

ISSN: 2773-7349

Sociedad & Tecnología

Revista del Instituto Tecnológico Superior Jubones

2019

Volumen / 2
Número / 2
Julio / Diciembre

MODELADO DEL SISTEMA AUTOMÁTICO DE DESCARGAS E IMPLEMENTACIÓN DEL MÓDULO DE DESCARGAS AUTOMÁTICAS

MODELING OF THE AUTOMATIC DOWNLOADS SYSTEM AND IMPLEMENTATION OF THE AUTOMATIC DOWNLOADS MODULE

Jesús Alejandro Carrasco Viera¹

E-mail: jacarrasco@ucf.edu.cu

ORCID: <https://orcid.org/0000-0001-9964-0937>

José Ignacio Cruz Moreira¹

E-mail: jcruz@ucf.edu.cu

ORCID: <https://orcid.org/0000-0003-1681-2364>

¹Departamento de Informática, Universidad de Cienfuegos, Cuba

Cita sugerida (APA, séptima edición)

Carrasco Viera, J. A., & Cruz Moreira, J. I. (2019). Modelado del sistema automático de descargas e implementación del módulo de descargas automáticas, *Revista Sociedad & Tecnología*, Volumen 2(2), 14-21.

RESUMEN

La presente investigación se llevó a cabo en la Universidad de Cienfuegos con el fin de diseñar un sistema informático e implementar el módulo de Descargas Automáticas para el control, seguimiento y procesamiento automático de las solicitudes de descargas hechas por parte de los usuarios de la red de dicho centro. Para el desarrollo de la aplicación informática se utilizó la arquitectura cliente-servidor y el patrón Modelo Vista Controlador; se usó Python como lenguaje de programación del lado del servidor y el framework Django. El Sistema de Gestión de Base de Datos usado fue PostgreSQL, el Proceso Unificado de Desarrollo de Software como metodología de desarrollo de software, Visual Paradigm para UML como herramienta CASE y PyCharm como Entorno de Desarrollo Integrado. El sistema desarrollado mantiene un control del flujo de información que hasta ahora

ocurría de forma manual y gestiona todas las solicitudes de descargas de los usuarios. Además, el módulo de Descargas Automáticas se encarga de procesar las descargas, informar a cada usuario el estado de su descarga y a cada administrador cualquier error que ocurra al procesar las solicitudes. También elimina cada archivo descargado luego que pase un determinado tiempo en los servidores. De esta forma se logró mejorar el proceso de solicitud de descarga en todos sus aspectos.

Palabras clave:

descarga, sistema automático de descarga, tareas programadas.

ABSTRACT

This research was carried out at the University of Cienfuegos in order to

design a computer system and implement the Automatic Downloads module for the automatic control, monitoring and processing of download requests made by users of the Internet. said center. For the development of the computer application, the client-server architecture and the Model View Controller pattern were used; Python was used as the server-side programming language and the Django framework. The Database Management System used was PostgreSQL, the Unified Software Development Process as a software development methodology, Visual Paradigm for UML as a CASE tool and PyCharm as an Integrated Development Environment. The developed system maintains a control of the information flow that until now occurred manually and manages all the users' download requests. In addition, the Automatic Downloads module is responsible for processing downloads, informing each user of the status of their download and each administrator of any error that occurs when processing requests. It also deletes each downloaded file after it spends a certain time on the servers. In this way, the download request process was improved in all its aspects.

Keywords:

download, automatic download system, scheduled tasks.

INTRODUCCIÓN

El Departamento de Redes de la Universidad de Cienfuegos brinda varios servicios a sus usuarios, entre ellos se encuentran el Correo Electrónico, Internet y Acceso Telefónico. También cuenta con una página web oficial donde se exponen noticias actualizadas de índole local, nacional e internacional(*Intranet UCf – Intranet UCf, s. f.*).

La Universidad se divide en varias áreas conformadas de acuerdo a la organización de la misma. Cada área cuenta con un Departamento de Red, conformado por el Administrador de Red, encargado de

atender a los usuarios. Debido a que el servicio de Acceso a Internet está limitado por una cuota semanal hay ocasiones donde los usuarios necesitan realizar descargas que superan con creces su cuota asignada.

Cualquier usuario puede solicitar la descarga de archivos, siempre y cuando sean para uso académico e investigativo. Los administradores de red son responsables de verificar su uso, descargar dicho archivo solicitado, publicarlo en el sitio descargas.ucf.edu.cu e informar al usuario en cuestión por el estado de su solicitud.

Luego que cada solicitud es procesada y descargada, y el archivo publicado en el sitio de descargas ningún Administrador de Red tiene control sobre si los archivos fueron descargados o no, esto conlleva al no aprovechamiento del almacenamiento de los servidores.

Metodologías, tecnologías y lenguajes utilizados

Metodología y el lenguaje de modelado utilizados en la propuesta

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología tienen la necesidad de venir documentadas para que los programadores que estarán dentro de la planeación del proyecto comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir.

RUP unifica a todo el equipo de desarrollo de software y optimiza la productividad de cada miembro del equipo, brindándoles la experiencia de los líderes de la industria y las lecciones aprendidas de miles de proyectos.(Philippe Kruchten, 2007)

Tecnologías

Arquitectura cliente-servidor: TCP es un protocolo orientado a conexión dónde las aplicaciones utilizan un modelo cliente-servidor en las comunicaciones. Un servidor es una aplicación que ofrece un servicio a usuarios de internet; un

cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.(*cliente-servidor*, 2017)

Python: Permite es mucho más simple y elegante para programar, permite dividir el programa en módulos reutilizables desde otros programas Python.(*Python - EcuRed*, s. f.)

Framework utilizado

Django: Inspirado en la filosofía de desarrollo Modelo Vista Controlador (MVC). Permite una generación automática de código, es útil para las entradas de formulario y validación, así como para el manejo de errores y permite una amplia personalización de aspectos y temas.(*Tutorial Django Parte 2*, s. f.)

Servidor de Base de Datos

PostgresSQL: Es un potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de alto rendimiento.(*PostgreSQL: About*, s. f.)

Herramientas Utilizadas

Visual Paradigm for UML: Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.(*Visual Paradigm para UML*, 2017)

PyCharm: Es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje

de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional, que incluye más características como el soporte a desarrollo web con varios precios.(*Pycharm - EcuRed*, s. f.)

Supervisor: Supervisor es un sistema cliente/servidor que permite a sus usuarios monitorear y controlar un número de procesos en sistemas operativos tipo UNIX, como lo es Linux. Es simple de usar ya que se configura a través de un archivo de configuración simple de estilo INI que es fácil de aprender, proporciona un lugar para comenzar, detener y monitorear los procesos, tiene un protocolo simple de notificación de eventos que los programas escritos en cualquier lenguaje pueden usar para monitorearlo, entre otras características.(*Supervisor: A Process Control System — Supervisor 4.2.0 documentation*, s. f.)

Análisis y diseño de la solución

Modelo del negocio

Cualquier usuario de la red de la Universidad puede solicitar una descarga en el nodo destinado para su área. El Administrador de Red de dicho nodo es el encargado de gestionar las solicitudes de usuarios interesados en descargar cualquier archivo, video de YouTube, o directorio de Internet y verificar que sean para uso docente o investigativo. Dichas solicitudes se hacen personalmente o se pueden enviar mediante correo electrónico a con asunto: "Solicitud de Descarga", aunque pudiera contener cualquier variante que el administrador entienda que se trate de una solicitud de descarga y en el cuerpo del correo contenga la URL de la descarga. Cabe destacar que, si la descarga es muy grande, el administrador pone esa descarga en cola, para que se descargue en un momento con bajo tráfico en la red,

que suele ser en el horario de la noche, de 8:00 PM hasta que finalice la cola, en otro caso, que el tamaño de la descarga sea pequeño puede descargarlo al momento.

Si la descarga solicitada se trata de un archivo, usualmente, se utiliza el software Internet Download Manager (IDM) en Windows, el comando wget en Linux o simplemente el navegador, dependiendo de si es necesario programar la descarga. En el caso que sea un video de YouTube lo que pide el usuario, se utiliza el IDM o un plugin para el navegador llamado SaveFromNetHelper. Por último, si es un directorio, la alternativa utilizada es wget, debido a las opciones de configuración que brinda.

Una vez culminada la descarga, en caso de ser un directorio, se comprime para facilitar su entrega al usuario. Luego, el Administrador ubica el archivo en el sitio <http://descargas.ucf.edu.cu> e informa al usuario la ubicación del mismo, ya sea respondiéndole el correo enviado con la solicitud o en persona.

Reglas del negocio

- Las descargas se realizan en un horario que no afecte la calidad de los servicios de la red, normalmente en el horario de la noche, a partir de las 8:00 PM.
- Una vez finalizada la descarga del administrador de red deberá informar al usuario que solicitó la descarga.
- Las descargas deben ser para uso docente e investigativo.

Diagrama de casos de uso del negocio

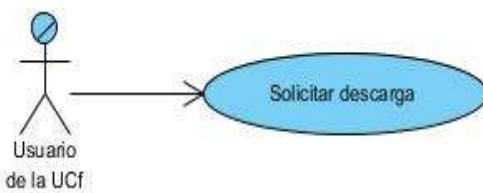


Figura 1 Diagrama de Casos de Uso del Negocio

Actores y Trabajadores del negocio

Tabla 1. Actor del Negocio

| Nombre del actor | Descripción |
|-------------------|---|
| Usuario de la UCF | Usuario de la red de la Universidad que se encarga de solicitar las descargas. Es el que se beneficia de la solicitud de descargas. |

Tabla 2. Trabajador del Negocio

| Nombre del trabajador | Descripción |
|-----------------------|--|
| Administrador de red | Se ocupa de aprobar cada descarga solicitada por los usuarios de la Universidad, realizar o programar la descarga y de informar al usuario en cuestión sobre el estado de las descargas. |

1. Captura de requisitos

Modelo de casos de uso del sistema

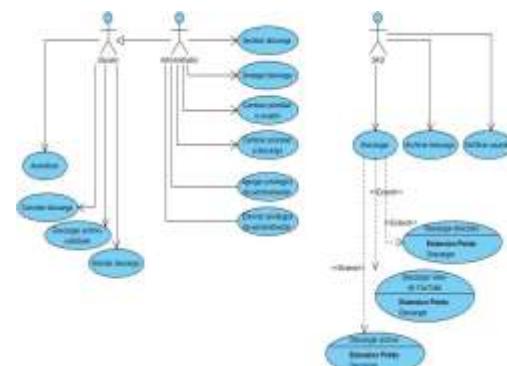


Figura 2. Diagrama de casos de uso del sistema

1. Diseño del sistema

Se obtienen en el diseño del sistema los esquemas lógico y físico de la base de datos y el diagrama de implementación del sistema.

Diagrama de base de datos

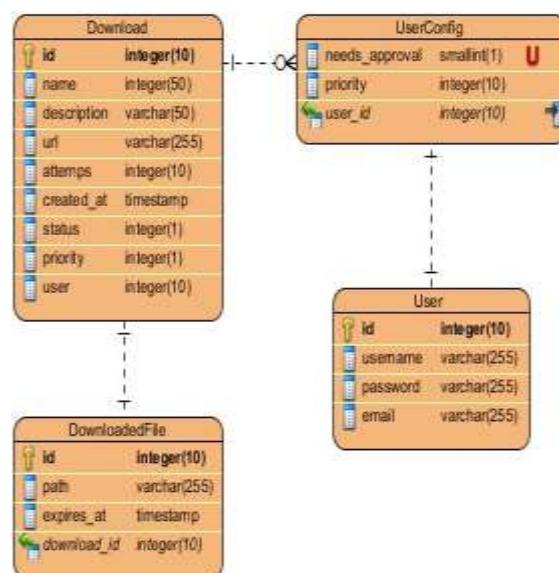


Figura 3 Diseño Físico de la Base de Datos

Diagrama de Implementación

Un diagrama de implementación muestra las organizaciones y dependencias lógicas entre componentes software (código fuente, binarios o ejecutables) y los componentes de hardware.

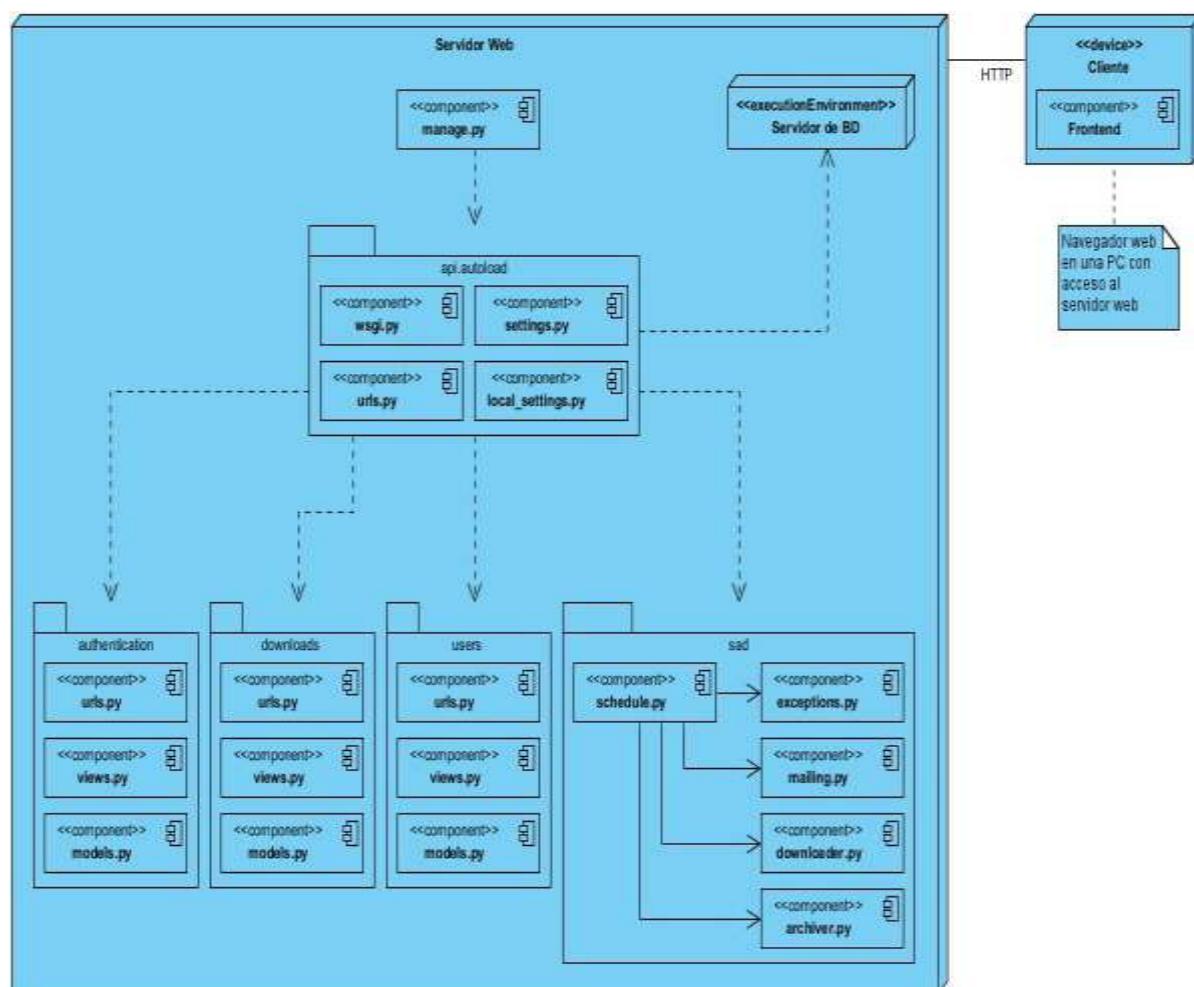


Figura 4. Diagrama de Implementación

Elaboración propia

Estudio de factibilidad

El estudio de factibilidad del producto de software se realiza para determinar si es factible o no el desarrollo del sistema que se propone y se efectúa un análisis entre costos y beneficios.

Para el cálculo de la factibilidad de un proyecto existen diferentes métodos de estimación, en este caso se utiliza el método de estimación mediante el análisis de Puntos de Casos de Uso, que al aplicarse ofrece una serie de resultados.

Para el cálculo del Esfuerzo total, se calculan ciertos indicadores, entre los que destaca: Esfuerzo = 672.336 horas/hombre.(Mario Peralta, s. f.)

El esfuerzo estimado anteriormente representa una parte del total del esfuerzo de todo el proyecto, en este caso un 40%, que se refiere al esfuerzo total para la Implementación, es decir, para el desarrollo de las funcionalidades especificadas en los Casos de Uso.

En la siguiente tabla se detallan la distribución en porcentaje (%), para el Esfuerzo total en el desarrollo del proyecto (E):

Tabla 3. Distribución del esfuerzo para el desarrollo del proyecto

| Actividad | Esfuerzo (%) | Valor (horas/hombre) |
|--------------------------------|--------------|----------------------|
| Análisis | 20 | 336 |
| Diseño | 20 | 336 |
| Implementación | 40 | 672 |
| Pruebas | 10 | 168 |
| Sobrecarga (otras actividades) | 10 | 168 |
| Total | 100 | 1680 |

Elaboración propia

A partir de estos indicadores se obtienen la estimación del tiempo de desarrollo del proyecto, que es de aproximadamente 8 meses y la estimación del costo de desarrollo resultante es de \$3200 en moneda nacional.

Validación de la solución

Para la validación del sistema se realizan un conjunto de pruebas funcionales en las cuales se verifica de forma dinámica el comportamiento del sistema. Las pruebas funcionales se le aplican al producto final, y permiten probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales(Pruebas Funcionales - Software Testing and QA, 2017).

Este formulario titulado "Solicitar descarga" requiere los siguientes campos:

- Nombre *: Un campo de texto con placeholder "El nombre es obligatorio".
- Descripción *: Un campo de texto con placeholder "La descripción es obligatoria".
- URL *: Un campo de texto con placeholder "La URL es obligatoria".

Al final del formulario hay un botón "Solicitar" y otro "Cancelar".

Figura 5. Caso de Prueba: Solicitar descarga
Elaboración propia

Este formulario titulado "Autentifíquese" requiere los siguientes campos:

- Usuario *: Un campo de texto con placeholder "El Usuario es obligatorio".
- Contraseña *: Un campo de texto con placeholder "La Contraseña es obligatoria".

Al final del formulario hay un botón "Entrar".

Figura 6. Caso de prueba: Autenticar
Elaboración propia

CONCLUSIONES

Según los objetivos trazados en la presente investigación, se arriba a la conclusión de que se realizó un análisis del proceso de solicitud de descargas en la Universidad de Cienfuegos, detectándose como deficiencias el no control de los archivos descargados, este

proceso ocurre de forma manual provocando un mayor uso de tiempo y recursos y el aumento de errores humanos.

Se seleccionó RUP como metodología de desarrollo de software, la arquitectura cliente-servidor y el patrón MVC selecciona. Se utilizó UML como lenguaje de modelado, Python como lenguajes junto al framework Django para el lado del servidor. Además, se usaron las herramientas Visual Paradigm para UML, PostgreSQL y PyCharm. También se implementó el módulo de Descargas Automáticas obteniéndose un sistema informático dividido en cuatro módulos fundamentales (Administración, Entradas, Informes y Notificación) resultantes de 22 requerimientos funcionales que fueron definidos y se determinó que para el desarrollo del sistema informático se necesita aproximadamente un total de 5 meses y un costo de \$2000.00 CUP. Igualmente se realizó el diseño de pruebas funcionales garantizando de esta forma el control de la calidad en el desarrollo del software.

REFERENCIAS BIBLIOGRÁFICAS

Cliente-servidor. (2017, febrero 20). <http://neo.lcc.uma.es/evirtual/cd/d/tutorial/aplicacion/cliente-servidor.html>

Intranet UCf – Intranet UCf. (s. f.). Recuperado 20 de junio de 2018, de <https://intranet.ucf.edu.cu/>

Mario Peralta. (s. f.). Estimación del esfuerzo basada en Casos de Uso.

Philippe Kruchten. (2007). The Rational Unified Process: An Introduction (3.^a, ilustrada ed.). Addison-Wesley, 2004.

PostgreSQL: About. (s. f.). Recuperado 20 de junio de 2018, de <https://www.postgresql.org/about/>

Pruebas Funcionales—Software Testing and QA. (2017, mayo 18). http://www.calidadysoftware.com/testing/pruebas_funcionales.php

Pycharm—EcuRed. (s. f.). Recuperado 11 de junio de 2018, de <https://www.ecured.cu/Pycharm>

Python—EcuRed. (s. f.). Recuperado 18 de junio de 2020, de <https://www.ecured.cu/Python>

Supervisor: A Process Control System—Supervisor 4.2.0 documentation. (s. f.). Recuperado 18 de junio de 2018, de <http://supvisord.org/>

Tutorial Django Parte 2: Creación del esqueleto del sitio web. (s. f.). Documentación web de MDN. Recuperado 18 de junio de 2018, de https://developer.mozilla.org/es/docs/Learn/Server-side/Django/skeleton_website

Visual Paradigm para UML. (2017, febrero 22). <http://www.software.com.ar/p/visual-paradigm-para-uml#product-description>