliadas.	código	Terminada	Ailyn Alfonso	1		1				
#22	Mostrar desde portada la última galería publicada y la opción de ver el resto de las galerías.	código	Terminada	Joel Armada	1						
#11	Mostrar los materiales relacionados en la vista de las noticias.	código	Terminada	Ailyn Alfonso	2	4	4	2	1	1	

#12	Mostrar los materiales relacionados en la noticia principal de portada.	código	Terminada	Ailyn Alfonso	2	4	4	2	1	1	
-----	---	--------	-----------	---------------	---	---	---	---	---	---	--

Sprint	Inicio	Duración
4	28-ene-08	1

<b>Velocidad</b>	0.6
------------------	-----

	L	M	M	J
	29-ene	30-ene	31-ene	01-feb
<b>Tareas pendientes</b>	2	1	1	0
<b>Horas de trabajo pendientes</b>	4	2	1	0
<b>Fecha de cierre</b>	01-feb	01-feb	01-feb	01-feb

<b>Sprint Backlog</b>										
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real				
#13	Modificar el "cobertura_portlet" para que se muestre también el logo identificativo de la misma antecediendo al título de cada cobertura.	código	Terminada	Ailyn Alfonso y Joel Armada	1	2				
#18	Evitar que las fotos de las secciones "De lo último" y "Destacado" se deformen.	código	Terminada	Joel Armada	1	4	3	2	1	
#19	Adicionar a la vista de la imagen ampliada la descripción de la misma.	código	Terminada	Ailyn Alfonso y Javier Chirino	1					
#20	Eliminar restricción de cantidad en "Otras noticias principales"	código	Terminada	Joel Armada	1		1			

#23	Eliminar de la portada la opción "Ver todos los titulares"	código	Terminada	Joel Armada	1						
-----	--	--------	-----------	-------------	---	--	--	--	--	--	--

Sprint	Inicio	Duración
5	05-feb-08	1

Velocidad	0.6
-----------	-----

	L	M	M	J	V	S
	05-feb	06-feb	07-feb	08-feb	09-feb	10-feb
Tareas pendientes	4	3	3	2	2	0
Horas de trabajo pendientes	16	9	5	2	2	0
Fecha de cierre	10-feb	10-feb	10-feb	10-feb	10-feb	10-feb

Sprint Backlog												
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real						
#14	Crear una visa para la carpeta "Infografías"	código	Terminada	Raidel Arenas	4	6	5	3	2	1	1	
#15	Crear una visa para la carpeta "Coberturas"	código	Terminada	Joel Armada	4	6	3					
#16	Crear una visa para la carpeta "Mirada Gráfica"	código	Terminada	Joel Armada	4	6	4	2	1			
#17	Crear una vista para la carpeta "Multimedia"	código	Terminada	Javier Chirino	4	6	4	4	2	1	1	

Sprint	Inicio	Duración
6	12-feb-08	1

Velocidad	0.5
-----------	-----

	L	M	M	J	V
	12-feb	13-feb	14-feb	15-feb	16-feb
<b>Tareas pendientes</b>	2	2	2	1	0
<b>Horas de trabajo pendientes</b>	6	5	2	1	0
<b>Fecha de cierre</b>	15-feb	15-feb	16-feb	16-feb	16-feb

<b>Sprint Backlog</b>											
ID	Tarea	Tipo	Estado	Responsable	Estimado	Real					
#24	Eliminar el producto ATRatings	código	Terminada	Ailyn Alfonso	1						
#25	Corregir el uso del tag A y Li	código	Terminada	Raidel Arenas	4	6	4	3	1	1	
#26	Poner el atributo rel="nofollow" en los enlaces para los que sea necesario.	código	Terminada	Javier Chirino	2						
#27	Corregir defecto de HTML en listados	código	Terminada	Manuel Vázquez	1						
#28	Poner el tag META de "nofollow" en las vistas que muestran noticias.	código	Terminada	Joel Armada	3		2	2	1		

## Pruebas

Al terminar cada *Sprint Backlog* se definieron un conjunto de casos de prueba a los que fue sometido el portal. Estas pruebas fueron dirigidas a comprobar el correcto desarrollo de determinadas tareas y a asegurar que con los cambios realizados en el código fuente, el sitio mejorara el tiempo de respuesta.

Para realizar las pruebas de estrés al portal se empleó la herramienta OpenSTA con la cual se simula la actividad de un número determinado de usuarios interactuando con el sitio simultáneamente.

Las pruebas se fueron realizando a lo largo del desarrollo del proyecto, basándose en sus resultados se fueron realizando cambios en el código fuente hasta lograr el resultado esperado.

### Prueba #1

Prueba de estrés realizada el 24 de enero del 2008 a la página principal del periódico Trabajadores.

#### Configuración:

- ✓ Total de usuarios virtuales: 100
- ✓ Intervalo entre cada envío: 12 segundos
- ✓ Cantidad de usuario por envío: 30

#### Tiempo total de ejecución de la prueba:

- ✓ 2:01 minutos.

#### Resultados:

- ✓ **Códigos de estado HTTP:** Todos los códigos de estado HTTP (*Hyper Text Transfer Protocol*) que se obtuvieron fueron 200 y 404 (Ver Anexo II). Este último siempre con la petición: <http://www.trabajadores.cu.dev/liveclock.js>

✓ Gráficas:

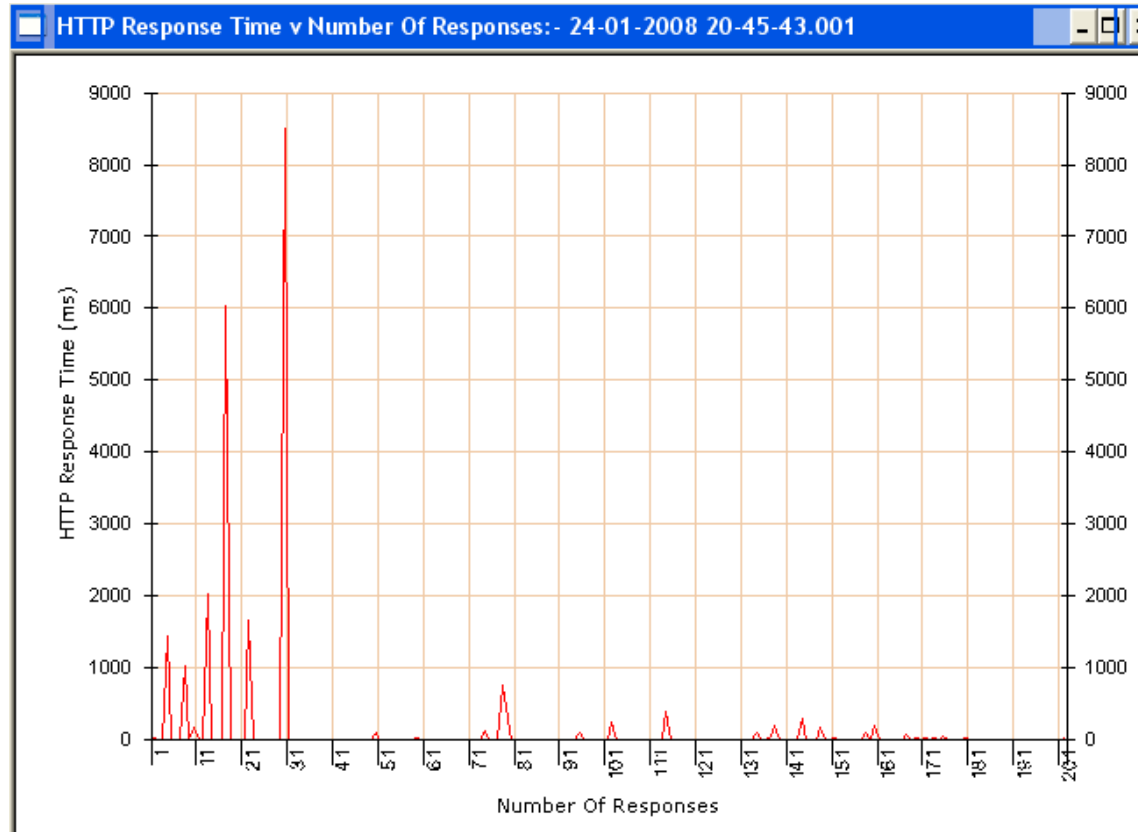


Figura 3. Número de Respuestas vs Tiempo de Respuestas

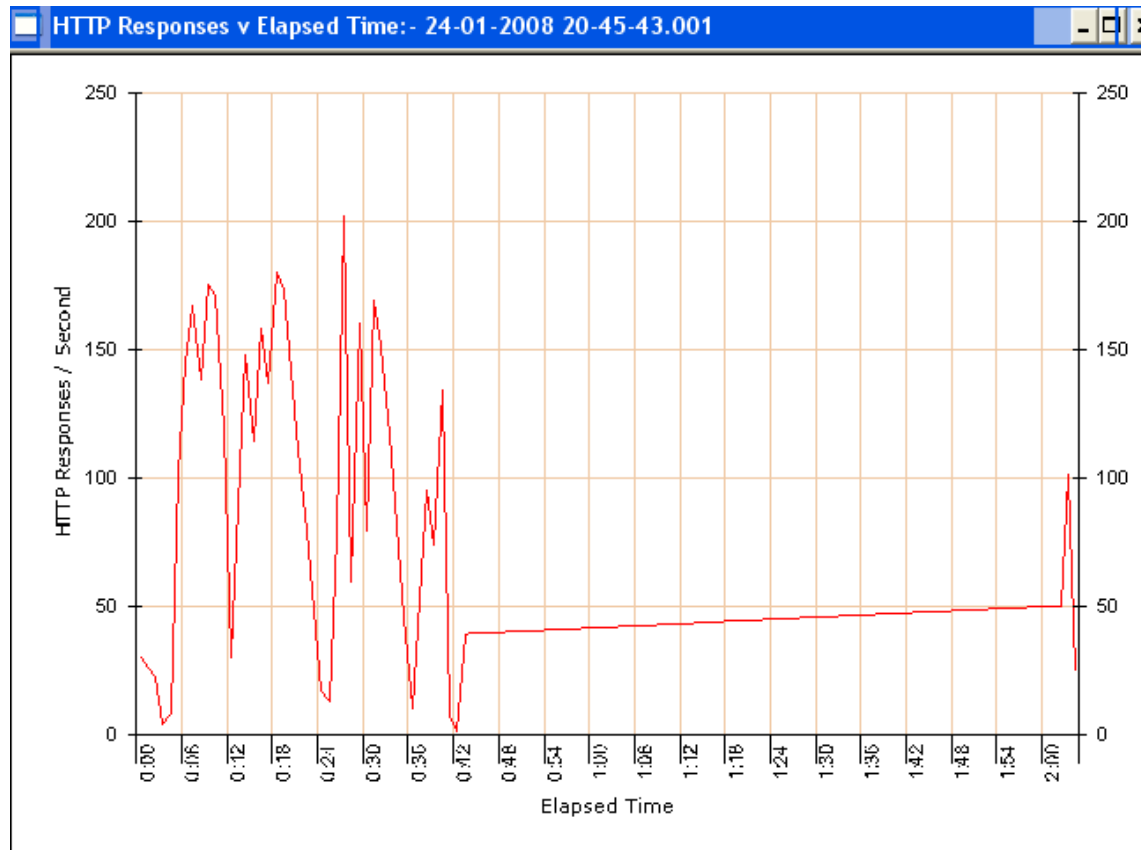


Figura 4. Tiempo Transcurrido vs Respuestas

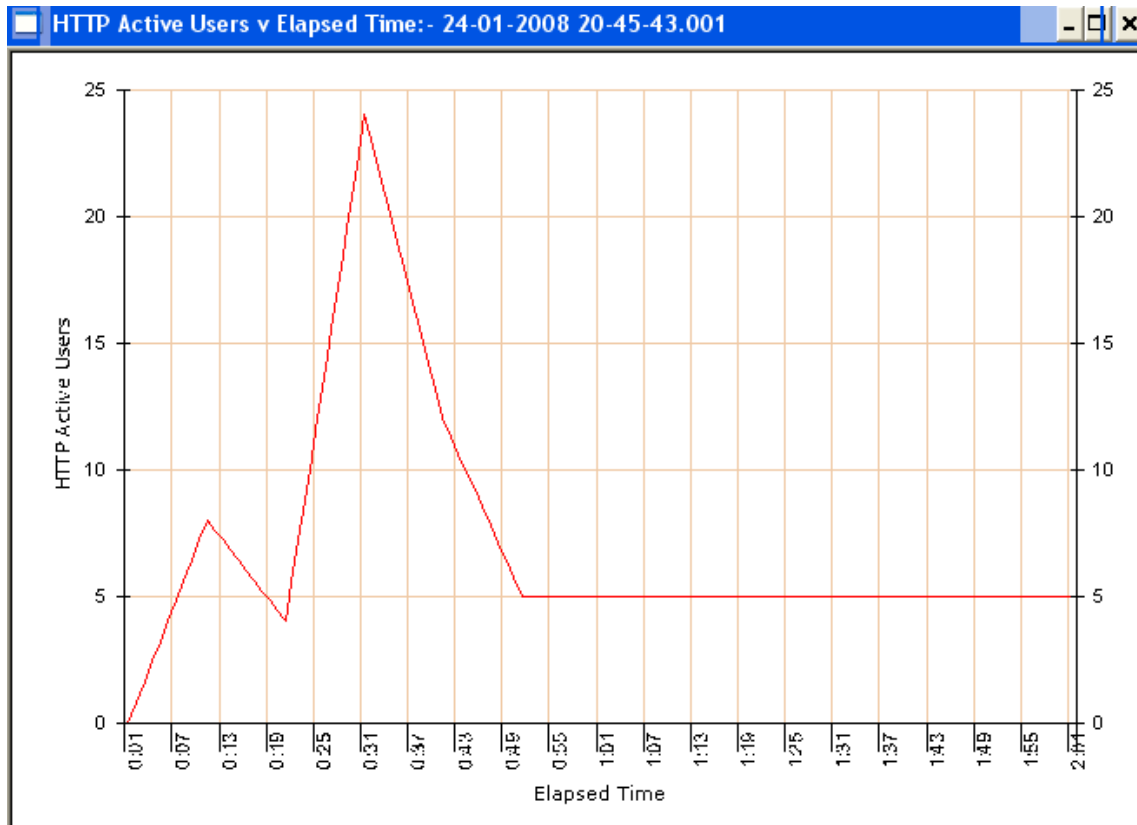


Figura 5. Tiempo transcurrido vs Usuarios activos

### Evaluación de la prueba:

Los resultados de la prueba no fueron satisfactorios ya que el portal no respondió a todas las solicitudes. Además de que se detectó que el archivo liveclock.js no se encontraba en el servidor.

### Prueba #2

Prueba de estrés realizada el 24 de enero del 2008 a la página principal del periódico Trabajadores.

### Configuración:

- ✓ Total de usuarios virtuales: 500.
- ✓ Intervalo entre cada envío: 12 segundos
- ✓ Cantidad de usuario por envío: 30

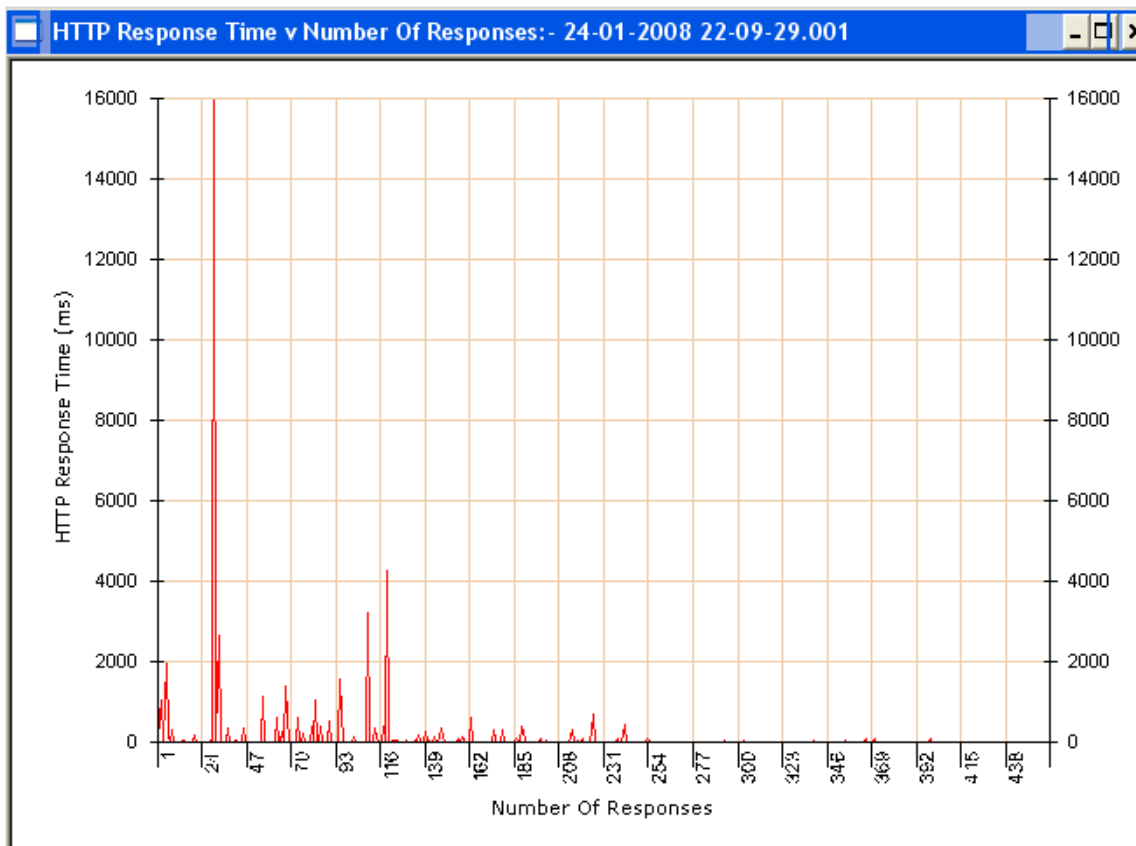


**Tiempo total de ejecución de la prueba:**

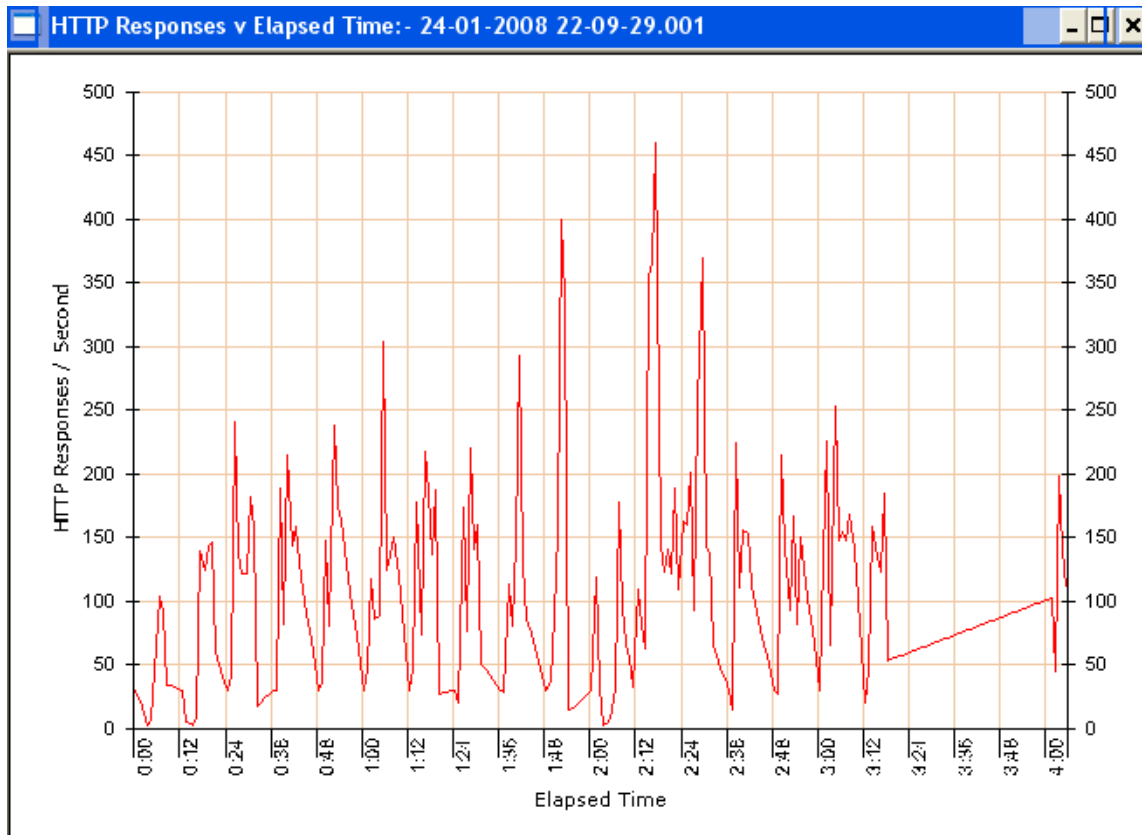
- ✓ 4:01 minutos.

**Resultados:**

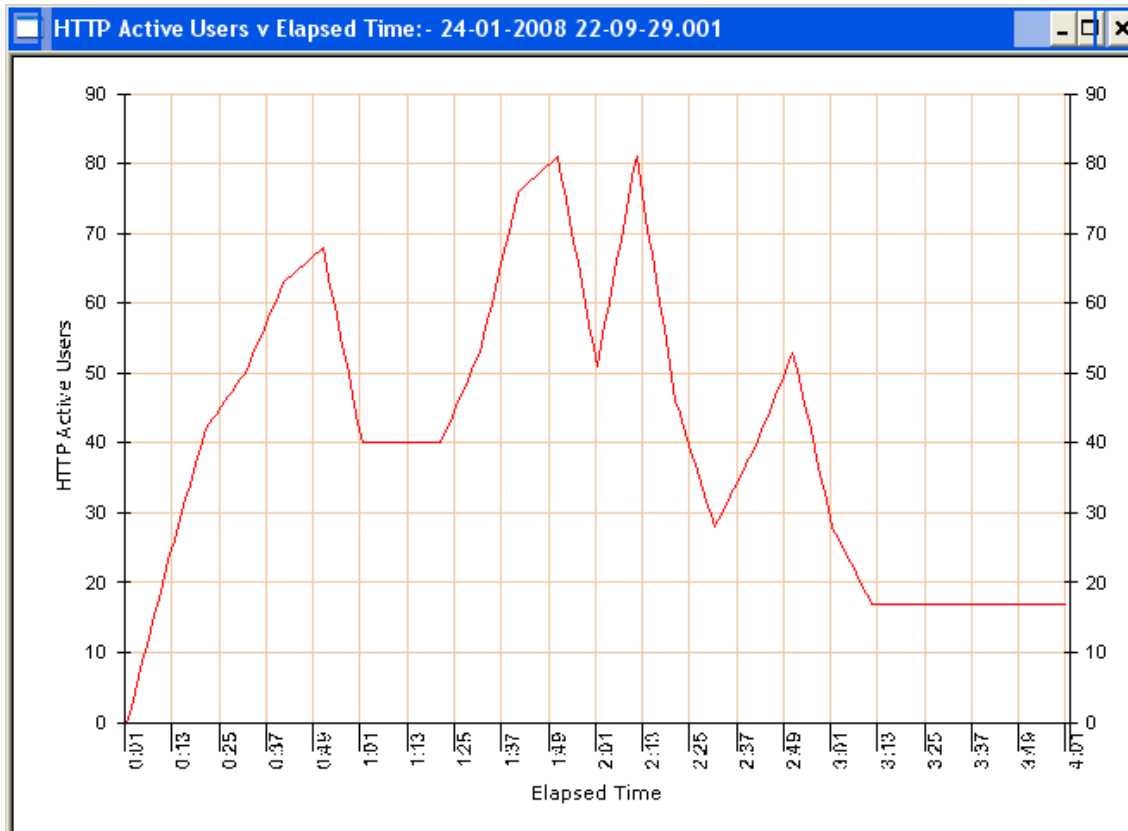
- ✓ **Códigos de estado HTTP:** Todos los códigos de estado HTTP (Hyper Text Transfer Protocol) que se obtuvieron fueron 200 y 404 (Ver Anexo II). Este último siempre con la petición: <http://www.trabajadores.cu.dev/liveclocks.js>
- ✓ **Gráficas:**



**Figura 6. Número de respuestas vs Tiempo de respuesta**



**Figura 7. Tiempo transcurrido vs Respuestas**



**Figura 8. Tiempo transcurrido vs Usuarios activos.**

### Evaluación de la prueba:

Al aumentar la cantidad de usuarios activos en el sitio el comportamiento del portal empeoró, teniendo tiempos de respuesta muy lentos, quedando un mayor número de usuarios sin ser atendidos. Además de que se detectó que continúa el problema con el archivo liveclock.js. El resultado de la prueba no fue satisfactorio.

### Prueba #3

Prueba de estrés realizada el 9 de mayo del 2008 a la página principal del periódico Trabajadores.

### Configuración:

- ✓ Total de usuarios virtuales: 100

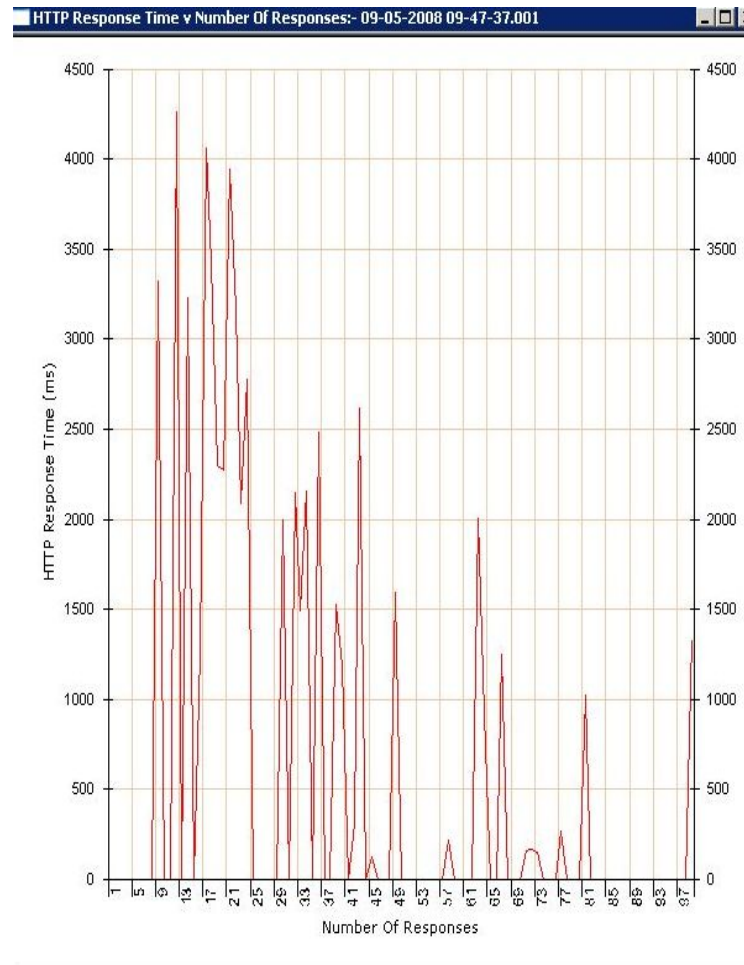
- ✓ Intervalo entre cada envío: 12 segundos
- ✓ Cantidad de usuario por envío: 30

**Tiempo total de ejecución de la prueba:**

- ✓ 50 segundos.

**Resultados:**

- ✓ **Códigos de estado HTTP:** Todos los códigos de estado HTTP (*Hyper Text Transfer Protocol*) que se obtuvieron fueron 200 (Ver Anexo II).
- ✓ **Gráficas:**



**Figura 9. Número de respuestas vs Tiempo de respuesta**

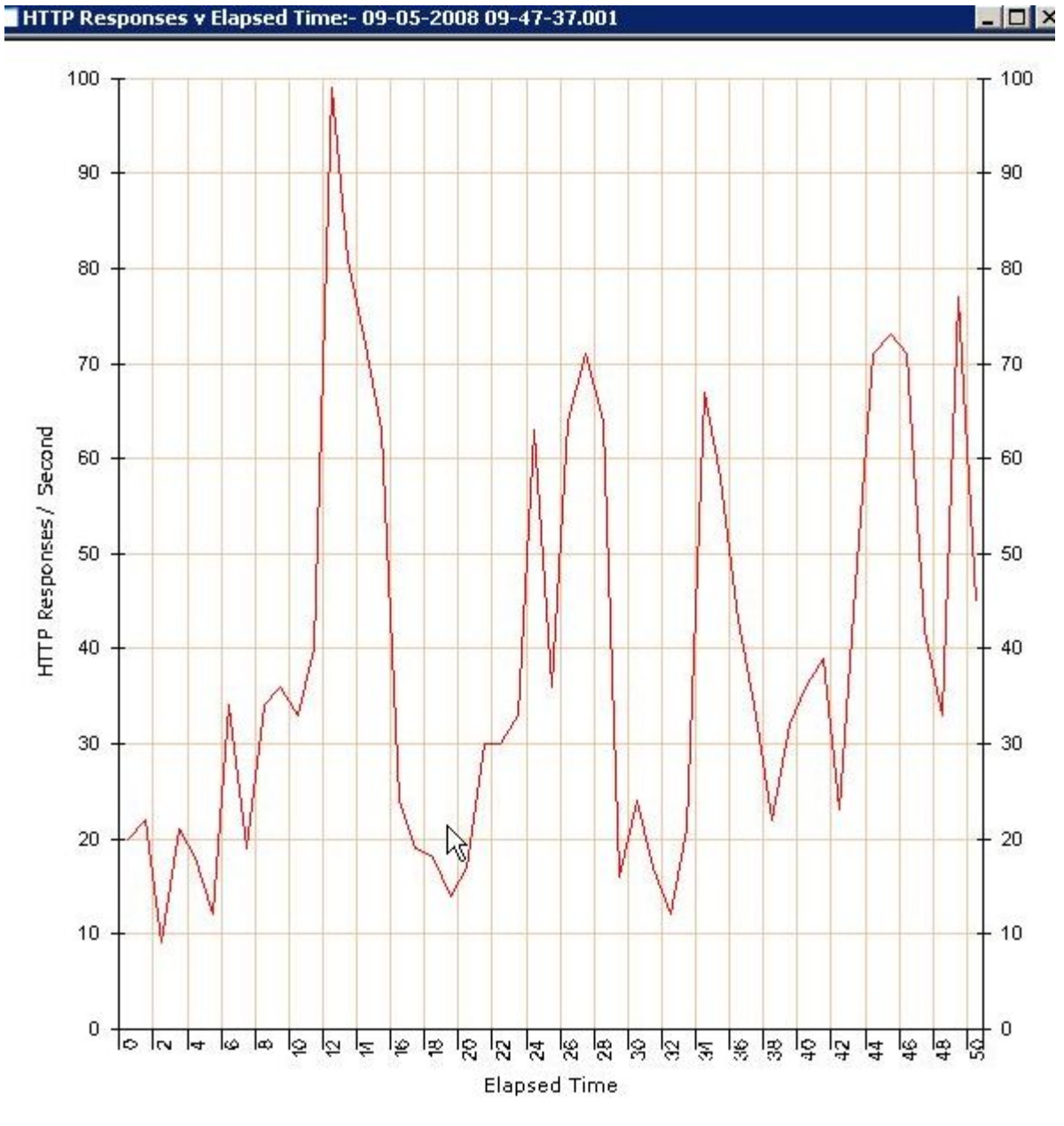


Figura 10. Tiempo transcurrido vs Respuestas



Figura 11. Tiempo transcurrido vs Usuarios activos.

**Evaluación de la prueba:**

Los resultados de la prueba fueron satisfactorios ya que el portal respondió a todas las solicitudes en un muy buen tiempo, demorando sólo 50 segundos en responder a los 100 usuarios.

## Prueba #4

Prueba de estrés realizada el 9 de mayo del 2008 a la página principal del periódico Trabajadores.

### Configuración:

- ✓ Total de usuarios virtuales: 500.
- ✓ Intervalo entre cada envío: 12 segundos
- ✓ Cantidad de usuario por envío: 30

### Tiempo total de ejecución de la prueba:

- ✓ 2:21 minutos.

### Resultados:

- ✓ **Códigos de estado HTTP:** Todos los códigos de estado HTTP (Hyper Text Transfer Protocol) que se obtuvieron fueron 200 (Ver Anexo II).
- ✓ **Gráficas:**

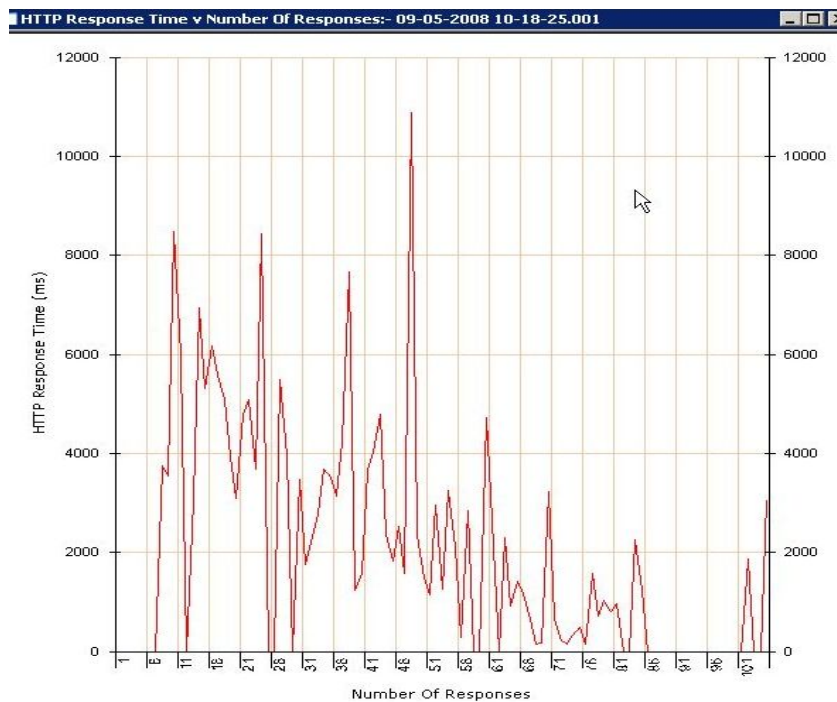


Figura 12. Número de respuestas vs Tiempo de respuesta

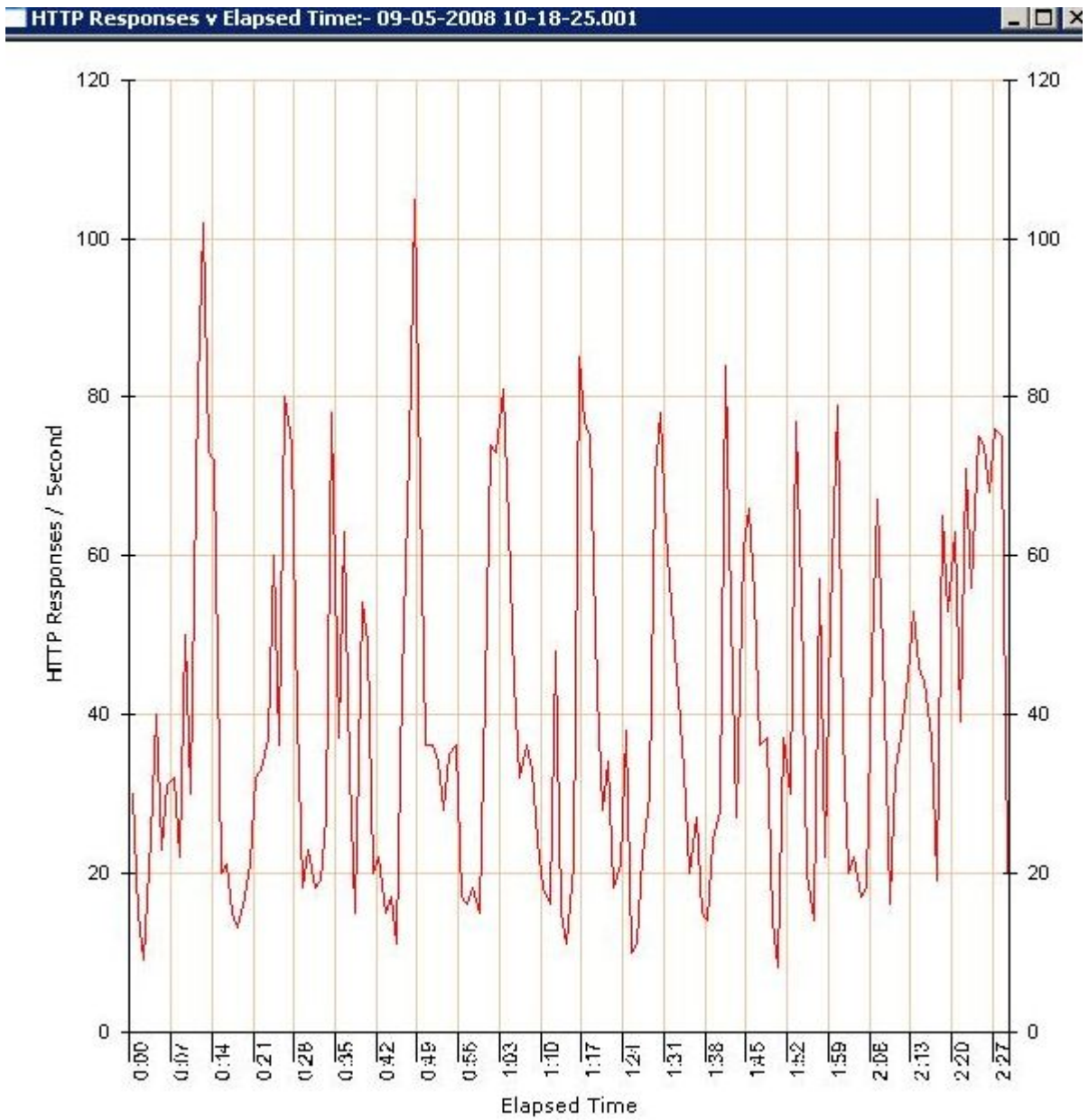


Figura 13. Tiempo transcurrido vs Respuestas





Figura 14. Tiempo transcurrido vs Usuarios activos.

**Evaluación de la prueba:**

Al aumentar la cantidad de usuarios activos en el sitio el comportamiento del portal se mantuvo estable, teniendo tiempos de respuesta muy buenos, los 500 usuarios fueron atendidos en 2 minutos y 21 segundos. El resultado de la prueba fue satisfactorio.

## **Prueba #5**

Con esta prueba se comprobará que los usuarios puedan votar anónimamente en las encuestas que sean publicadas en la página principal del periódico Trabajadores.

### **Condiciones de ejecución:**

Ninguna.

### **Entrada:**

El usuario accede al portal del periódico Trabajadores a través de un navegador web.

Se dirige a la encuesta que se encuentra publicada en portada, selecciona una o varias de las opciones de respuestas que están disponibles y envía la misma.

### **Resultado esperado:**

Se muestra al usuario un mensaje de agradecimiento por haber participado en la encuesta.

### **Evaluación de la prueba:**

Prueba satisfactoria.

## **Prueba #6**

Esta prueba comprobará que los usuarios puedan suscribirse al boletín de noticias del periódico Trabajadores.

### **Condiciones de ejecución:**

Ninguna.

### **Entrada:**

El usuario accede al portal del periódico Trabajadores a través de un navegador web.

El usuario se dirige al formulario que se muestra en la portada para la suscripción al boletín del periódico, introduce en la casilla correspondiente su dirección de correo electrónico y presiona el botón enviar.

**Resultado esperado:**

Se muestra al usuario un mensaje advirtiéndolo que para confirmar la solicitud debe acceder al enlace que le fue enviado por correo electrónico.

**Evaluación de la prueba:**

Prueba satisfactoria.

Todas las tareas que estaban propuestas para esta segunda etapa de desarrollo del sitio Trabajadores se cumplieron de forma satisfactoria, logrando con esto aumentar la estabilidad, accesibilidad y usabilidad del portal. Se realizaron pruebas al sitio a partir de las cuales se controlaba el funcionamiento de este y detectándose errores que fueron corregidos durante el transcurso de la etapa de desarrollo hasta lograr el resultado esperado.

## Conclusiones

---

Con este trabajo se presenta una versión mejorada del portal web del periódico “Trabajadores” el cual se pretende establecer como sitio modelo a seguir para el desarrollo de futuros portales web informativos utilizando el CMS Plone con lo cual se agilizará el proceso de informatización de la prensa, garantizando también la calidad del producto final.

El desarrollo del proyecto fue guiado según la metodología Scrum. Se utilizó el sistema manejador de contenidos Plone, seleccionado por la dirección del PCC (Partido Comunistas de Cuba) en coordinación con la Universidad de las Ciencias Informáticas.

El producto final cumple con estándares de programación web, reúne los servicios y funcionalidades más comunes según la prensa cubana. Las inconformidades planteadas por parte de la dirección del periódico fueron solucionadas.

De esta forma podemos concluir que los objetivos propuestos al inicio de este trabajo fueron cumplidos satisfactoriamente. Se plantean además una serie de recomendaciones a tener en cuenta para dar continuación a este proyecto.

## Recomendaciones

---

Se recomienda seguir trabajando en perfeccionar el sitio web del periódico Trabajadores, además de capacitar a los periodistas y miembros del proyecto de Informatización de la Prensa buscando con esto seguir mejorando la rapidez y eficiencia en el desarrollo de futuros portales web utilizando el CMS Plone. Es de vital importancia también mejorar la comunicación entre los medios de prensa y la Universidad de las Ciencias Informáticas.

## Bibliografía

---

*Desarrollo ágil de software. Wikipedia.*

*Encuesta. Wikipedia.*

*The international association for Open Source Content Management. Open source content management.*

*Lista de códigos de estado HTTP. Wikipedia. Disponible en:*

[http://es.wikipedia.org/wiki/Lista\\_de\\_c%C3%B3digos\\_de\\_estado\\_HTTP](http://es.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP)

*opensourceCMS. Disponible en: <http://www.opensourcecms.com>*

*Plone. Wikipedia.*

*Programación Extrema. Wikipedia. WIKIPEDIA.*

*Scrum. Wikipedia.*

*SCRUM. 2007.*

*Utilizar Plone. PloneChile, 2005.*

*Zope Architecture. Zope Corporation. Disponible en: <http://www.zope.org/ZopeArchitecture>*

*BARRANCO, A. Diferencias entre usabilidad y accesibilidad, 2004.*

*CASTILLO, A. D. Zope: El servidor de aplicaciones libre.*

*CODINA, L. Evaluación de recursos digitales en línea: conceptos, indicadores y métodos. Revista Española de Documentación Científica, 2000. 9-44.*

*FERNÁNDEZ, V. Á. Aspectos a considerar para el análisis del posicionamiento a partir del estudio de los sitios de la prensa cubana en Internet, ProyectoWeb.org, 2005.*

*FRASER, S. Real World ASP.NET: Building a Content Management System. Apress, 2002. p.*

*GARCÍA, J. C. Hacia un Web accesible, Acimed, 2007.*

*---. Usabilidad de los sitios accesibles, 2004.*

*GARCIA, X. C. and J. M. ALFONSO. Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto, 2004.*

*HALL, A. Posicionamiento en buscadores, En buscadores.*

*KRUG, S. No me hagas pensar. Madrid, Prentice Hall, 2001. p.*

*MARTÍNEZ, A. S. El posicionamiento de la prensa digital cubana. Ciego de Ávila, Cuba, Periódico Invasor.*

*MATUS, O. E. G. La ciencia del Scrum.*

MCDONOUGH, C. *The Zope Book*. Edición 2.6. Sams, 2003. p. 0735711372

MCKAY, A. and A. RUNYAN. *Definitive Guide to Plone*. Primera Edición. Springer, 2004. p. 1590593294

MENÉNDEZ, R. and A. BARZANALLANA. *Metodología de desarrollo de software*, Wikilearning, 2005.  
[Disponible en:  
[http://www.wikilearning.com/curso\\_gratis/metodologias\\_de\\_desarrollo\\_de\\_software-metodologia\\_de\\_desarrollo\\_de\\_software/3617-1](http://www.wikilearning.com/curso_gratis/metodologias_de_desarrollo_de_software-metodologia_de_desarrollo_de_software/3617-1)

NIELSEN, J. *Usabilidad. Diseño de sitios web*. Madrid, Prentice Hall, 2002. p.

PAPERS. *Case Studies, Step Two*.

REYES, L. *Presencia de los Servicios de Valor Agregados en la prensa digital cubana*. La Habana, Cuba.

RHYNO, A. *The Ten Commandments of Content Management*, 2003.

ROBERTSON, J. *How to evaluate a content management system*, Step Two, 2002.

---. *Looking towards the future of CM*, Step Two, 2003.

---. *So, what is a content management system?*, Step Two, 2003.

ROBERTSON, S. and M. R. BERNSTEIN. *The Zope Bible*. Edición Illustrated. John Wiley & Sons Inc, 2002. p. 9780764548574

SANCHEZ, M. A. M. *Metodologías De Desarrollo De Software*, 2004.

SERRANO, J. *Explicando Scrum a mi abuela*, 2007.

SUH, P. *Content Management Systems*, Glasshaus, 2003.

W3C. *Introducción a la Accesibilidad Web*, W3C. Disponible en:  
<http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

---. *Techniques for Web Content Accessibility Guidelines 1.0*, 2000. [Disponible en:  
<http://www.w3.org/TR/2000/NOTE-WCAG10-TECHS-20001106/>

---. *Web Accessibility Initiative*, W3C. Disponible en: <http://www.w3.org/WAI>

WEITERSHAUSEN, V. *Web Component Development with Zope 3*. Segunda Edición. Springer, 2005. p. 3540223592

ZÚNICA, R. R. *Accesibilidad a la Red*, Temalia, 2001.

## Referencias

---

BARRANCO, A. *Diferencias entre usabilidad y accesibilidad*, 2004.

GARCÍA, J. C. *Usabilidad de los sitios accesibles*, 2004.

GARCIA, X. C. and J. M. ALFONSO. *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto*, 2004.

*Lista de códigos de estado HTTP*. Wikipedia. Disponible en:

[http://es.wikipedia.org/wiki/Lista\\_de\\_c%C3%B3digos\\_de\\_estado\\_HTTP](http://es.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP)

MATUS, O. E. G. *La ciencia del Scrum*.

REYES, L. *Presencia de los Servicios de Valor Agregados en la prensa digital cubana*. La Habana, Cuba.

*Zope Architecture*. Zope Corporation. Disponible en: <http://www.zope.org/ZopeArchitecture>



### Anexo I. Metodología Scrum<sup>(MATUS)</sup>

La metodología *Scrum* define cada proceso como iterativo e incremental. El esqueleto de *Scrum* se muestra en la figura 3. El círculo inferior representa una iteración del desarrollo de las actividades que ocurren una tras otra. El producto de cada iteración es un incremento en el producto. El círculo superior representa la reunión diaria que ocurre durante la iteración, en la cual los miembros individualmente del grupo conocen, inspeccionan las actividades y hacen los cambios apropiados. Como resultado de la iteración queda una lista de requerimientos. Este ciclo se repite a lo largo de todo el proyecto.

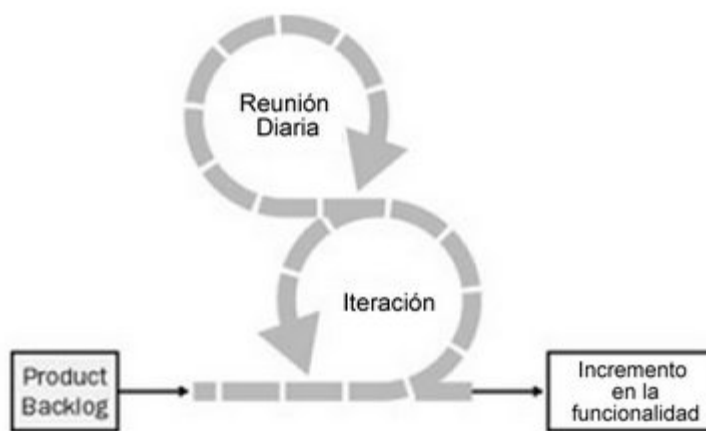


Figura 1: Esqueleto de Scrum

Figura 1. Esqueleto de Scrum.

Este esqueleto opera de esta manera:

1. Al comienzo de la iteración, el equipo revisa que es lo que debe hacer.
2. Luego, selecciona lo que cree que puede hacer para tener un incremento y un potencial prototipo funcional al término de la iteración.
3. El equipo se separa y hace su mejor esfuerzo por el resto de la iteración. Cuando ésta termina, el equipo presenta el incremento de la funcionalidad que construyó, de manera que los otros miembros del equipo puedan revisar las funcionalidades y hacer las modificaciones necesarias al proyecto.

## Los roles en Scrum

Hay solo tres roles en *Scrum*:

- Product Owner.
- Team (el equipo).
- Scrum Master.

Todas las responsabilidades de manejo de un proyecto se dividen entre estos tres papeles.

### Product Owner

Es el responsable de cuidar los intereses de cada uno de los participantes. El *Product Owner* estima el financiamiento inicial y el requerido en el curso del proyecto mediante la creación de *The project's initial overall requirements* (los requisitos totales e iniciales del proyecto), preocupándose de retornar los objetivos de inversión (ROI), y los planes de revisión.

La lista de requerimientos se llama *Product Backlog*. El *Product Owner* es el responsable de usar el *Product Backlog* para asegurarse que las funcionalidades de mayor valor sean producidas e implementadas. Frecuentemente se prioriza el primer elemento del *Product Backlog* ya que se van ordenando los requisitos de mayor a menor valor.

### Team

El equipo de desarrollo tiene la responsabilidad, en cada iteración, de transformar el *Product Backlog* en un incremento funcional para el producto, con vistas a lograr esto deben planificar su propio trabajo. Los miembros del equipo son responsables en conjunto del éxito de cada iteración y del proyecto en su totalidad.

### Scrum Master

Es responsable del proceso *Scrum*. Debe enseñar la metodología *Scrum* a cada integrante implicado en el proyecto, preocupándose de poner la metodología en práctica y hacerla parte de la cultura de trabajo de la organización, asegurándose de que cada miembro del equipo siga las reglas y prácticas de *Scrum*.

### El flujo de Scrum

Un proyecto de *Scrum* comienza con una visión del sistema que se irá desarrollando a medida que este avanza. El *Product Owner* es responsable de financiar el proyecto y además debe entregar sus ideas de

manera que se maximice su ROI. También debe formular un plan que pueda ser incluido en el *Product Backlog*.

El *Product Backlog* es una lista de los requisitos funcionales y no funcionales donde los elementos de mayor valor funcional son priorizados.

## **Sprint**

A las tareas del *Product Backlog* se les va dando cumplimiento trabajando en *Sprints*. Cada Sprint es una iteración de 30 días consecutivos (esto puede adaptarse a las necesidades del proyecto.).

Se inicia cada *Sprint* con un *Sprint Planning Meeting* (*Reunión de planeamiento del Sprint*), donde el *Product Owner* y el *Team* idean juntos lo que se espera para el siguiente *Sprint*, seleccionando del *Product Backlog* las tareas de más prioridad. De esta forma *Product Owner* le dice al *Team* qué desea, y el *Team* le dice al *Product Owner* cuánto de lo que quiere se puede transformar en funcionalidad durante el *Sprint*. Las reuniones de planeamiento del *Sprint* no pueden durar más de ocho horas. La meta es conseguir trabajar, no pensar en el trabajo.

La reunión de planeamiento del *Sprint* se divide en dos.

- Las primeras cuatro horas se dedican al *Product Owner* que presenta la prioridad más alta del *Product Backlog* al equipo. El *Team* le pregunta sobre el contenido, el propósito, el significado, y las intenciones del *Product Backlog*. Posteriormente el equipo selecciona del *Product Backlog* lo que cree poder transformar en un incremento funcional para el final del Sprint.
- Durante las segundas cuatro horas el equipo planifica su propio *Sprint*. Las tareas que componen este plan se ponen en una lista llamada *Sprint Backlog*.

Diariamente, el equipo realiza una reunión minuciosa de 15 minutos llamada *Daily Scrum* en la cual cada miembro del equipo contesta a tres preguntas:

1. ¿Qué haz terminado desde la última *Daily Scrum*?
2. ¿Qué planeas hacer a favor del proyecto entre esta y la próxima *Daily Scrum*?
3. ¿Tienes algún impedimento que te dificulte cumplir con el Sprint y con el proyecto?

El propósito de la reunión es sincronizar el trabajo de todos los miembros del equipo y programar cualquier reunión que sea necesaria para seguir avanzando.

Al final del Sprint, se realiza una Reunión de Revisión del Sprint. Ésta es una reunión de unas cuatro horas, en la cual el equipo presenta al *Product Owner* qué desarrolló durante el Sprint.

Después de la Reunión de Revisión del Sprint y antes de la Reunión de Planeamiento del próximo Sprint, el *Scrum Master* convoca a una *Sprint Retrospective* (Reunión Retrospectiva del Sprint) con el equipo. En esta reunión de tres horas, el *Scrum Master* hace que el equipo revise su proceso de desarrollo Scrum, tratando de hacerlo más eficaz y agradable para el próximo Sprint.

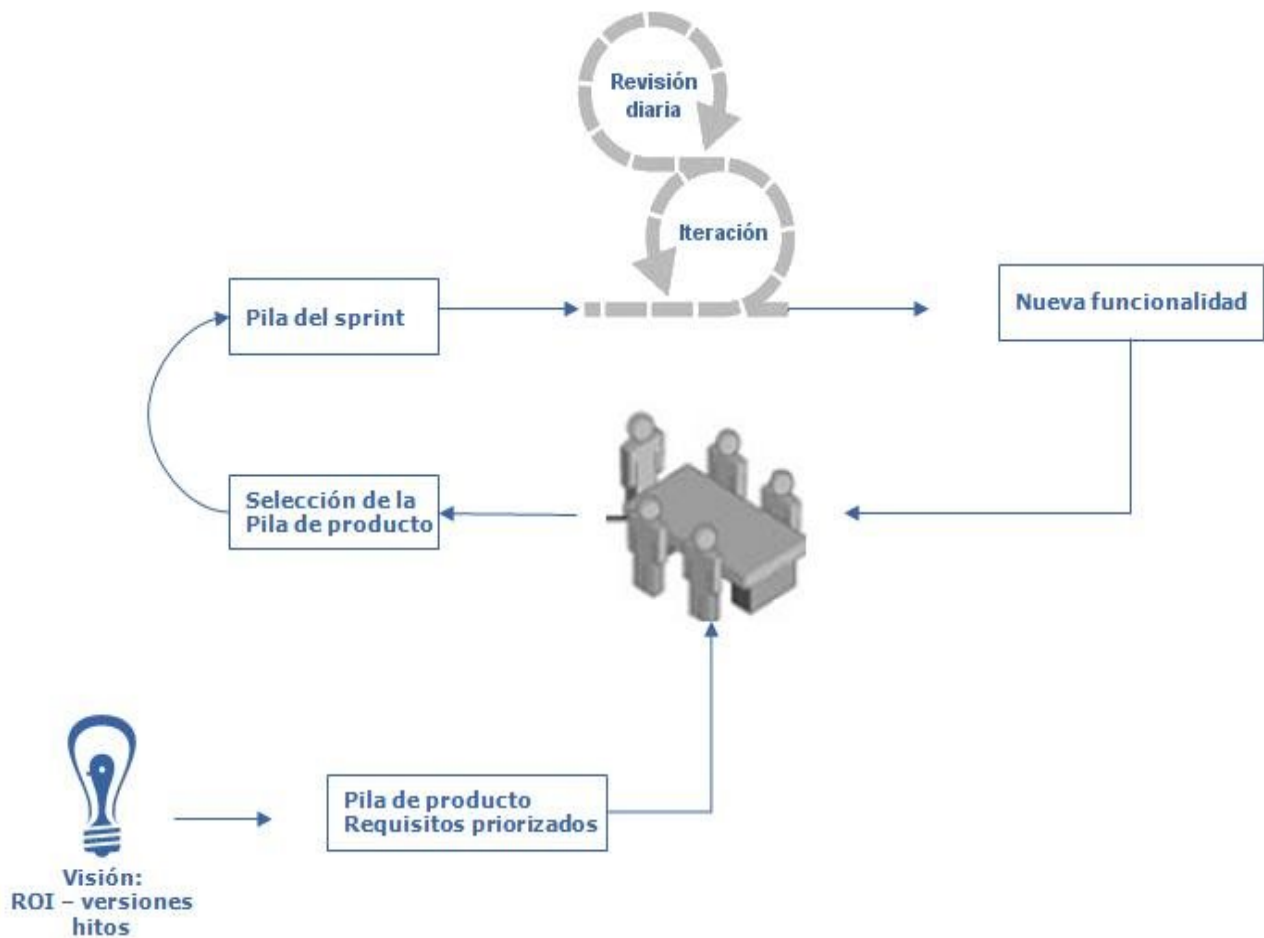


Figura 2. Diagrama del proceso Scrum.

## Artefactos de Scrum.

### Product Backlog

Id	Módulo	Descripción	Est.	Por
<b>Critico</b>				
1		Plataforma tecnológica	30	AR
2	Cliente	Interfaz de usuario	40	LR
3	Cliente	Un usuario se registra en el sistema	40	LR
4	Trastienda	El operador define el flujo y textos de un expediente	60	AR
5	Trastienda	Etc...	999.	XX
<b>Necesario</b>				
6	Cliente	El usuario modifica su ficha personal	30	AR
7	Cliente	El usuario consulta los expedientes asignados	15	LR
8	Cliente	El usuario tramita un expediente	35	LR

**Figura 3. Product Backlog**

Los requisitos del producto a desarrollar se enumeran en el *Product Backlog*. El *Product Owner* es responsable del contenido, priorización, y disponibilidad del *Product Backlog*.

Es simplemente una estimación inicial de los requisitos. Se desarrolla paralelamente a medida que el producto y el ambiente en el cual se trabaja evoluciona. Es dinámico. Maneja constantemente los cambios para identificar que necesita el producto para ser: apropiado, competitivo, y útil. Mientras exista un producto, el *Product Backlog* también existe.

## Sprint Backlog

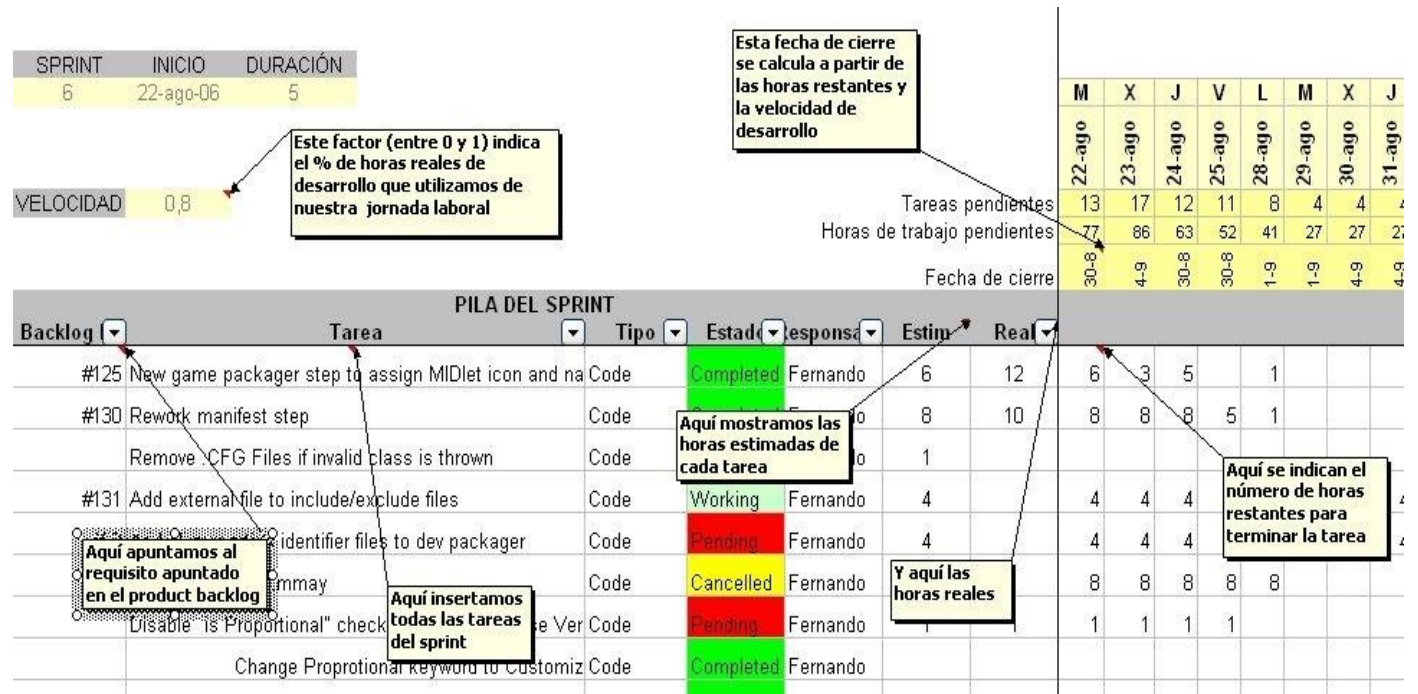


Figura 4. Sprint Backlog

El *Sprint Backlog* define las tareas seleccionadas del *Product Backlog* que el equipo desarrollará para lograr incrementos potencialmente funcionales del producto. El equipo crea una lista inicial de estas tareas en la segunda parte de la reunión de planificación del *Sprint*. Las tareas deben ser divididas de modo que cada una demore entre 1 a 16 horas finalizarlas. Las tareas de largo mayor de 16 horas se consideran secundarias, ya que todavía no se han definido apropiadamente. Solamente el equipo puede cambiar el *Sprint Backlog*.

## Anexo II. Códigos de estado HTTP (Lista de códigos de estado HTTP)

La siguiente es una lista de códigos de respuesta del HTTP y frases estándares asociadas, destinadas a dar una descripción corta del estatus. Estos códigos de estatus están especificados por el RFC 2616, y algunos fragmentos en los estándares RFC 2518, RFC 2817, RFC 2295, RFC 2774 y RFC 4918; otros no están estandarizados, pero son comúnmente utilizados.

El primer dígito del código de respuesta especifica una de las cinco clases de respuesta.

### **1xx: Respuestas informativas**

Petición recibida, continuando proceso.

Esta clase de código de estatus indica una respuesta provisional, que consiste únicamente en la línea de estatus y en encabezados opcionales, y es terminada por una línea vacía. Desde que HTTP/1.0 no definía códigos de estatus 1xx, los servidores *no deben* enviar una respuesta 1xx a un cliente HTTP/1.0, excepto en condiciones experimentales.

#### **100 Continúa**

Esta respuesta significa que el servidor ha recibido los encabezados de la petición, y que el cliente debería proceder a enviar el cuerpo de la misma (en el caso de peticiones para las cuales el cuerpo necesita ser enviado; por ejemplo, una petición Hypertext Transfer Protocol. Si el cuerpo de la petición es largo, es ineficiente enviarlo a un servidor, cuando la petición ha sido ya rechazada, debido a encabezados inapropiados. Para hacer que un servidor cheque si la petición podría ser aceptada basada únicamente en los encabezados de la petición, el cliente debe enviar: `Expect: 100-continue`, como un encabezado en su petición inicial y verificar si un código de estado 100 Continue es recibido en respuesta, antes de continuar (o recibir `417 Expectation Failed` y no continuar).

#### **101 Conmutando protocolos**

#### **102 Procesando (WebDav - RFC 2518)**

## **2xx: Peticiones correctas**

Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.

### **200 OK**

Respuesta estándar para peticiones correctas.

### **201 Creado**

La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

### **202 Aceptada**

La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente pudiere no ser satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.

### **203 Información no autoritativa (desde HTTP/1.1)**

### **204 Sin contenido**

### **205 Recargar contenido**

### **206 Contenido parcial**

La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.

### **207 Estado múltiple (Multi-Status, WebDav)**

El cuerpo del mensaje que sigue es un mensaje XML y puede contener algún número de códigos de respuesta separados, dependiendo de cuántas sub-peticiones sean hechas.



### **3xx: Redirecciones**

El cliente tiene que tomar una acción adicional para completar la petición.

Esta clase de código de estado indica que una acción subsecuente necesita efectuarse por el agente de usuario para completar la petición. La acción requerida puede ser llevada a cabo por el **agente de usuario** sin interacción con el usuario si y sólo si el método utilizado en la segunda petición es `GET` o `HEAD`. El agente de usuario no debe redirigir automáticamente una petición más de 5 veces, dado que tal funcionamiento indica usualmente un “Bucle infinito”.

#### **300 Múltiples opciones**

Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para video, listar archivos con distintas extensiones o *word sense disambiguation*.

#### **301 Movido permanentemente**

Esta y todas las peticiones futuras deberían ser dirigidas a la URI dada.

#### **302 Movido temporalmente**

Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar. La especificación HTTP/1.0 (RFC 1945) requería que el cliente realizara una redirección temporal (la frase descriptiva original fue "*Moved Temporarily*"), pero los navegadores populares lo implementaron como **303 See Other**. Por tanto, HTTP/1.1 añadió códigos de estado 303 y 307 para eliminar la ambigüedad entre ambos comportamientos. Sin embargo, la mayoría de aplicaciones web y librerías de desarrollo aún utilizan el código de respuesta 302 como si fuera el 303.

#### **303 Vea otra (desde HTTP/1.1)**

La respuesta a la petición puede ser encontrada bajo otra URI utilizando el método `GET`.

#### **304 No modificado**

Indica que la petición a la URL no ha sido modificada desde que fue requerida por última vez. Típicamente, el cliente HTTP provee un encabezado como *If-Modified-Since* para indicar una fecha y hora

contra la cual el servidor pueda comparar. El uso de este encabezado ahorra ancho de banda y reprocesamiento tanto del servidor como del cliente.

### **305 Utilice un proxy (desde HTTP/1.1)**

Muchos clientes HTTP (como Mozilla e Internet Explorer) no se apegan al estándar al procesar respuestas con este código, principalmente por motivos de seguridad.

### **306 Cambie de proxy**

Esta respuesta está descontinuada.

### **307 Redirección temporal (desde HTTP/1.1)**

Se trata de una redirección que debería haber sido hecha con otra URI, sin embargo aún puede ser procesada con la URI proporcionada. En contraste con el código 303, el método de la petición no debería ser cambiado cuando el cliente repita la solicitud. Por ejemplo, una solicitud POST tiene que ser repetida utilizando otra petición POST.

## **4xx Errores del cliente**

La solicitud contiene sintaxis incorrecta o no puede procesarse.

La intención de la clase de códigos de respuesta 4xx es para casos en los cuales el cliente parece haber errado la petición. Excepto cuando se responde a una petición HEAD, el servidor **debe** incluir una entidad que contenga una explicación a la situación de error, y si es una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud (como GET o POST). Los agentes de usuario deben desplegar cualquier entidad al usuario. Estos son típicamente los códigos de respuesta de error más comúnmente encontrados.

### **400 Solicitud incorrecta**

La solicitud contiene sintaxis errónea y no debería repetirse.

### **401 No autorizado**

Similar al *403 Forbidden*, pero específicamente para su uso cuando la autenticación es posible pero ha fallado o aún no ha sido provista. Vea autenticación HTTP básica y *Digest access authentication*.

### **402 Pago requerido**

La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de Dinero electrónico o micropagos, pero eso no sucedió, y este código nunca se utilizó.

### **403 Prohibido**

La solicitud fue legal, pero el servidor se rehúsa a responderla. En contraste a una respuesta 401 No autorizado, la autenticación no haría la diferencia.

### **404 No encontrado**

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

### **405 Método no permitido**

Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza *GET* en una forma que requiere que los datos sean presentados vía *POST*, o utilizando *PUT* en un recurso de sólo lectura.

### **406 No aceptable**

### **407 Autenticación Proxy requerida**

### **408 Tiempo de espera agotado**

El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro.

### **409 Conflicto**

### **410 Ya no disponible**

Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Este código debería ser utilizado cuando un recurso haya sido quitado intencionalmente; sin embargo, en la práctica, un código 404 No encontrado es expedido en su lugar.

### **411 Requiere longitud**

### **412 Falló precondition**

#### **413 Solicitud demasiado larga**

#### **414 URI demasiado larga**

#### **415 Tipo de medio no soportado**

#### **416 Rango solicitado no disponible**

El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.

#### **417 Falló expectativa**

#### **421 Hay muchas conexiones desde esta dirección de internet**

#### **422 Entidad no procesable (WebDav - RFC 4918)**

La solicitud está bien formada pero fue imposible seguirla debido a errores semánticos.

#### **423 Bloqueado (WebDav - RFC 4918)**

El recurso al que se está teniendo acceso está bloqueado.

#### **424 Falló dependencia (WebDav) (RFC 4918)**

La solicitud falló debido a una falla en la solicitud previa.

#### **425 Colección sin ordenar**

Definido en los **drafts** de WebDav Advanced Collections, pero no está presente en "Web Distributed Authoring and Versioning (WebDav) Ordered Collections Protocol" (RFC 3648).

#### **426 Actualización requerida (RFC 2817)**

El cliente debería cambiarse a TLS/1.0.

#### **449 Reintente con**

Una extensión de Microsoft: *La petición debería ser reintentada después de hacer la acción apropiada.*

## **5xx Errores de servidor**

El servidor falló al completar una solicitud aparentemente válida.

Los códigos de respuesta que comienzan con el dígito "5" indican casos en los cuales el servidor tiene registrado aún antes de servir la solicitud, que está errado o es incapaz de ejecutar la petición. Excepto cuando está respondiendo a un método *HEAD*, el servidor debe incluir una entidad que contenga una explicación de la situación de error, y si es una condición temporal o permanente. Los agentes de usuario deben desplegar cualquier entidad incluida al usuario. Estos códigos de repuesta son aplicables a cualquier método de petición.

### **500 Error interno**

Es un código comúnmente emitido por aplicaciones empotradas en servidores web, las que generan contenido dinámicamente, por ejemplo aplicaciones montadas en IIS o Tomcat, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.

### **501 No implementado**

### **502 Pasarela incorrecta**

### **503 Servicio no disponible**

### **504 Tiempo de espera de la pasarela agotado**

### **505 Versión de HTTP no soportado**

### **506 Variante también negocia (RFC 2295)**

### **507 Almacenamiento insuficiente (WebDav - RFC 4918)**

### **509 Límite de ancho de banda excedido**

Este código de estatus, a pesar de ser utilizado por muchos servidores, no es oficial.

### **510 No extendido (RFC 2774)**

## Glosario de Términos

---

**Blog (Web log):** Diario en formato web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

**CMF (Content Management Framework):** Entorno a programación de aplicaciones enfocado al desarrollo de CMS.

**CMS (Content Management System):** Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de webs.

**GPL (General Public License):** Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

**CSS (Cascading Style Sheets):** Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una web. Se llaman "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en una web.

**HTML (HyperText Markup Language):** Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en webs.

**HTTP (HyperText Transfer Protocol):** Protocolo cliente-servidor utilizado para el intercambio de páginas web (HTML)

**Metadatos:** Datos sobre los datos. Información que describe el contenido de los datos. Por ejemplo de un documento serian metadatos, entre de otros, su título, el nombre del autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.

**Proxy:** Hace referencia a un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la de servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP.

**Python:** Lenguaje interpretado de alto nivel orientado a objetos.

**RFC:** Petición de comentarios (el plural es Peticiones de Comentarios, en cuyo caso sus siglas en inglés son RFCs) es uno dentro de una serie de documentos informativos numerados de Internet y estándares que tanto el software comercial y el freeware en Internet y las comunidades Unix siguen ampliamente.

**TLS (Transport Layer Security):** Seguridad de la Capa de Transporte es un protocolo criptográfico que proporciona comunicaciones seguras en Internet. Existen pequeñas diferencias entre SSL 3.0 y TLS 1.0, pero el protocolo permanece sustancialmente igual. El término "SSL" según se usa aquí, se aplica a ambos protocolos a menos que el contexto indique lo contrario.

**URL (Uniform Resource Locator):** Dirección de un recurso en la web. Tiene el formato `protocol://màquina.domini:port/ruta/recurs`. Por ejemplo `http://www.uoc.edu/dt/20396/index.html` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

**WAI (Web Accessibility Initiative):** Iniciativa del Consorcio de la World Wide Web para asegurar que las webs están diseñadas pensando en el acceso de personas con discapacidades.

**Web:** Sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el web, un web) se refiere a un sitio web entero, en cambio si se utiliza en femenino (la web, una web) se refiere a una página web concreta dentro del sitio web.

**WebDav (Web-based Distributed Authoring and Versioning):** Es una extensión del protocolo HTTP que permite a los usuarios editar y administrar ficheros de forma colaborativa en servidores web remotos (definición <http://www.webdav.org/de>)

**WYSIWYG (What You See Is What You Get):** Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.